



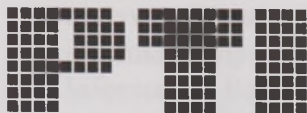
# SYSTEMY INFORMATYCZNE ZASTOSOWANIA I WDROŻENIA 2002

**TOM I**

Pod redakcją  
JANUSZA K. GRABARY  
JERZEGO S. NOWAKA

Wydawnictwa Naukowo-Techniczne  
Warszawa - Szczyrk 2002





# **SYSTEMY INFORMATYCZNE ZASTOSOWANIA I WDROŻENIA 2002**

**TOM I**

**Pod redakcją  
JANUSZA K. GRABARY  
JERZEGO S. NOWAKA**

**Wydawnictwa Naukowo-Techniczne  
Warszawa - Szczyrk 2002**

**Recenzenci:**

**Prof. WSM dr hab. Marek Greniewski**  
**Prof. P.Cz. dr hab. Nadiya Gubareni**  
**Prof. P.Cz. dr hab. inż. Sławomir Iskierka**  
**Prof. dr hab. Aleksander Katkow**  
**Prof. PWr. dr hab. Zygmunt Mazur**  
**Prof. Dr hab. inż. Bohdan Mochnacki**  
**Prof. P.Cz. dr hab. Henryk Piech**  
**Prof. P.Cz. dr hab. Janusz Szopa**

Wydanie publikacji dofinansowane przez Komitet Badań Naukowych  
i Zarząd Główny Polskiego Towarzystwa Informatycznego

**ISBN 83-204-2761-4 Tom 1-2**

**ISBN 83-204-2762-2 Tom 1**

Indeks autorów opracował mgr inż. Jarosław Łapeta  
Redakcja techniczna – mgr inż. Tomasz Lis

Fotokopie, druk i oprawę

Wykonano w Zakładzie Graficznym Politechniki Śląskiej w Gliwicach

zam. 137/02

nakł. 300+25



Polskie Towarzystwo Informatyczne organizuje corocznie konferencję w Szczyrku poświęconą sprawom rozwoju informatyki. W 2001 r. motywem przewodnim był temat efektywności zastosowań systemów informatycznych, wynikający ze współpracy z Instytutem Ekonometrii i Informatyki Politechniki Częstochowskiej. Duże zainteresowanie tą tematyką w aspekcie obserwowanych trudności w szybkich wdrożeniach aplikacji informatycznych spowodowało, że w 2002 r. postanowiono przygotować dla uczestników XIV Górskiej Szkoły PTI – Szczyrk 2002 opracowanie stanowiące reprezentatywny przegląd doświadczeń z zakresu oceny efektywności zastosowań informatyki, wzbogacone o problematykę tzw. *business intelligence* i zarządzania wiedzą (Redaktorzy opracowania nie podejmują tu dyskusji na temat poprawności językowej tych terminów). Zaproszenia wystosowano do przedstawicieli placówek naukowych, przedsiębiorstw i instytucji oraz firm informatycznych.

Nad doborem artykułów czuwała Rada Programowa konferencji w składzie:

Prof. dr hab. Jerzy Kisielnicki – Przewodniczący

Prof. dr hab. Witold Chmielarz

Dr inż. Juliusz Czarnowski

Dr Jarosław Deminet

Mgr inż. Piotr W. Fuglewicz

Prof. dr hab. Jan Goliński

Prof. dr hab. Jerzy Gołuchowski

Dr inż. Janusz K. Grabara

Prof. dr hab. Marek Greniewski

Dr inż. Waław Iszkowski

Prof. dr hab. Piotr Jędrzejowicz

Dr inż. Kazimierz Krupa

Prof. dr hab. Mirosława Lasek

Prof. dr hab. Andrzej Marciniak

Prof. dr hab. Zygmunt Mazur

Dr inż. Marek Miłoz

Prof. dr hab. Mieczysław Muraszekiewicz

Dr inż. Krzysztof Nałęcki

Mgr inż. Jerzy S. Nowak – Sekretarz Rady

Prof. dr hab. Wojciech Olejniczak

Prof. dr hab. Józef Oleński

Dr Małgorzata Pańkowska

Prof. dr hab. Stanisław Paszczyński

Dr Bogdan Pilawski  
Dr Jerzy T. Skrzypek  
Dr Witold Staniszkis  
Dr inż. Jacek Stochlak  
Prof. dr hab. Janusz Szopa  
Prof. dr hab. Zdzisław Szyjewski  
Prof. dr hab. Ryszard Tadeusiewicz  
Dr inż. Marek Valenta  
Mgr inż. Mieczysław Wroński

Powstałe w ten sposób opracowanie zawiera artykuły poświęcone problematyce szacowania efektywności zastosowań systemów informatycznych, strategii informatyzacji przedsiębiorstw, metodyce wdrożeń, kierowania projektem informatycznym oraz zastosowaniom metod matematycznych w gospodarce. Poważną część opracowania zajmują relacje typu Case Study z przebiegu wdrożeń systemów informatycznych w przedsiębiorstwach i instytucjach. Zwrócono uwagę na problematykę outsourcingu, TCO (Total Cost of Ownership), zastosowania norm ISO 9000 w informatyce, strategicznej karty wyników (Balanced ScoreCard) itp. Szereg autorów przedstawiło oryginalne relacje z wdrożeń systemów informatycznych oraz możliwości i warunki adaptacji systemów klasy MRPII/ERP do wymagań przedsiębiorstwa. Warto tu zwrócić uwagę na specyficzne zastosowania informatyki w górnictwie węgla brunatnego oraz do kontroli lotów w przestrzeni powietrznej Polski. O systemach MRPII/ERP wypowiadają się zarówno przedstawiciele nauki jak i praktycy. Tytułem eksperymentu postanowiono ująć w opracowaniu szereg artykułów o problematyce bazodanowej. Do czytelników należy ocena, czy była to trafna decyzja. O ile polscy autorzy odważnie przeprowadzają rozważania teoretyczne na temat baz danych, to pewien niedosyt budzi trudność z pozyskaniem artykułów przedstawiających praktyczne zastosowania aplikacji bazodanowych ze szczególnym uwzględnieniem hurtowni danych. Udało się również zamieścić w jednym rozdziale cykl artykułów o zastosowaniach informatyki w szkołach wyższych, zarówno do obsługi uczelni jak i omówienia prac studentów. Pozytywnie należy odnotować szereg artykułów poświęconych nowemu zjawisku na polskim rynku informatycznym, a mianowicie rozwiązaniom *business intelligence* i zarządzaniu wiedzą. Rozdział zawierający te artykuły przewrotnie nazwaliśmy „inteligencja biznesu” licząc na poczucie humoru czytelników. Sygnalizujemy tu jednak problem braku poprawnej terminologii w języku polskim, przez co te rozwiązania informatyczne są bardzo często traktowane nieufnie z powodu używania niewłaściwego nazewnictwa,

niezrozumiałego w firmach i przedsiębiorstwach. Warto również zwrócić uwagę na szereg artykułów dotyczących problematyki projektowania systemów informatycznych – jest to o tyle ważne, że praktycznie spotykamy tego rodzaju opracowania nadzwyczaj rzadko z racji powszechnej niestety praktyki wdrażania tzw. gotowych systemów. Wyrażamy obawę, że sztuka projektowania systemów informatycznych upada, co nie jest dobrym prognozą dla polskiej informatyki. Fakt pojawienia się tego rozdziału powinien stanowić dla czytelników wyraźny sygnał, że w 2003 r. zamierzamy poświęcić tym zagadnieniom znacznie więcej miejsca.

Szereg artykułów ma charakter przeglądowy i przez to czytelnik uzyskuje możliwość dostępu do aktualnych ocen tych zjawisk w krajowej literaturze informatycznej.

Różnorodny wybór nadesłanych artykułów spowodował, że redaktorzy postanowili przygotować dwie pozycje książkowe – pierwsza pod tytułem „Efektywność zastosowań systemów informatycznych – 2002” i druga nosząca tytuł „Systemy informatyczne – zastosowania i wdrożenia 2002”. Obydwa opracowania należy traktować łącznie. Redaktorzy wyrażają przekonanie, że opracowania wydane w 2001 i 2002 r. sumptem Polskiego Towarzystwa Informatycznego stanowią jeden z najbardziej obszernych przeglądów problematyki zastosowań systemów informatycznych w kraju i zapraszają czytelników do nadsyłania propozycji artykułów do wydania w 2003 r.

Przygotowanie wydawnictwa wymagało znacznego wysiłku organizacyjnego i dlatego redaktorzy opracowania składają podziękowania za pomoc w pracy Paniom Halinie Czarnowskiej i Annie Gembalczyk z Oddziału Górnośląskiego Polskiego Towarzystwa Informatycznego oraz Katarzynie Pikule z Wydziału Zarządzania Politechniki Częstochowskiej, a także Wydawnictwom Naukowo-Technicznym za znakomitą współpracę.

Redaktorzy

Janusz K. Grabara  
Jerzy S. Nowak





# Spis treści

STR.

WSTĘP

## ROZDZIAŁ 1

### INFORMATYKA – APLIKACJE BAZODANOWE

1. **Jacek CIARKOWSKI**  
WEB FARMING JAKO WSPARCIE DLA SYSTEMÓW TYPU  
E-BIZNES ..... 5
2. **Grzegorz CYSEWSKI, Krzysztof GOCZYŁA**  
ZASTOSOWANIE TECHNOLOGII WEB FARMING DLA  
POPRAWY PROCESU WYTWARZANIA OPROGRAMOWANIA 13
3. **Jacek GRUBER, Zygmunt MAZUR**  
NARZĘDZIA PROJEKTOWANIA PROTOTYPOWANIA I PRO-  
GRAMOWANIA SIECIOWYCH SYSTEMÓW BAZ DANYCH .... 23
4. **Iwona ISKIERKA**  
EKSPLOACJA DANYCH JAKO ELEMENT SYSTEMU  
WSPOMAGAJĄCEGO PODEJMOWANIE DECYZJI..... 77
5. **Stanisław KĘDZIERSKI**  
ROZDZIELANIE OBOWIĄZKÓW W PROCESACH BIZNESOWYCH 81
6. **Mikołaj MORZY, Marek WOJCIECHOWSKI**  
MECHANIZM PERSPEKTYW MATERIALIZOWANYCH W  
EKSPLOACJI DANYCH ..... 93
7. **Ryszard NIKODEM**  
ZASTOSOWANIE ZINTEGROWANYCH USŁUG KATALO-  
GOWYCH ..... 109
8. **Henryk PIECH, Łukasz KUCZYŃSKI, Dariusz LEKS**  
GENETYCZNA KONCEPCJA ZAGOSPODAROWANIA ROZPRO-  
SZONYCH ZASOBÓW..... 119
9. **Tomasz TRACZYK**  
ZASTOSOWANIE XML W HETEROGENICZNEJ ROZPRO-  
SZONEJ BAZIE DANYCH WSPIERAJĄCYCH WIELKI EKS-  
PERYMENT FIZYKI JĄDROWEJ ..... 129
10. **Marek WOJCIECHOWSKI, Maciej ZAKRZEWICZ**  
AUTOMATYCZNA PERSONALIZACJA SERWERÓW WWW  
Z WYKORZYSTANIEM METOD EKSPLOACJI DANYCH ..... 141

11. **Maciej ZAKRZEWICZ**  
KOSZTOWA OPTYMALIZACJA ZAPYTAŃ W PRAKTYCE..... 153
12. **Anna ZYGMUNT, Marek A. VALENTA**  
METODY POPRAWY EFEKTYWNOŚCI EKSPLOKACJI  
DANYCH METODĄ ASOCJACJI..... 163

## ROZDZIAŁ 2

### METODYKI WDROŻEŃ. NORMY I STANDARDY

13. **Piotr T. BIERNACKI**  
DOKUMENTOWANIE I SYMULOWANIE PROCESÓW BIZNE-  
SOWYCH – OD ISO 9000 DO SIX SIGMA ..... 175
14. **Edward GAWEŁ**  
STUDIUM PRZEDSIĘWZIĘCIA DLA PROJEKTU INFORMA-  
TYCZNEGO DLA POTRZEB ADMINISTRACJI PUBLICZNEJ  
-PROPOZYCJA STANDARDU..... 189
15. **Janusz K. GRABARA**  
WYKORZYSTANIE NARZĘDZI INFORMATYCZNYCH DO  
OCENY I CERTYFIKACJI ZARZĄDZANIA BEZPIECZEŃ-  
STWEM I HIGIENĄ PRACY ..... 207
16. **Wiesław KOSIERADZKI**  
JAK WDROŻYĆ METODYKĘ ZARZĄDZANIA PROJEKTAMI  
„PRINCE 2”®..... 213
17. **Kazimierz KRUPA**  
METODYKI WDRAŻANIA KM..... 235
18. **Jarosław ŁADYGA**  
METODY WDRAŻANIA SYSTEMÓW INFORMATYCZNYCH  
W URZĘDZIE MIASTA I GMINY ..... 247
19. **Andrzej NIEMIEC**  
ZARZĄDZANIE JAKOŚCIĄ W IT W ŚWIETLE NORM ISO ..... 261
20. **Lilianna WIERZCHOŃ**  
SYSTEM ZARZĄDZANIA JAKOŚCIĄ W PROJEKTACH PRO-  
DUKCJI OPROGRAMOWANIA W COMPUTERLAND S.A..... 291

## ROZDZIAŁ 3

### SZTUKA PROJEKTOWANIA SYSTEMÓW

21. **Anna BOBKOWSKA**  
STRATEGIA ZAPEWNIENIA JAKOŚCI OPROGRAMOWANIA . 321
22. **Witold CHMIELARZ**  
ASPEKTY PROJEKTOWANIA STRON INTERNETOWYCH  
DLA DZIAŁALNOŚCI GOSPODARCZEJ ..... 331

<b>23. Beata CZARNACKA-CHROBOT</b>	
METODA COSMIC-FFP – UNIWERSALNY SPOSÓB POMIARU I ESTYMACJI ROZMIARU FUNKCJONALNEGO SI.....	357
<b>24. Iwona GRABARA</b>	
WIZUALIZACJA INFORMACJI W SYSTEMACH INFORMA- TYCZNYCH NA PRZYKŁADZIE LOGISTYCZNYCH SYSTE- MÓW INFORMATYCZNYCH .....	379
<b>25. Wojciech KULIK, Zdzisław SZYJEWSKI</b>	
ANALIZA PROCESÓW PRACY .....	385
<b>26. Tomasz LIS, Stefan NOWAK</b>	
NIEPOWODZENIA W PROJEKTOWANIU SYSTEMÓW INFOR- MATYCZNYCH – CZYNNIKI WPŁYWAJĄCE NA SUKCES LUB NIEPOWODZENIE .....	397
<b>27. Jerzy Stanisław NOWAK</b>	
O STAROŚWIECKIM PODEJŚCIU DO PROJEKTOWANIA SYSTEMÓW INFORMATYCZNYCH .....	403
<b>28. Piotr WELENC</b>	
WYMAGANIA STAWIANE SYSTEMOM INFORMATYCZNYM W ŚWIECIE ZNOWELIZOWANEJ USTAWY O RACHUNKO- WOŚCI I PRZEPISÓW WYKONAWCZYCH .....	411
<b>29. Marzena WINIARSKA</b>	
PRZYROSTOWA BUDOWA APLIKACJI: ANALIZA BIZNESO- WA I SYSTEMOWA.....	433
<b>30. Robert ZYSKOWSKI</b>	
PRZYROSTOWA BUDOWA APLIKACJI: PROJEKTOWANIE I IMPLEMENTACJA .....	447
<b>INDEKS AUTORÓW</b>	465





## ROZDZIAŁ 1

# INFORMATYKA – APLIKACJE BAZODANOWE



# WEB FARMING JAKO WSPARCIE DLA SYSTEMÓW TYPU E-BIZNES

Jacek CIARKOWSKI

**Streszczenie:** Celem artykułu jest zaprezentowanie metodologii Web Farming oraz przykładów jej użycia. Dzięki wykorzystaniu wielu technologii związanych z Internetem (Web Mining, hurtownia danych, CRM i wiele innych) oferuje ona a bardzo silne wsparcie dla systemów e-biznes. Umożliwia on przeszukiwanie zasobów Internetu w sposób wydajny i systematyczny w celu pozyskania aktualnej informacji zgromadzonej w specyficznej formie usystematyzowanej do dalszej dystrybucji oraz analizy. W artykule omówiono przykładowe problemy oraz sposoby ich rozwiązania przy użyciu wymienionych powyższej technologii.

**Słowa kluczowe.** Web Farming, e-biznes, hurtownia danych, agenci WWW.

## Wstęp

Gwałtowny rozwój Internetu daje użytkownikom nieograniczony dostęp do informacji oraz swobodę w umieszczaniu jej na stronach WWW. Istnieje wiele czynników mających wpływ na rozwój popularności Internetu:

- liczebność użytkowników
- powszechność dostępu
- relatywnie niski koszt dostępu do pojedynczego użytkownika
- możliwość umieszczania treści nieocenzurowanych.

Czynniki mające wpływ na rozwój Internetu mogą równocześnie go hamować. Zbyt dużo użytkowników oraz nadmierna ilość wysyłanych zgłoszeń w sieci mogą być przyczyną przeciążenia, a czasami nawet zerwania połączeń. Łatwość umieszczania informacji utrudnia jej systematyzowanie oraz weryfikację pod kątem aktualności oraz poprawności informacji. Większość portali internetowych, w celu ułatwienia dostępu do informacji stosuje wiele ułatwień jak:

- umieszczanie mechanizmów wyszukiwawczych
- personalizacja użytkownika
- umożliwianie użytkownikom tworzenia ich własnych stron.

Te rozwiązania nie są tak efektywne, jakimi chcieli byśmy je widzieć. Indywidualni użytkownicy oraz przedsiębiorstwa stosują różne metody w celu uatrakcyjniania i rozbudowy ich stron WWW. Te metody często są najnowszymi rozwiązaniami i nie są przyjazne użytkownikom, stanowią barierę przy wyszukiwaniu pożądanej informacji. Każdy, kto kiedykolwiek próbował wyszukać informację na żądany temat wykorzystywał przynajmniej jedną z następujących metod:

- 1) wyszukiwanie po słowach kluczowych

2) wyszukiwanie portali specjalizowanych tematycznie – (tematyka jest zawarta w nazwie adresu URL).

Pierwsza z metod była efektywna, jeśli strona była tworzona przy wykorzystaniu HTML. Wówczas, agenci WWW wyszukiwali hosty, znajdowali strony WWW umieszczone na nich, wykorzystywali sortowanie względem słów kluczowych (umieszczonych w kodzie nagłówek stron) lub według słów zapisanych bezpośrednio w kodzie strony.

Obecnie, strony WWW zawierają nie tylko statyczny kod, w którym można łatwo odnaleźć linki (i inne użyteczne informacje) ale często dynamicznie generowane elementy graficzne stanowiące część interfejsu. Często interfejs może być wzbogacony o np. o aplety Javy, których kod może być kłopotliwy do analizy, ponieważ jest on przekazywany do przeglądarki w formie binarnej. Taka sytuacja może powodować, że część zasobów przeszukiwanego serwera może być niemożliwa do zlokalizowania, i do poddania dalszej analizie.

Drugi sposób jest efektywny tylko wtedy gdy przedsiębiorstwo lub organizacja zarezerwuje odpowiednią domenę jednoznacznie identyfikującą tę firmę. W wielu przypadkach, przedsiębiorstwa wykupują domeny, których nazwy mogą być kojarzone ze znanymi firmami lub znakami handlowymi w celu zwiększenia oglądalności ich portali lub dalszej odsprzedaży.

W trakcie przeszukiwania zasobów Internetu możemy natrafić na strony zawierające sprzeczne lub nieaktualne informacje. Ich autorzy umieszczają słowa często używane w wyszukiwaniu w nagłówkach kodu stron WWW. Ma to na celu zwiększenie ilości odwiedzin danej strony. W wyniku tego, strony są często klasyfikowane jako zgodne z tematem poszukiwań, chociaż de facto nie mają z nim nic wspólnego. prowadzi to do przeciążenia sieci WWW przez nadmierny transfer niepotrzebnych informacji. Oczywistym jest w takim przypadku wprowadzenie mechanizmów optymalizacji przesyłanej informacji oraz skrócenie czasu operacji jej wyszukiwania. Jednym z najbardziej obiecujących rozwiązań rozwiązujących powyższe problemy jest Web Farming.

## 1. Czym jest Web Farming?

Web Farming może być wykorzystywany w wielu dziedzinach naszego życia. Często nie zdajemy sobie sprawy z tego, że Web Farming jest bardzo blisko nas, ponieważ może być on nieświadomie wykorzystywany nie tylko w przedsiębiorstwach których działalność bazuje na Internecie, ale także w wielu innych.

Jednoznaczne zdefiniowanie terminu nie jest łatwe. W artykule posłużymy się definicją zaproponowaną przez Richarda Hackathorna:

*“Web Farming jest systematycznym procesem polegającym na przekształcaniu pobranych informacji z sieci WWW w celu wykorzystania ich w biznesowych” [5].*



Jak możemy zauważyć, definicja jest bardzo ogólna i wymaga sprecyzowania. Web Farming ma poniższe zadania do realizacji:

- ciągłe odwiedzanie miejsc ściśle związanych z biznesem (*pozyskiwanie*)
- zdobywanie informacji poprawnej składniowo mieszczącej się w zadanym okresie czasu (*gromadzenie*)
- strukturyzacja zdobytej informacji oraz transformacja jej na postać kompatybilną ze strukturą bazy danych w której będzie przechowywana (*organizowanie*)
- rozpowszechnianie informacji do właściwych osób, takich, które mogą ją wykorzystać w procesie biznesowym (*rozpowszechnianie*)
- zarządzanie w sposób systematyczny operacjami związanymi z przechowywaniem danych

Ogólna struktura systemu jest przedstawiona na rysunku 1. System Web Farming jest mechanizmem pośrednim pomiędzy Internetem, a klientem, który chce pobrać informacje z sieci WWW. Przepływ danych odbywa się według czterech kroków:

- 1) Klient zgłasza swoje zapotrzebowania do systemu Web Farming, który z kolei przygotowuje w jego imieniu zapytania oraz definiuje zakres wiedzy, jaką klient ma otrzymać.
- 2) System przesyła przygotowane zapytania do swoich agentów WWW (lokalnych lub zdalnych). System powinien prawidłowo identyfikować potrzeby klienta, ponieważ klient może także zgłosić nieprawidłowe lub nieprecyzyjne zapytania.
- 3) System gromadzi zebrane dane oraz przekształca je w wiedzę.
- 4) System wysyła przygotowaną wiedzę do klienta.

Używanie systemów Web Farming przez klienta jest łatwiejszym sposobem pozyskiwania informacji niż tradycyjne metody, ponieważ klient nie musi samodzielnie wyszukiwać informacji w sieci WWW ani analizować otrzymanych wyników (np. czy są zgodne z oczekiwanymi czy nie).

## 2. Web Farming w e-biznesie

Na początku powinniśmy sformułować definicję pojęcia e-biznes. Jest to trudne, ponieważ funkcjonuje równocześnie wiele definicji i żadna z nich nie określa precyzyjnie tego terminu. W tym artykule pod pojęciem e-biznesu będziemy rozumieli:

*“Prowadzenie działalności przy wykorzystaniu Internetu w dowolnym obszarze działalności firmy.”*

Oczywiście e-biznes nie musi być tylko fragmentem działalności przedsiębiorstwa, ale może stanowić jego podstawę funkcjonowania [7]. Wśród firm prowadzących działalność możemy wyróżnić trzy grupy podmiotów:

- aktywność biznesowa jest ograniczona do Internetu,
- aktywność biznesowa nie wykorzystuje Internetu,
- aktywność biznesowa przyjmuje obie powyższe formy.

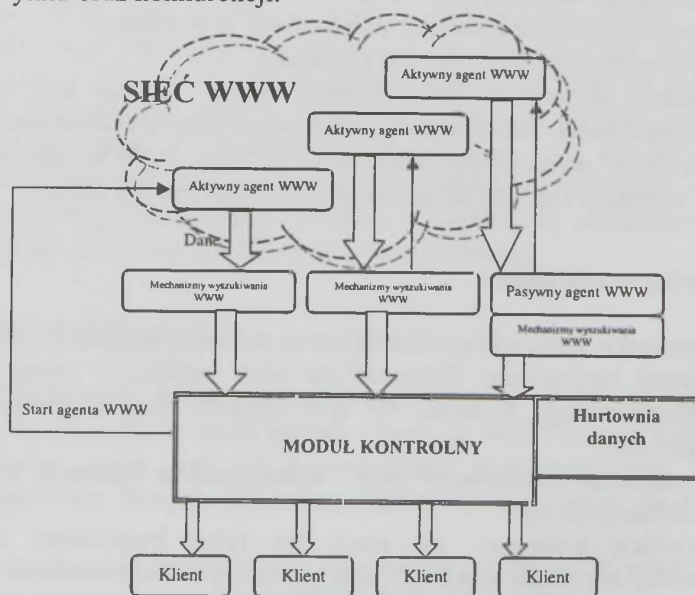
Przed stworzeniem systemu Web Farming powinniśmy zidentyfikować wszystkie główne potrzeby firmy związane z pozyskiwaną w ich imieniu informacją. Wszystkie przedsiębiorstwa zadają zazwyczaj te same grupy pytań wymienione poniżej:

- Kto oferuje taki sam produkt lub usługę?
- Jakie ceny oferuje konkurencja na swoje produkty i usługi?
- Jak pozyskują inni swoich klientów (ogłoszenia, billboardy, bannery WWW, inne)?
- Jakie odnieśli sukcesy i jakie porażki?
- Jaka jest ich kondycja?
- Gdzie znaleźć ich referencje?

Znalezienie odpowiedzi na powyższe pytania nie jest łatwe, lecz system Web Farming może być przy tym bardzo użytecznym narzędziem. Może on przeszukiwać zasoby sieci i dzięki niemu możemy zdobyć wiele użytecznej informacji.

Podczas przeszukiwania zasobów sieci WWW możemy znaleźć informacje o innych przedsiębiorstwach prowadzących podobny typ działalności.

W wyniku zapytań otrzymujemy adresy (które mogą być użyteczne do kontaktu z konkurencją), obszar ich aktywności oraz inne. Używając tych samych rozwiązań, możemy otrzymać listy cen, techniczną charakterystykę ich produktów, dane dealerów, adresy oraz inne użyteczne informacje umożliwiające określenie potrzeb na rynku oraz konkurencji.



Rys. 2. Szczegółowa struktura systemu Web Farming

Posiadając informacje (zdjęcia, ceny, parametry techniczne) o produktach, możemy również uzyskać informacje o ich klientach. Większość przedsiębiorstw posiadających własną stronę WWW umieszcza na niej informacje dotyczące referencji. Posiadając nazwę, adres i inne informacje o klientach naszej konkurencji możemy wykorzystać (używając tej samej metody, tzn. przeszukując zasoby sieci WWW) informacje o ich potrzebach. Tym sposobem możemy osiąść wiedzę o potrzebach klientów naszej konkurencji oraz samej konkurencji, co da nam zdecydowaną przewagę nad nimi.

Przeszukując zasoby sieci WWW poprzez stosowanie systemów Web Farming możemy także uzyskać informacje o sukcesach innych przedsiębiorstw, jak nowe inwestycje, wygrane procesy (w szczególności z ich klientami oraz konkurencją) i inne. Wiedza o porażkach, defektach produktów może być również przydatna.

Tę grupę informacji możemy w szczególności znaleźć na stronach WWW wydawnictw prasowych (w archiwach) oraz na forum dyskusyjnym.

Podsumowując, Web Farming może mieć duży wpływ zarówno pośrednio jak i bezpośrednio na zwiększenie efektywności przedsiębiorstw.

### 3. Ogólna architektura systemu

Cała architektura systemu jest zorientowana na efektywny i bezpieczny przepływ informacji, począwszy od hosta, z którego jest ona pobierana, a skończywszy na kliencie.

Jednym z najważniejszych problemów jest zabezpieczenie systemu przed nieautoryzowanym dostępem. Częściowo problem ten rozwiązuje zastosowanie systemów typu firewall. System musi także posiadać mechanizmy, które będą zabezpieczały go przed atakiem wirusów oraz innych aplikacji, przypadkowo pobranych wraz z poszukiwaną informacją przez system Web Farming. Dzięki temu możliwe jest bezpieczne przekazywanie informacji klientowi.

System Web Farming składa się z czterech typów elementów (patrz na rysunek 2):

- Aktywni i pasywni agenci WWW,
- Mechanizmy wyszukiwania WWW,
- Moduł kontrolny,
- Hurtownia danych.

Aktywni i pasywni agenci są bezpośrednio odpowiedzialni za pozyskiwanie informacji z sieci WWW. Aktywni agenci to aplikacje aktywowane na serwerach, z których mają pobierać informacje. Przeszukują one jego zasoby oraz transferują pozyskane dane do systemu Web Farming. Agenci mogą być również odpowiedzialni za aktualizację informacji przechowywanej w hurtowniach danych w systemie. Z powodu wielu włamań do systemów agenci muszą zapewnić bezpieczeństwo ich właścicielom oraz samemu systemowi, z którego dane są udostępniane agentom. Dlatego też powinniśmy posiadać autoryzację użytkownika systemu na używanie agentów oraz wykorzystywać technologie zapewniające



bezpieczeństwo przesyłanych danych pomiędzy agentami, systemem zawierającym dane oraz systemem Web Farming.

Pasywni agenci działają nieco inaczej. Są oni aktywowani na terminalach będących częścią systemu Web Farming. Agenci wyszukują serwery i pobierają całą przechowywaną na nich informację, a następnie przekazują ją do dalszej analizy. Ten rodzaj agentów jest bardzo popularny wśród użytkowników Internetu, którzy chcą przeglądać strony w trybie off line.

Kolejną częścią systemu Web Farming system są mechanizmy wyszukiwawcze WWW. Na tym etapie jest przetwarzana cała informacja pobrana wcześniej przez agentów. Przetwarzanie może być częściowo automatyczne, lecz ostatni etap powinien być weryfikowany przez człowieka, np. przy użyciu popularnych przeglądarek jak Microsoft Explorer czy Netscape Navigator. Im bardziej inteligentnych agentów używamy, tym mniej pracy będzie trzeba wykonywać ręcznie. Na tym etapie system (oraz częściowo agenci WWW) mogą wykorzystywać technologię sieci neuronowych [1], która jest pomocna w budowaniu inteligentnych aplikacji.

Główną częścią systemu jest moduł kontrolny. Jest on odpowiedzialny za analizę oraz weryfikację informacji przechowywanej w systemie. W przypadku bardziej zaawansowanych systemów, na tym etapie jest także zmieniany format przetworzonej informacji na postać zgodną z formatem danych przechowywanych w hurtowni danych, skąd mogą być one pobierane do kolejnych analiz. Informacja powinna być strukturyzowana zgodnie ze schematem, który umożliwi w sposób łatwy analizę oraz przetworzenie pobranej informacji na wiedzę. Schemat powinien uwzględniać co najmniej informacje wyszczególnione poniżej:

- źródło informacji (URL, IP itd.)
- typ serwera na którym znajduje się pożądana informacja (komercyjny, bezpłatny lub inny)
- słowa kluczowe umieszczone w nagłówku kodu strony (po wyeliminowaniu słów niezgodnych z zawartością strony)
- stopień zgodności tematyki strony ze słowami kluczowymi umieszczonymi w nagłówku
- data ostatniej modyfikacji
- kod strony wraz z plikami i innymi elementami umieszczonymi na stronie.

Posiadając te informacje, musimy także znać źródło pochodzenia informacji, ponieważ w celu dalszego jej użycia i zamiany na wiedzę musimy mieć pewność, że posiadamy użyteczną i pewną informację. Może być to bardzo pomocne w przypadku, gdy chcemy pobrać w przyszłości aktualną informację z autoryzowanego serwera.

Ostatnim zadaniem systemu jest dystrybuowanie informacji do klienta. Może to być przeprowadzane na wiele sposobów jak np.:

- strony WWW,
- forum dyskusyjne,
- wiadomości e-mail,
- wiadomości tekstów,



- faksy,
- inne.

Ostatnią główną częścią systemu Web Farming jest hurtownia danych. Składa się ona nie tylko z tradycyjnych baz danych ale również wielu dodatkowych aplikacji i wbudowanych mechanizmów pomocniczych. Hurtownia danych powinna służyć jako repozytorium wiedzy, którą możemy pozyskać od klienta. Jest oczywiste, że w wielu przypadkach relacyjne bazy danych nie są wystarczające do zapewnienia złożoności i rozproszenia danych. W takim przypadku odpowiednie wydaje się zastosowanie dla systemów Web Farming modelu obiektoworelacyjnego [3], [4], lub modelu opartego na języku XML [10].

#### **4. Przykłady zastosowań systemów Web Farming w e-biznesie**

Handel jest jednym z wielu dziedzin zastosowania technologii Web Farming. Kupujący oraz sprzedający potrzebują informacji na temat zmiany kursów walut, cen oraz porównywania ich w celu określenia trendów na rynku. Takie informacje w łatwy sposób może im udostępnić system Web Farming.

Kolejnym zastosowaniem systemu Web Farming jest oferowanie pomocy dla tych przedsiębiorstw, które poszukują specjalistycznych informacji. Na przykład, kancelarie prawnicze są zainteresowane odbytymi rozprawami sądowymi, ich wynikami, linią obrony w celu zastosowania najskuteczniejszych rozwiązań dla swoich klientów.

Jednym z istotnych miejsc zastosowań systemów Web Farming jest szykanie kooperantów w celu stworzenia sieci współpracujących ze sobą przedsiębiorstw. Przeszukując zasoby można odnaleźć firmy oferujące produkty lub usługi umożliwiające podniesienie efektywności przedsiębiorstwa.

Rynek turystyczny wymaga również stosowania aplikacji Web Farming do efektywnego prowadzenia biznesu. Każdy znany przewoźnik, wypożyczalnia pojazdów oraz hotel posiada własną stronę WWW na której udostępnia swoją ofertę. Dzięki systemom Web Farming potencjalny klient (lub firma udostępniająca usługę) może posiadać informację na temat takich ofert. Umożliwi to klientowi wybranie oferty najatrakcyjniejszej, a przedsiębiorcy dostosowanie swoje oferty, w taki sposób, aby być atrakcyjnym na rynku.

Web Farming może być użyteczny dla przedsiębiorstw, które chcą posiadać wiedzę o rynku i konkurencji. Przeszukując zasoby sieci WWW, system może dostarczyć informacje o innych przedsiębiorstwach obecnych na rynku, ich klientach, produktach i innych. Posiadając tę wiedzę, przedsiębiorcy mogą szybko zwiększać swój wpływ na rynku oraz skutecznie rozwijać swoją firmę.

#### **5. Problemy do dalszych badań**

W punkcie 4 opisano główne funkcje oraz generalną architekturę systemów Web Farming zorientowanych na rozwój oraz ułatwienie rozwoju

przedsięwzięcia. Pozostaje kilka otwartych problemów do dalszych szczegółowych badań:

- 1) określenie zadań jakie powinni realizować agenci WWW
- 2) zaprojektowanie schematu hurtowni danych odpowiedniego dla aplikacji Web Farming wykorzystywanych w e-biznesie.
- 3) Zaprojektowanie operacje wykonywanych na danych zgromadzonych w hurtowniach danych (odpowiednie dla e-biznesu).

Wymienione powyżej problemy są aktualnie badane. Celem jest stworzenie prototypowego systemu, który byłby użyteczny dla przedsiębiorstw prowadzących e-biznes przy realizacji ich przedsięwzięć.

## Literatura

1. Bartlett P., Anthony M.: Neural Network Learning: Theoretical Foundations, Cambridge University Press, 1999.
2. Berson A., Thearling H., Smith S.J.: Building Data Mining Applications for CRM. McGraw Hill Professional Publishing, 1999.
3. Cattell R.G.G., Barry Douglas K.: The Object Data Standard ODMG 3.0. San Francisco, Morgan Kaufman, 2000.
4. Gulutzan P., Pelzer T.: SQL-99 Complete, Really. Lawrence, Miller Freeman, 1999.
5. Hackathorn R.D.: Web Farming for the Data Warehouse. San Francisco: Morgan Kaufmann Publishers, 1999.
6. Hartman Amir, Sifonis John, Kador JohnK.E.liber s.c., E-biznes, strategie sukcesu w gospodarce internetowej, Warszawa 2001
7. Kimball R.: The Data Warehouse Toolkit – Practical Techniques for Building Dimensional Data Warehouses. New York: John Wiley & Sons, 1996.
8. Mena J.: Web Mining for Profit – E-business Optimisation. Butterworth-Heinemann, 2001.
9. WebFarming.com - Farming the Web for Systematic Business Intelligence <http://www.webfarming.com>, 2002.
10. Źródło informacji na temat języka XML [www.w3c.org/XML](http://www.w3c.org/XML) Oct 2000.

Jacek Ciarkowski  
Politechnika Gdańska  
Ul. G. Narutowicza 11/12, 80-952 Gdańsk, Polska  
E-mail: [ciara@pg.gda.pl](mailto:ciara@pg.gda.pl)  
Tel/Fax: +48 58 347 27 27

# ZASTOSOWANIE TECHNOLOGII WEB FARMING DLA POPRAWY PROCESU WYTWARZANIA OPROGRAMOWANIA

Grzegorz CYSEWSKI, Krzysztof GOCZYŁA

**Streszczenie:** Web Farming jest technologią mającą na celu odkrywanie źródeł informacji w sieci Web, zbieranie odkrytych informacji, strukturalizowanie ich i rozpowszechnianie do odpowiednich odbiorców. Poniżej prezentujemy koncepcję wykorzystania technologii Web Farming w sieci intranet firmy produkującej oprogramowanie w celu poprawy procesu wytwórczego. Zaprezentowano ogólną architekturę systemu *SoFar* oraz sposób jego wykorzystania przez różne grupy udziałowców procesu wytwarzania oprogramowania.

## 1. Cele

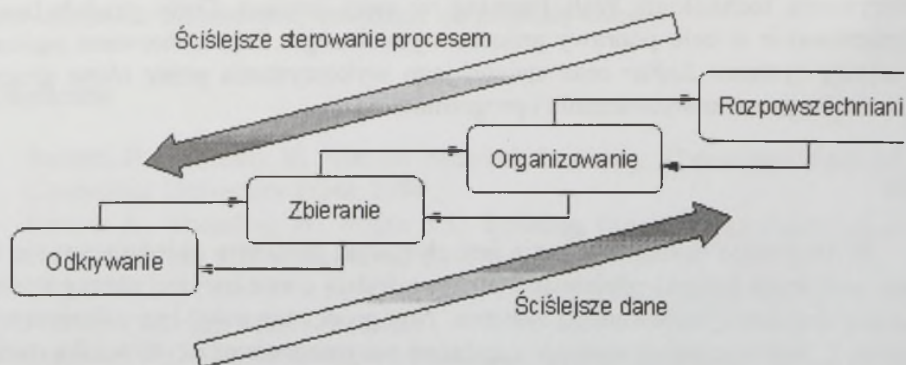
Wytwarzanie oprogramowania jest złożonym procesem składającym się z etapów, w których kolejni udziałowcy procesu dodają coraz bardziej szczegółową informację dotyczącą budowanego systemu. Aby proces ten mógł być zakończony sukcesem, tj. wytworzeniem systemu zgodnego z wymaganiami użytkownika musi być spełnionych szereg wymagań związanych ze środowiskiem wytwórczym oraz efektywnym zespołem. Każda organizacja tworząca oprogramowanie tworzy swoje własne środowisko, różniące się w zależności od rozmiaru organizacji, jak i rozmiaru budowanego systemu. Z punktu widzenia koncepcji prezentowanej poniżej interesują nas środowiska wytwórcze dla średnich i dużych projektów informatycznych. W średnich projektach często wykorzystywane są narzędzia CASE (ang. *Computer Aided Software Engineering*), wspierające wszystkie lub wybrane fazy procesu wytwarzania oprogramowania. W dużych projektach wykorzystywane są narzędzia typu CSCW (ang. *Computer Supported Cooperative Work*), tworzące środowisko składające się z narzędzi wersjonujących, dokumentujących i innych wspomagających.

Z reguły im większy jest projekt, tym bardziej rygorystyczne zasady współpracy muszą być przestrzegane. Zasady te obejmują zadania dokumentacyjne, które muszą być regularnie wykonywane przez członków zespołu. Dokumentacja ta musi być regularnie wstawiana do repozytorium projektu. W ten sposób prowadzący projekt może w każdej chwili śledzić postępy prac i ocenić stopień zaawansowania prac. Niestety, zadania dokumentacyjne zazwyczaj zwiększają obciążenie członków zespołu, co więcej, są przez niektórych z nich traktowane jako zbędne i nadmiarowe. W rezultacie repozytorium projektu pozostaje niekompletne i nieaktualne.

Problem ten dotyczy nie tylko dokumentacji projektu jako całości, ale również poszczególnych komponentów systemu (klas, funkcji), które mogłyby zostać ponownie użyte w innych projektach. Zwykle organizacje tworzą wydzielone repozytorium takich komponentów. Jednakże w takim rozwiązaniu



istnieje problem wypełniania repozytorium – zazwyczaj istnieje wydzielony administrator, który jest odpowiedzialny za wyszukiwanie nowych komponentów i wstawianie ich do repozytorium. Stanowi to duży problem, gdyż programiści i projektanci nie trzymają się często standardów wymaganych przez repozytorium komponentów ponownego użycia – komponenty nie są tworzone w celu jego ponownego użycia (ang. *for reuse*). Dodatkowo wytwórcy komponentów często nie uaktualniają repozytorium w przypadku zmiany komponentów. Ponownie zatem repozytorium takie często jest niekompletne i nieaktualne.



Rys. 1. Cztery fazy Web Farming

Trzecim problemem, z którym borykają się organizacje wytwarzające oprogramowanie, jest dostarczanie informacji dla pionu zarządzającego. Menedżerowie organizacji nie oczekują szczegółowych informacji na temat projektu, a jedynie informacji dotyczących postępu prac, zagrożeń oraz przewidywanego zakończenia projektu. Centralne repozytorium takich strategicznych informacji niestety wykazuje te same wady, co repozytoria wymienione powyżej – musi być regularnie zasilane aktualną informacją przez prowadzących projekty.

Niniejszy artykuł przedstawia koncepcję podejścia, które niweluje wymienione oraz inne problemy związane z zarządzaniem procesem wytwarzania oprogramowania. W podejściu tym informacja potrzebna na różnych poziomach projektu (wytwórca, menedżer projektu, menedżer organizacji) pozostaje rozproszona w sieci intranet organizacji i jest udostępniana dla procesu Web Farming. Proces ten zbiera potrzebną informację z serwerów źródłowych i udostępnia je użytkownikowi. W ten sposób dwa główne problemy znikają – po pierwsze: nie ma potrzeby uzupełniania repozytorium przez członków zespołu wytwórczego i po drugie: informacja zebrana w ten sposób jest zawsze aktualna.

## 2. Web Farming w procesie wytwarzania oprogramowania

Web Farming jest technologią, która umożliwia zbieranie i udostępnianie informacji z sieci Web [4]. Ogólna idea Web Farming zawiera zaplanowanie i implementację w organizacji czterofazowego procesu:

1. *Discovery (odkrywanie)*. W fazie tej (“oranie pola”) sieć Web jest przeszukiwana w celu odnalezienia źródeł zawierających istotne z punktu widzenia biznesowego dla naszej organizacji informacje. W procesie Web Farming w globalnej sieci jest to najtrudniejsza faza, wymagająca najwięcej ingerencji operatora. Jednakże w procesie Web Farming ograniczonym do lokalnego intranetu organizacji faza ta może zostać bardziej zautomatyzowana, gdyż zbiór przeszukiwanych źródeł jest ograniczony oraz mogą istnieć pewne istotne założenia dotyczące struktury odszukiwanych informacji. W procesie wytwarzania oprogramowania serwerami źródłowymi mogą być repozytoria projektów (np. repozytoria narzędzi CASE) oraz lokalne węzły poszczególnych członków zespołu.
2. *Acquisition (zbieranie)*. Podczas tej fazy (“sianie”) informacja odkryta w poprzedniej fazie jest zbierana i przechowywana w repozytorium. Fazę tę znacznie łatwiej zautomatyzować, nawet w procesie Web Farming w sieci globalnej. Ważną kwestią tej fazy jest zachowywanie informacji historycznej, zebranej wcześniej (szczególnie w aspekcie zbierania informacji o procesie wytwarzania oprogramowania, w celu określenia postępów w pracach itp.).
3. *Structuring (strukturalizacja)*. W trakcie tej fazy (“pielęgnowanie zasiewów”) zebrana informacja jest analizowana, walidowana i transformowana do formatu wymaganego przez użytkownika (odbiorcę). Błędy, braki oraz powtórzenia danych powinny zostać wykryte i usunięte. Końcowy format informacji zależy od grupy użytkowników będącej odbiorcą informacji (wytwórca, menedżer projektu, menedżer organizacji), stąd też faza ta musi być bardzo elastyczna i w pełni parametryzowalna. Strategia Web Farming zakłada, że dane zebrane w poprzednich fazach powinny zostać ustrukturalizowane w postaci składnicy danych będącej źródłem dalszych analiz [5].
4. *Dissemination (rozpowszechnianie)*. Końcowa faza Web Farming (“zbieranie plonów”) polega na udostępnieniu danych dla odbiorców (ludzi lub innych repozytoriów). Docelowo dane mogą zostać poddane dodatkowym analizom przy użyciu specjalnych narzędzi prezentacyjnych oraz narzędzi *data mining* [2, 6]. W opisywanym procesie Web Farming zbiór odbiorców informacji jest ograniczony, zatem nie jest wymagany żaden uniwersalny mechanizm.

Rysunek 1 prezentuje przebieg procesu Web Farming. Im wyższy poziom fazy, tym wyższa jakość informacji. Im niższy poziom fazy, tym wymagane jest dokładniejsze sterowanie.

Web Farming jest technologią udostępniającą również narzędzia do wykorzystania w każdej z faz: przeglądarki oraz agenci w fazie *discovery* i *acquisition* oraz narzędzia do transformacji i analizy danych, hurtownie danych i narzędzia zarządzania wiedzą w fazie *structuring* i *dissemination*. W procesie Web Farming w wytwarzaniu oprogramowania agenci Web wydają się być najbardziej

użytecznymi w fazie *discovery* i *acquisition*. Lokalni agenci, tzn. umieszczeni na pojedynczej stacji roboczej, mogą inicjować fazę zbierania informacji we współpracy z innymi lokalnymi agentami na innych stacjach. Również specjaliści agencji zdalni, tzn. aktywnie penetrujący sieć, mogą zostać użyci, gdyż wewnątrz organizacji będą oni traktowani jako agenci zaufani. Dla faz *structuring* i *dissemination* wymagane są specjalizowane narzędzia, współpracujące z narzędziami CASE używanymi w organizacji.

### 3. Struktura informacji

Web Farming w procesie wytwarzania oprogramowania (*SoFar*, ang. *Software Farming*) działa na informacji zebranej w lokalnym intranecie organizacji. Zakłada się, że formatem analizowanych informacji będzie XML/HTML (inne formaty przed analizą zostaną poddane transformacji do XML).

Zwykle w projekcie informatycznym informacja podzielona jest na szereg logicznych kategorii (węzłów logicznych), które mogą być postrzegane jako producenci i/lub konsumenci informacji w procesie Web Farming. Węzły te mogą być następującego typu (rysunek 2):

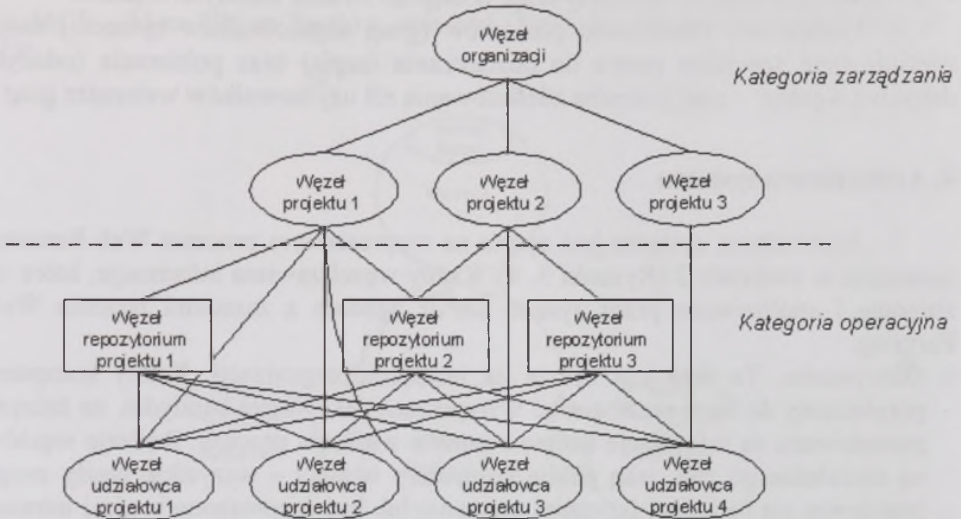
- węzeł organizacji (konsument),
- węzeł projektu (producent/konsument),
- węzeł repozytorium projektu (producent),
- węzeł udziałowca projektu (producent/konsument).

*Węzeł organizacji* zawiera informacje o wszystkich projektach realizowanych przez organizację, ich aktualnym statusie i postępie w pracach. Węzeł ten pobiera dane z węzłów projektów. *Węzeł projektu* zawiera kompletne informacje o pojedynczym projekcie – zadania, zasoby, budżet itp. Dostarcza dane dla węzła organizacji i pobiera informacje dostarczane przez węzły repozytoriów projektów oraz węzły udziałowców projektów. *Węzeł repozytorium projektu* zawiera informacje pochodzące ze zdefiniowanego repozytorium projektu oraz repozytoriów używanych w procesie wytwórczym narzędzi (CASE, CSCW). Węzeł ten dostarcza informacje dla węzłów projektów oraz węzłów udziałowców projektów. *Węzeł udziałowca projektu* jest węzłem wytwórcy (projektant, programista, ang. *developer*) zaangażowanego w dany projekt i zawiera informacje wytworzone przez niego na rzecz projektu. Węzeł ten pobiera informacje z wszystkich węzłów repozytoriów projektów, zaś dostarcza informacje dla węzłów projektów, w których dany wytwórca jest zaangażowany.

Węzły mogą zostać zaklasyfikowane jako aktywne lub pasywne. Węzeł aktywny (reprezentowany na rysunku 2 jako owal) może rozpocząć proces *SoFar*. Węzeł pasywny (prostokąt) jest tylko producentem danych dla węzłów aktywnych. Węzeł aktywny może znajdować się również w stanie pasywnym – wówczas jest producentem danych dla aktualnie aktywnych węzłów. Klasa węzła (aktywny/pasywny) jest związana z jego rolą (konsument/producent), tzn. węzeł jest w stanie aktywnym, jeśli konsumuje dane, a jest w stanie pasywnym, jeśli je produkuje.



Węzły zostały podzielone na dwie kategorie: kategoria zarządzania oraz kategoria operacyjna. Węzły kategorii zarządzania są istotne z punktu widzenia menedżera organizacji i zawierają informacje pobrane bezpośrednio przez menedżerów organizacji i projektów. Węzły kategorii operacyjnej są używane przez wytwórców systemu i dostarczają dane dla węzłów kategorii zarządzania.



Rys. 2. Klasy informacji

Rezultatem procesu *SoFar* jest informacja zbierana na trzech poziomach abstrakcji:

- poziom menedżera organizacji,
- poziom menedżera projektu,
- poziom wytwórcy.

*Poziom menedżera organizacji* zawiera informacje zebrane przez menedżerów organizacji, którzy chcą mieć ogólny wgląd na stan projektów: użyte zasoby, status, koszty, szacowany wysiłek itp. Informacja z tego poziomu zasila węzeł organizacji.

*Poziom menedżera projektu* zawiera informację o statusie projektu z punktu widzenia jego menedżera. Informacja zawiera dane na temat statusu każdego ze zdefiniowanych zadań, planów, zasobów projektu itp. Informacja z tego poziomu zasila węzły projektów.

*Poziom wytwórcy* zawiera informacje użyteczne dla wytwórcy zaangażowanego w projekt, tj. szczegóły projektowe, projekty i kody źródłowe, komponenty oraz inne elementy ogólnie pojętego *know-how*. Informacja z tego poziomu zasila węzły udziałowców projektów.

Poziomy abstrakcji definiują również poziomy bezpieczeństwa. Wymagana jest identyfikacja użytkowników, tzn. nie wszystkie informacje o

projektach są dostępne dla każdego. Przyjmuje się, że im wyższy poziom abstrakcji tym większy dostęp do informacji:

- członkowie poziomu menedżera organizacji mają dostęp do wszystkich danych,
- członkowie poziomu menedżera projektu mają dostęp do danych ich własnego projektu,
- członkowie poziomu wytwórcy mają dostęp do swoich własnych węzłów.

Dodatkowo członkowie poziomów (grup użytkowników systemu) mają zdefiniowane specjalne prawa do dostarczania (zapis) oraz pobierania (odczyt) danych z węzłów – stąd potrzeba zdefiniowania ról użytkowników wewnątrz grup.

#### 4. Architektura systemu

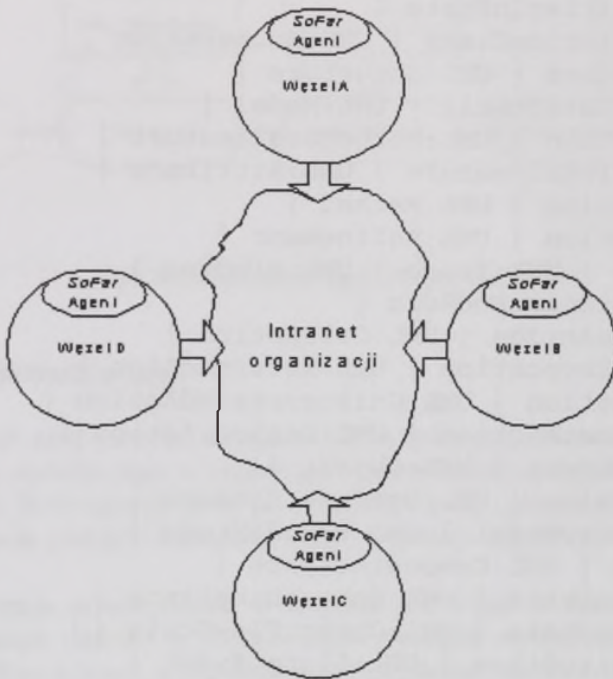
Architektura systemu jest oparta na rozproszonym procesie Web Farming opisanym w rozdziale 2 (Rysunki 3, 4). Każdy węzeł zawiera informacje, które są zbierane i analizowane przez system *SoFar* zgodnie z zasadami procesu Web Farming.

1. *Odkrywanie*. Ta faza jest oparta na intranecie organizacji. Każdy komputer przyłączony do sieci może zostać wytypowany jako węzeł-kandydat, na którym poszukiwane są informacje istotne z punktu widzenia procesu. Rodzaje węzłów są niezależne od roli, jaką pełnią komputery w sieci – wszystkie węzły mogą znajdować się na jednej fizycznej maszynie lub być rozproszone w sieci intranet organizacji. Najważniejszym zadaniem tej fazy jest sklasyfikowanie znalezionej informacji i przypisanie do logicznych węzłów.
2. *Zbieranie*. W tej fazie użyci zostają specjalizowani lokalni i zdalni agenci, którzy aktywowani są przez węzeł aktywny (konsument) na węzłach pasywnych (producent). Zebrana informacja jest przekazywana poprzez sieć. Agenci mogą pracować w sposób ciągły, zatem proces *SoFar* może być uruchomiony w tle innych zadań organizacji. Założeniem jest, iż informacja pobrana w tej fazie występuje w formacie XML; jeśli nie, to zakłada się, iż istnieje możliwość transformacji danych do formatu XML. Obecnie wszystkie narzędzia typu CASE umożliwiają taką transformację. Możliwe jest również opracowanie własnych procedur transformacji dedykowanych systemowi *SoFar*.
3. *Organizowanie*. Podczas tej fazy informacja zostaje podzielona na trzy poziomy abstrakcji opisane powyżej i przechowywana w wewnętrznym repozytorium, które powinno umożliwiać prezentację danych w sposób użyteczny dla końcowych odbiorców. W ostatnim czasie producenci systemów baz danych (Oracle, IBM DB2) udostępniają rozszerzenia swoich systemów umożliwiające łatwe manipulowanie dokumentami XML. Jednakże struktura repozytorium pozostaje ważnym tematem badawczym.  
Najważniejsza część informacji w procesie *SoFar* jest uzyskiwana z węzłów repozytoriów projektów. W podejściu *SoFar* dodatkowym założeniem jest, iż proces wytwórczy oparty jest na procesie RUP (ang. *Rational Unified Process*) [8] oraz UML. Założenie to umożliwia przedstawienie modeli UML zawartych w repozytoriach projektów w postaci XML za pomocą jednej z istniejących



definicji XML dedykowanych modelom UML. Rozważane jest użycie *OMG XML Metadata Interchange (XMI)* [7], którego głównym celem jest wymiana danych pomiędzy różnymi narzędziami do modelowania. XMI integruje trzy standardy przemysłowe:

- XML - eXtensible Markup Language, W3C standard [9],
- UML - Unified Modelling Language, standard modelowania OMG [7],
- MOF - Meta Object Facility, standard OMG dla metamodeli i metadanych [7].



Rys. 3. Architektura fizyczna systemu *SoFar*

Poniżej przedstawiono część definicji UML w postaci XML DTD.

```

<!ELEMENT UML:Binding.argument (UML:ModelElement |
UML:Comment |
UML:Namespace | UML:GeneralizableElement |
UML:Feature | UML:Parameter |
UML:Constraint | UML:Dependency |
UML:Generalization | UML:AssociationEnd |
UML:Request | UML:ActionSequence |
UML:Action | UML:Link | UML:LinkEnd |
UML:Instance | UML:AttributeLink |
UML:MessageInstance | UML:Interaction |
UML:Message | UML:StateMachine |

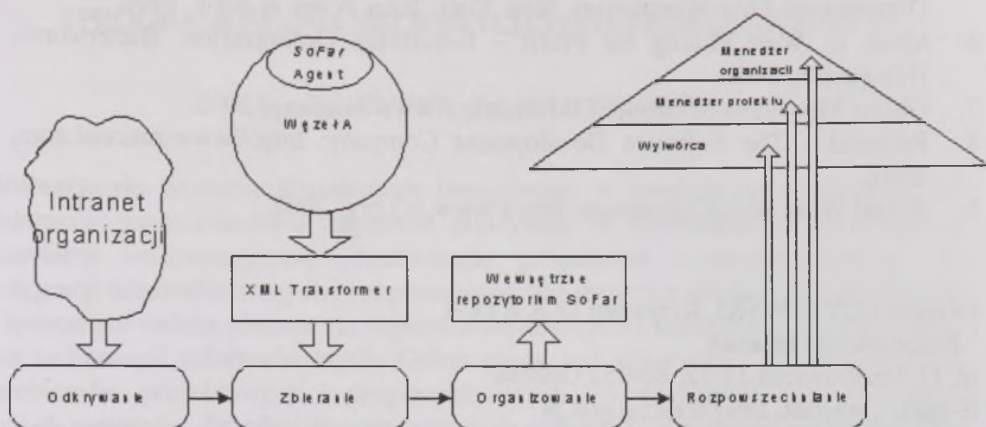
```

```

UML:Guard | UML:StateVertex |
UML:Transition | UML:Event |
UML:Partition | UML:Collaboration |
UML:Classifier | UML:Association |
UML:Stereotype | UML:Package |
UML:Signal | UML:Node | UML:Component |
UML:Interface | UML:Class | UML:DataType |
UML:Subsystem | UML:ClassifierRole |
UML:Actor | UML:UseCase |
UML:ClassifierInState |
UML:AssociationClass | UML:Enumeration |
UML:Primitive | UML:Structure |
UML:AssociationRole | UML:Model |
UML:Exception | UML:StructuralFeature |
UML:BehavioralFeature | UML:Attribute |
UML:Operation | UML:Method |
UML:Reception | UML:Refinement |
UML:Usage | UML:Trace | UML:Binding |
UML:AssociationEndRole |
UML:CreateAction | UML:CallAction |
UML:LocalInvocation | UML:ReturnAction |
UML:SendAction | UML:UninterpretedAction |
UML:TerminateAction | UML:DestroyAction |
UML:LinkObject | UML:Object |
UML:DataValue | UML:UseCaseInstance |
UML:ActivityModel | UML:PseudoState |
UML:State | UML:CompositeState |
UML:SimpleState | UML:SubmachineState |
UML:ActionState | UML:ObjectFlowState |
UML:ActivityState | UML:SignalEvent |
UML:CallEvent | UML:TimeEvent |
UML:ChangeEvent)* >

```

4. *Rozpowszechnianie*. Celem tej fazy jest zaprezentowanie odpowiednich danych w odpowiednim czasie odpowiednim odbiorcom. W fazie tej informacja z repozytorium procesu *SoFar* jest przekazywana odpowiednim konsumentom. Ważnym aspektem tej fazy jest struktura rozpowszechnianych informacji: muszą one być prezentowane w sposób jak najbardziej zbliżony do ogólnie przyjętych w organizacji standardów.



Rys. 4. Architektura logiczna systemu *SoFar*

## 5. Wnioski i przyszłe prace

Artykuł miał na celu zaprezentowanie koncepcji zbierania informacji na temat procesu wytwórczego w organizacji wytwarzającej oprogramowanie w celu jego poprawy. Rozwiązanie takie jest alternatywą dla „standardowej” procedury dokumentowania projektu informatycznego w postaci repozytorium dokumentacji projektowej.

Koncepcja przedstawiona w artykule jest wciąż w fazie uściślenia. Jej walidacja nastąpi dopiero w rzeczywistym procesie wytwórczym w wybranej organizacji. Obecne prace są ukierunkowane na zdefiniowaniu struktury informacji dla systemu *SoFar* w celu opracowania prototypu systemu. Zaproponowane rozwiązanie pozwala na przyrostowe tworzenie systemu, rozpoczynając od poziomu wydawcy, poprzez poziom menedżera projektu, aż do poziomu menedżera organizacji.

## Literatura

1. Bays M.E.: *Software Release Methodology*, Prentice-Hall, 1999.
2. Berson A., Thearling H., Smith S.J.: *Building Data Mining Applications for CRM*. McGraw Hill Professional Publishing, 1999.
3. Cotterell M., Hughes B.: *Software Project Management*. International Thomson Publishing Co., 1995
4. Hackathorn R.D.: *Web Farming for the Data Warehouse*. San Francisco: Morgan Kaufmann Publishers, 1999.

5. Kimball R.: The Data Warehouse Toolkit – Practical Techniques for Building Dimensional Data Warehouses. New York: John Wiley & Sons, 1996.
6. Mena J.: Web Mining for Profit – E-business Optimization. Butterworth-Heinemann, 2001.
7. Object Management Group (OMG): <http://www.omg.org>, 2002.
8. Rational – The Software Development Company: <http://www.rational.com>, 2002.
9. World Wide Web Consortium: <http://www.w3.org>, 2002.

Grzegorz CYSEWSKI, Krzysztof GOCZYŁA

<sup>1</sup> Politechnika Gdańska

ul. G. Narutowicza 11/12, 80-952 Gdańsk

E-mail: {grechut, kris}@eti.pg.gda.pl

Tel: +48 58 347 13 18

Fax: +48 58 347 27 27



# NARZĘDZIA PROJEKTOWANIA PROTOTYPOWANIA I PROGRAMOWANIA SIECIOWYCH SYSTEMÓW BAZ DANYCH

Jacek GRUBER, Zygmunt MAZUR

**Streszczenie.** Systemy bazodanowe integrowane w sieciach intranetowych i w Internecie mają kluczowe znaczenie praktyczne w nowoczesnych systemach z zasobami informacyjnymi. Technologie projektowe i programistyczne do integracji scentralizowanych i rozproszonych baz danych z sieciami intranetowymi i Internetem należą obecnie do najnowocześniejszych i najszybciej rozwijających się technologii informatycznych. Celem pracy jest uchwycenie najistotniejszych problemów projektowych i programistycznych występujących przy tworzenie takich systemów. Przedmiotem rozważań są standardy oraz systemy i środowiska bazodanowe do integracji rozproszonych baz danych w sieciach korporacyjnych i Internecie.

**Słowa kluczowe:** aplikacje wielowarstwowe, bazy danych, sieci intranetowe, Internet

## Wstęp

Systemy bazodanowe, biznesowe i systemy zarządzania - w skład w których wchodzi bazy danych, aplikacje, systemy zarządzania bazami danych *DBMS* (skr. ang. *Data Base Management Systems*) oraz, coraz częściej, systemy przepływu i archiwizacji dokumentów i przepływu pracy (ang. *workflow*) - ewaluują w kierunku jednolitych korporacyjnych, rozproszonych środowisk systemowych i informacyjnych. Rozwiązania i technologie stosowane do budowy takich systemów ulegały stopniowej ewolucji i obecnie do najbardziej zaawansowanych należą architektury trójwarstwowe oraz technologie aplikacji wielowarstwowych.

Całkowicie zasadne jest stwierdzenie, że internetowe środowisko *WWW* (skr. ang. *World Wide Web*) coraz bardziej wchłania bazy danych i związane z nimi zastosowania systemów bazodanowych. Za pomocą *WWW* można już obecnie nie tylko uzyskać dostęp do części zasobów informacyjnych klasycznych baz danych ale i zasoby te powiększać. Powszechne jest już publikowanie niektórych danych z baz danych - np. dla celów reklamy lub innych celów komercyjnych. Istnieje ponadto możliwość wprowadzania danych do baz - np. w celu zdalnego dokonania zakupów i realizacji innych usług i działań komercyjnych.

Współczesne bazy danych muszą być sprawnymi systemami biznesowymi oraz powinny umożliwiać zarządzanie rozproszonymi terytorialnie instytucjami korporacjami. Jednocześnie systemy bazodanowe powinny być elastyczne, ponieważ muszą nadążać za rozwojem organizacyjnym, technologicznym i materialnym instytucji oraz przyrostem liczby użytkowników. Ważną cechą

jakościową współczesnych korporacyjnych baz danych jest możliwość udostępniania części ich zasobów informacyjnych w globalnym środowisku sieciowym i operacyjnym jakim jest Internet. Sieci lokalne i Internet są naturalnymi środowiskami funkcjonowania zarówno aplikacji wielowarstwowych jak i systemów o architekturach klient-serwer.

Różnice technologiczne pomiędzy Internetem a intranetami są nieznaczne. Intranet jest rozwiązaniem pośrednim pomiędzy Internetem a klasyczną siecią lokalną. Intranety wykorzystują dobrze znane, udokumentowane oraz projektowo i programistycznie rozwinięte protokołowe technologie komunikacyjne *TCP/IP*. Istotne jest tutaj, że sieć intranetowa dziedziczy po obu rozwiązaniach ich zalety.

Intranet ułatwia i upraszcza system komunikacji i dystrybucji informacji wewnątrz firmy. Można zarządzać zasobami informacyjnymi z jednego źródła a modyfikacja i aktualizacja zasobów informacyjnych jest natychmiastowa. Dostępem do informacji można sterować. Intranety mogą być łatwo - w sposób wystarczająco bezpieczny - włączone do Internetu.

Znaczącą większość oprogramowania, usług oraz doświadczenie wdrożeniowe, eksploatacyjne i użytkowe można łatwo adaptować z Internetu do sieci intranetowych. Stopień złożoności i koszty budowy intranetów są znacząco mniejsze niż nakłady na zbudowanie heterogenicznych środowisk sieciowych, bazodanowych i informacyjnych. Aplikacje działające w Internecie można bez większego trudu zaadoptować do środowiska intranetowego. Intranety są dla instytucji bardzo wygodnym narzędziem do realizowania dostępu do baz danych, archiwizowania, katalogowania i rozsyłania dokumentów [5,20,22,24].

Rozwój technologii intranetowych jest przedmiotem licznych serwisów internetowych poświęconych wyłącznie tej tematyce. Do najciekawszych należą [23,34,35].

Najważniejszym serwisem w Internecie i w intranetach jest *WWW*. Na stronach *WWW* udostępnia się dane i informacje firmy, co oznacza tworzenie i modyfikację zawartości tych stron. Podstawowym narzędziem do przygotowywania tych publikacji jest edytor *HTML*. Do korzystania z tak publikowanych informacji wystarczają standardowe przeglądarki internetowe.

Usług intranetów i Internetu używa się do realizacji innych procesów informacyjnych, np.: ankietowania on-line, wypełniania różnych formularzy, organizowania forów dyskusyjnych, publikowania książek telefonicznych pracowników i schematów organizacyjnych instytucji, wymiany poczty elektronicznej, wymiany plików między komputerami za pomocą usługi *FTP*, organizowania dyskusji na żywo (ang. *Live Chat*) za pomocą usługi *IRC* (skr. ang. *Internet Relay Chat*), organizowania wideokonferencji.

Celem praktycznym artykułu jest uchwycenie najistotniejszych problemów projektowych i programistycznych występujących przy tworzeniu aplikacji wielowarstwowych. Analizę przeprowadzimy na wybranych reprezentantach środowisk i narzędzi programistycznych do tworzenia rozproszonych systemów bazodanowych.

Ważnym aspektem technologii projektowych i programistycznych do projektowania aplikacji wielowarstwowych i zintegrowanych sieciowo systemów bazodanowych są standardy komunikacji obiektowej dostępu do baz danych w sieciach intranetowych i w Internecie.

Obszerność tych standardów, wynikająca ze stosunkowo dużego stopnia komplikacji objętych nimi mechanizmów, a w ślad za tym duża różnorodność oprogramowania narzędziowego do projektowania i implementacji standardowych bazodanowych mechanizmów komunikacyjnych, uniemożliwia dokonanie w niniejszej pracy nawet pobieżnego ich przeglądu. Omówieniu tych standardów musiałaby być poświęcona osobna, obszerna praca.

Na potrzeby niniejszej pracy należałoby omówić - chociaż pobieżnie - standardy wybranych mechanizmów komunikacyjnych stosowanych w stosie protokołowym TCP/IP [6], standardy dotyczące współpracy obiektów - *COM* (skr. ang. *Component Object Model*) i *OLE* (skr. ang. *Object Linking and Embedding*), standard komunikacji obiektowej *ActiveX* [19], standardy obiektów rozproszonych *DCOM* (skr. ang. *Distributed Component Object Model*) [12] oraz *CORBA* (skr.ang. *Common Object Request Broker Architecture*) [3,15,17], a także mechanizmy wykorzystywane do prototypowania i implementacji mechanizmu zdalnego wywoływania procedur *RPC* (skr. ang. *Remote Procedure Call*) [16]. W pracy poprzestajemy na stwierdzeniu, czy wymienione standardy są - i w jakim stopniu - uwzględniane w zbadanych technologiach projektowych i programistycznych.

## 1. Internetowe i intranetowe systemy bazodanowe

Do niedawna aplikacje systemów bazodanowych realizowały dostęp do danych jedynie za pomocą technologii klient-serwer. Udostępnianie użytkownikowi danych z bazy umożliwiało oprogramowanie klienta działające na stacjach roboczych.

Bardziej zaawansowanymi technologicznie i funkcjonalnie są architektury wielowarstwowe aplikacji i systemów bazodanowych. Z wielu powodów są one najbardziej przydatne i wygodne w eksploatacji w Internecie i w sieciach intranetowych. Szczególnym przypadkiem architektury wielowarstwowej jest architektura trójwarstwowa. W konsekwencji dynamicznie rozwinęły się technologie projektowe i techniki programistyczne nakierowane na zintegrowanie systemów bazodanowych ze środowiskiem WWW. Technologie te są stale rozwijane i już obecnie umożliwiają projektowanie i budowę skomplikowanych scentralizowanych i rozproszonych systemów bazodanowych - nie tylko relacyjnych, ale i obiektowych - wykorzystujących możliwości funkcjonalne interaktywnych serwerów *Webu*. Zarówno Internet jak i sieci intranetowe stały się dobrej jakości środowiskiem dla systemów bazodanowych. Sieciowe systemy bazodanowe w WWW pod wieloma względami dorównują już systemom baz danych i systemom zarządzania bazami danych działającym w sposób klasyczny w



środowiskach operacyjnych i sieciowych. Dobrym przykładem takich systemów są wyszukiwarki sieciowe.

W architekturze trójwarstwowej system bazodanowy składa się z trzech warstw: interfejsu użytkownika, logiki biznesowej implementującej funkcjonalności systemu oraz bazy danych (Rys. 1.). Warstwy te są zaalokowane fizycznie na różnych komputerach i pracują w środowisku sieciowym. W takiej architekturze łatwo jest administrować systemem bazodanowym i dokonywać jego modyfikacji. W poszczególnych warstwach można to wykonywać niezależnie. Jest to podstawowa zaleta w stosunku do architektury klient-serwer, architektury aplikacji zablokowanych i jednoużytkownikowych systemów bazodanowych.



Rys. 1. Szkic topologii architektury trójwarstwowej

**Interfejs użytkownika.** Na stacjach roboczych alokuje się warstwę interfejsu użytkownika realizującą prezentację danych, ich przygotowanie oraz przekazanie warstwie aplikacji w celu wprowadzenia do bazy. Warstwę interfejsu użytkownika można zaimplementować jako stronę WWW a jej obsługę powierzyć przeglądarce. Mechanizmy komunikacji interfejsu użytkownika z warstwą aplikacyjną (biznesową) systemu bazodanowego pokrywają się wtedy z komunikacją z serwerem WWW za pomocą protokołu HTTP. Część aplikacyjną systemu bazodanowego można zatem utożsamiać z aplikacją WWW alokowaną na serwerze WWW. Podobnie postrzega się komunikowanie się warstwy aplikacji z zaalokowaną na sieciowym komputerze intranetu lub w Internecie bazą danych systemu bazodanowego, o ile istnieją mechanizmy projektowo-implementationalne udostępniania bazy z tej warstwy.

**Logika aplikacji.** Logika aplikacji implementująca reguły biznesowe jest centralną częścią całości aplikacji WWW. W warstwie tej mogą działać standardowe mechanizmy komunikacji międzywarstwowej, albo standaryzowane mechanizmy z technologii firmowych. Wykorzystując ich możliwości można tworzyć aplikacje - w najprostszym przypadku aplikacje skryptowe - i osadzać ich kod na stronach HTML. Można też różnicować wygląd i zachowanie się stron dla różnych użytkowników oraz zarządzać połączeniami z bazami danych w trzeciej warstwie aplikacji.



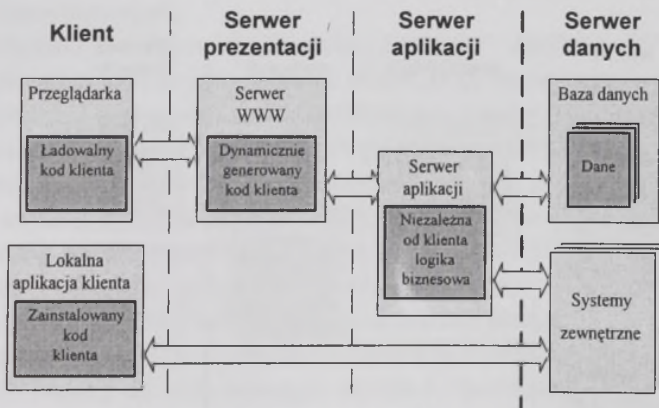
**Baza danych.** Trzecia warstwą bazodanowych aplikacji WWW stanowią bazy danych łącznie z systemami zarządzania tymi bazami. Trzeba zatem dysponować mechanizmami projektowo-implementacyjnymi do realizacji dynamicznego dostępu do danych w bazach oraz administrowania tymi bazami.

## 2. Kierunek rozwoju aplikacji internetowych

Współczesne wymagania stawiane aplikacjom i systemom bazodanowym budowanym i udostępnianym w środowiskach intranetowych i w Internecie, wytyczają ich rozwój w kierunku architektury wielowarstwowej. Architekturę tą można nazwać uogólnioną platformą internetową. Naszkicowano ją na rysunku 2.

Zauważmy, że na schemacie tym mamy warstwę, w której umieszczony jest dodatkowy serwer - serwer prezentacji świadczący usługi na rzecz warstwy klienta oraz warstwy aplikacji. Z tego powodu w warstwie klienta występują dwa rodzaje tzw. „cienkich” klientów. „Cienki” klient działa w oparciu o przeglądarkę HTML i realizuje dostęp z dowolnego sieciowego miejsca i w dowolnym czasie. Funkcjonalnością „cienkiego” klienta jest, w zasadzie jedynie, wyświetlanie interfejsu GUI (skr. ang. *Graphics User Interface*) użytkownika. Jest to pierwszy rodzaj „cienkiego” klienta.

Złożone GUI można realizować w przeglądarce za pomocą dodatkowego kodu aplikacji klienta wytworzonego z komponentów (ang. *plug-in*) przeglądarki albo za pomocą zbudowanych dodatkowo dla przeglądarki skryptów. Można też zainstalować część kodu aplikacji na stacji roboczej klienta i uruchamiać go poza przeglądarką. Mamy wtedy drugi rodzaj klienta - „grubego” klienta.



Rys. 2. Schemat ogólnej platformy internetowej

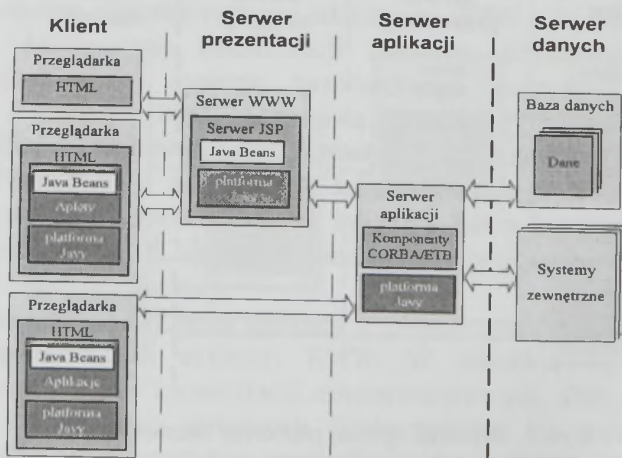
Warstwa serwera prezentacyjnego zawiera serwery WWW, które obsługują polecenia HTTP i generują dynamicznie kod prezentacji do wykonania i

wyświetlenia na kliencie w formie stron *HTTP*. W warstwie tej mogą być zaimplementowane i skonfigurowane wymagania na generowanie kodu udostępniającego *GUI* prezentacji na klientach, np. na podstawie jego identyfikatora *ID*, roli w systemie bazodanowym lub informacyjnym, przynależności do grupy użytkowników.

W warstwie serwerów aplikacyjnych alokuje się logikę biznesową z kodem operowania na danych w jednej lub kilku bazach danych lub bazie rozproszonej sieciowo. Dostęp do usług tej bazy jest wielokrotny i możliwy dla wielu klientów i wielu aplikacji współbieżnie albo nawet równoległe - wieloprosesowo a często wielowątkowo. Dąży się do tego, aby logika biznesowa była też łatwo dostępna zewnętrznym serwerów i zewnętrznym klientów, co jest istotne przy dokonywaniu tzw. integracji aplikacji. Często w warstwie logiki biznesowej występują również specjalizowane serwery raportów i analiz.

### 3. Standard projektowo-implementacyjny trójwarstwowych systemów bazodanowych

Idea platformy internetowej pokazanej na rysunku 2 ma swój standardowy odpowiednik implementacyjny, który jest jej programistycznym uszczegółowieniem. Odpowiednikiem tym jest tzw. *Java 2 Platform*. Jest to model budowy aplikacji korporacyjnych (ang. *Enterprise Edition Application Programming Model*). Model ten został wyspecyfikowany przez firmę Sun Microsystems i jest przeznaczony dla projektantów i programistów programujących bazodanowe systemy i aplikacje internetowe i intranetowe w języku *Java*. Model ten pokazano na rysunku 3. Jego precyzyjną specyfikację można znaleźć w Internecie [31].



Rys. 3. Model Java 2 Platform do budowy aplikacji korporacyjnych

Z braku miejsca nie będziemy tego modelu omawiać dokładniej. Zauważmy tylko, że istnieje możliwość zaimplementowania aplikacji tzw. „grubego” klienta, wykorzystując np. biblioteki komponentów wielokrotnego użytku *Java Beans*. Aplikacje takie są zwykle bogate funkcjonalnie i wydajne oraz oprogramowane w całości w języku *Java*. Mogą być zarówno apletami rozszerzającymi możliwości *HTML* w przeglądarce opartej na *Javie* jak i niezależnymi aplikacjami na stacjach roboczych. „Grubi” klienci mogą się przyłączać albo do serwerów aplikacyjnych za pomocą specjalizowanego protokołu (ang. *IIOF*), albo bezpośrednio do serwerów danych za pomocą interfejsów *JDBC* (skr. ang. *Java DataBase Connectivity*).

Warto też dodać, że jadowe serwery aplikacyjne zawierają biznesowe komponenty rozproszone wielokrotnego użytku. Komponenty te znajdują się w *Java CORBA Object Request Broker* (skr. *ORB*) lub na serwerze *Enterprise JavaBean* (skr. *EJB*).

#### 4. Technologie dostępu do baz danych w sieciach intranetowych i w Interencie

Bardzo ważnym zastosowaniem Internetu i intranetów jest udostępnianie baz danych poprzez *WWW*. *WWW* stanowi uniwersalny interfejs komunikowania się użytkowników i administratorów z bazami danych. Przeglądarki internetowe realizują jedynie wyświetlanie danych z baz. Konieczne jest zatem oprogramowanie pośredniczące do łączenia się z bazami i realizacji odczytu danych. Pośredniczące mechanizmy dostępowe tworzą strony *WWW* współpracujące z bazami. Mechanizm tworzenia stron umożliwia wprowadzanie danych, upraszcza zadawanie zapytań do baz oraz wyświetla odpowiedzi z serwerów bazodanowych.

Składniki interfejsu użytkownika „czystego” standardu *HTML* ograniczają się do możliwości wybierania spośród widocznych na stronie *WWW* odnośników hipertekstowych i tym samym poruszania się po „pajęczynie” dokumentów. Każda wychodząca poza te ramy funkcja wymaga odpowiedniego programu, który musi być zainstalowany po stronie komputera-serwera. Do realizacji takiego sposobu interakcji użytkownika ze stronami *WWW* stosuje się protokół *HTTP*, którego poważną wadą jest brak możliwości zapamiętywania informacji o wcześniejszych wywołaniach - każde polecenie wykonywane jest niezależnie od wyniku poprzednich. Nie można tworzyć witryn z interaktywnym sposobem modyfikacji publikowanych informacji. Opracowano nowe technologie, które zlikwidowały tę wadę *HTTP*. Należy do nich zaliczyć: standard i technologię *CGI*, standard *ISAPI* oraz jadowy interfejs *JDBC*, *ActiveX*, *ASP* oraz *cookies*. Poniżej scharakteryzujemy trzy pierwsze z nich, najdłużej i najczęściej stosowane. Wspomnimy również o rzadziej stosowanych, ale interesujących technologicznie



rozwiązaniach do budowy interaktywnych aplikacji internetowego i intranetowego dostępu do baz danych.

Do pewnego czasu technologia i standard *CGI* (skr. ang. *Common Gateway Interface*) był jedynym sposobem komunikacji z bazami danych poprzez Internet i w intranetach. Mechanizmy *CGI* dostarczyły statycznym stronom *HTML* elementów interaktywności. W *CGI* zawarto także usługi interfejsu pośredniczącego między przeglądarkami a bazami danych. Stosowanie takiego interfejsu ma jednak znaczące wady. Powstało już szereg innych, ulepszonych technologii i wciąż powstają nowe. Posiadają też one coraz większe możliwości.

Na początek przedstawimy ogólny model aplikacji sieciowych, na którym opierają się współczesne zaawansowane systemy z przeglądarkami internetowymi jako standardowym interfejsem użytkownika.

**Technologia i standard CGI.** *CGI* jest najstarszym technologicznie i historycznie interfejsem umożliwiającym uruchamianie programów na serwerze *WWW* w odpowiedzi na dane pochodzące z przeglądarki. Interakcję użytkownika ze stronami *WWW*, np. wyszukiwanie rekordów w bazach danych alokowanych w środowisku intranetowym lub w Internecie umożliwiają skrypty *CGI*. Skrypt *CGI* działa na serwerze *WWW*. Dane uruchomieniowe skryptu pochodzą z przeglądarki. Akcje wykonawcze są realizowane przez serwery baz danych. Rolę skryptów z serwera mogą też pełnić programy skompilowane, pliki wsadowe, inne pliki wykonywalne, a nawet systemy zarządzania bazami danych na dowolną platformę systemową lub z każdego środowiska programistycznego, które tutaj nazywa się również skryptami.

Skrypty te mogą pełnić dwie funkcje:

- (1) odbierać dane przesłane przez przeglądarkę *WWW* klienta do serwera i np. zapisać przetworzone dane na dysku twardym serwera,
- (2) wysłać dane do przeglądarki nadając im formę np. strony w *HTML*. *CGI* posiada wiele zalet i dlatego wciąż jest stosowane do realizacji serwisów *WWW*. Najważniejsze z zalet to:
  - (a) dostarczanie interaktywności serwisom internetowym przez możliwość zadawania zapytań i uzyskiwanie szybkiej odpowiedzi,
  - (b) łatwość tworzenia aplikacji - czasami bardzo złożonych, np. przeglądarek i sklepów internetowych,
  - (c) duże możliwości realizowania dostępu do baz danych praktycznie z każdym serwerem bazodanowym,
  - (d) duża niezależność od platform operacyjnych, sieciowych i sprzętowych - skrypty można uruchamiać na większości serwerów *WWW*, a w razie ograniczeń serwerowych można zastosować rozszerzenia, np. *PHP* lub *Cold Fusion* [39],
  - (e) skrypty można pisać w każdym języku programowania operującym na standardowym wejściu/wyjściu.

Niestety, *CGI* posiada też istotne wady. Tworzenie bardzo złożonych systemów za pomocą skryptów jest trudne. Są one niewydajne oraz trudno zapewnić właściwą synchronizację miedzymodułową skryptów, ponieważ każdy

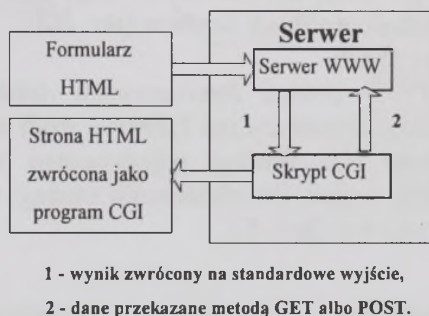


ze skryptów działa jako samodzielny proces z własną przestrzenią adresową i zasobami. Stosowanie skryptów tworzy też luki w zabezpieczeniach maszyn serwerowych.

Poza tym jedyną możliwością wymiany danych między aplikacjami i modułami jest metoda wymiany poprzez bazy danych albo zmienne środowiskowe. Brak jest standardowej metody wywoływania metod z innych aplikacji, co bardzo ogranicza zastosowanie CGI w nowoczesnych systemach obiektowych.

Wady te usunięto w nowszych technologicznie interfejsach API (skr. ang. *Application Programming Interface*) dla serwera WWW. Aplikacje korzystające z API uruchamiane są także po stronie serwera, ale są bardziej wydajne i bezpieczniejsze od aplikacji CGI. Najbardziej znane API to ISAPI z serwera firmy Microsoft oraz NSAPI z serwera firmy Netscape.

Działanie internetowej aplikacji CGI przedstawiono na rysunku 4.



Rys. 4. Działanie internetowej aplikacji CGI

Widać, że skrypty są wywoływane przez serwer na podstawie informacji z przeglądarki. Skrypt do wykonania na serwerze jest wskazywany przez adres URL. Adres ten może pojawić się wszędzie, gdzie można użyć „zwykłego” URL, np. w polu ACTION formularza, w odnośniku lub obrazku. Za jego pomocą przeglądarka kontaktuje się z serwerem.

Bardzo łatwo jest napisać prostą aplikację CGI, np. w języku Perl, która wyświetla na podstawie podanego identyfikatora pewne informacje z bazy danych. W tym celu należy wykorzystać dwa dostępne w języku Perl moduły - CGI do obsługi interaktywnych aplikacji oraz DBI (skr. ang. *Data Base Interface*) [14,25,26] implementujący uogólniony interfejs do wielu systemów zarządzania bazami danych ukrywający szczegóły implementacyjne procesu łączenia się z systemem bazodanowym, zadawania zapytań, odbierania wyników i modyfikacji bazy danych.

**Standard ISAPI.** Cechą charakterystyczną technologii ISAPI (skr. ang. *Internet Server Application Programming Interface*) firmy Microsoft jest mocna integracja rozszerzeń interfejsu użytkownika aplikacji z serwerem WWW. Interaktywna aplikacja z interfejsem strony WWW oraz pozostałe elementy

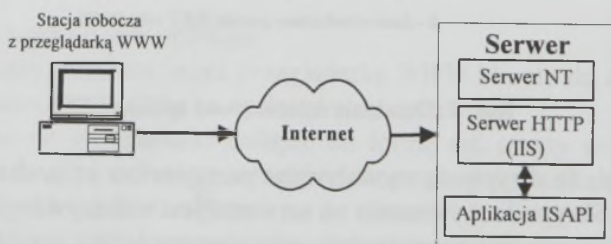
rozszerzenia *API* pozostają w pamięci od chwili pierwszego wywołania. Zasoby serwera pozostają zainicjowane i gotowe do wykorzystania przez kolejne żądania aplikacji. Wydajność takiej komunikacji i przetwarzania znacznie się zwiększa. *ISAPI* jest powszechnie stosowany na platformach systemowych *MS Windows*. W systemach *Windows NT/2000* można wykorzystywać zawierający *ISAPI* serwer *IIS* (skr. ang. *Internet Information Server*), a w *Windows 9x* serwer *PWS* (skr. ang. *Personal Web Server*). Rozszerzenia z *IIS* mają większe możliwości, np. zwiększone bezpieczeństwo serwera i kontrolę uprawnień do skryptów i rozszerzeń *API*.

*ISAPI* jest rozszerzeniem niskopoziomowym *API*. Ograniczona jest liczba działających instancji *ISAPI*, zatem mamy zwiększoną wydajność serwera. Aplikacje wykonują się w obszarze adresowym maszyny-serwera i dlatego lepiej są wykorzystywane zasoby serwera *WWW*.

Biblioteki *DDL* na potrzeby *ISAPI* można tworzyć dwoma metodami:

- (1) tworzyć *Internetowe Aplikacji Serwera* (skr. *ISA*),
- (2) pisać tzw. *Filtry ISAPI*.

Rozszerzenie *ISAPI* w postaci *Internetowych Aplikacji Serwera* realizuje biblioteka *DLL* (skr. ang. *Dynamic Link Library*), czyli standardowy mechanizm dla platformy *Windows*. Do obsługi pojedynczego żądania z przeglądarki powoływany jest osobny wątek. Po zakończeniu obsługi żądania zakańczany jest wątek obsługi wielowątkowej (Rys. 5.).



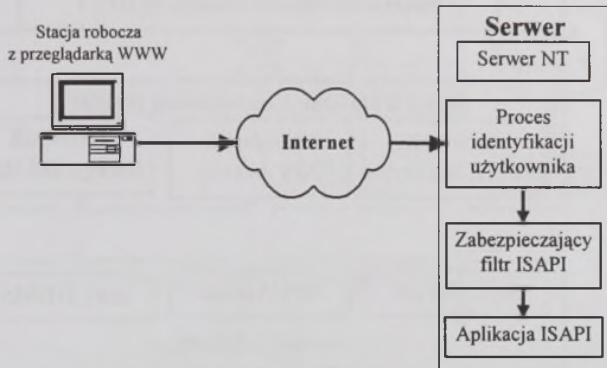
Rys. 5. Położenie rozszerzenia *ISA* w procesie działania aplikacji *WWW*

Biblioteka tworząca *ISA* eksportuje trzy funkcje: udostępniania serwerowi numeru rozszerzenia, wykonywania czynności łagodnie kończących pracę wątków i biblioteki oraz przetwarzania żądań klienta (stacji roboczej z przeglądarką). Żądania klienta natomiast są realizowane z wykorzystaniem czterech funkcji, których adresy są przekazywane w rekordzie, z jakim serwer przy każdym żądaniu woła funkcję przetwarzania żądań. Te cztery funkcje odpowiadają akcjom na standardowym wejściu/wyjściu i pobieraniu zawartości zmiennych środowiskowych w aplikacjach *CGI*.

Specyficzny sposób rozszerzenia możliwości serwera oferują *Filtry ISAPI*. Są one również bibliotekami *DLL*, ale są wywoływane nie tylko w odpowiedzi na żądania przeglądarki (klienta), lecz w odpowiedzi na każdą akcję serwera. Jest to

zatem specjalny sposób rozszerzenia możliwości serwera. Dzięki temu istnieje możliwość przetwarzania zawartości przygotowywanych stron witryny WWW. Filtry przechwytyją żądania niezależnie od tego czy jest ono związane z plikiem, skryptem CGI czy też aplikacją ISAPI.

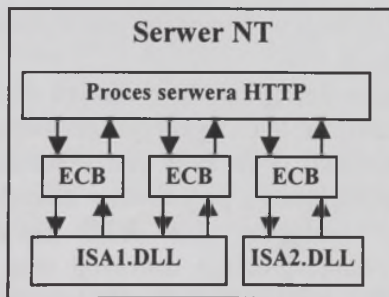
Filtry ISAPI można zatem wykorzystywać do logowania użytkowników, autoryzacji dostępu, kompresji i dekompresji oraz kodowania informacji, badania wykorzystania poszczególnych fragmentów witryny WWW i wielu innych zastosowań internetowych i intranetowych. Jedno z wymienionych zastosowań pokazano na rysunku 6.



Rys. 6. Klasyczny przykład zastosowania filtra ISAPI w procesie autoryzacji

Podobnie jak biblioteki ISA, filtry są realizowane za pomocą zestawu pewnych funkcji. Filtry ISAPI wykorzystuje się do obsługi języków osadzonych na stronach WWW - np. dla potrzeb interpreterów ASP lub PHP.

Komunikacja pomiędzy serwerem WWW i aplikacją ISA (ISAPI) odbywa się przez strukturę bloków ECB (skr. ang. *Extension Control Block*) i wymianę informacji za pomocą zmiennych środowiskowych serwera oraz standardowego wejścia/wyjścia (Rys. 7.).

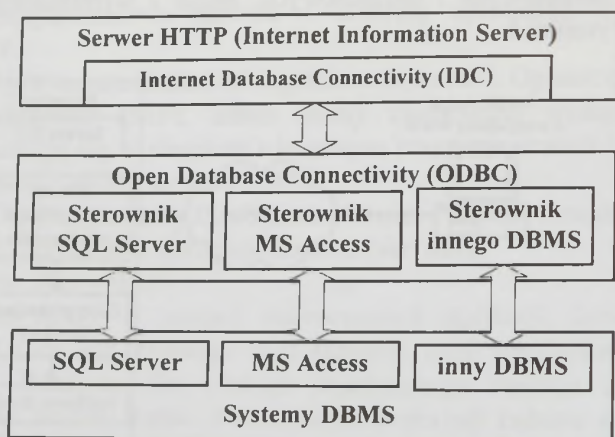


Rys. 7. Bloki ECB w procesie uruchamiania programów ISAPI



Aplikacja *ISAPI* musi być wielowątkowa, ponieważ w tym samym czasie musi obsłużyć wiele żądań.

**Konektory IDC.** Jednym z ważnych przykładów zastosowania *Filtrów ISAPI* jest wbudowany do *IIS* mechanizm dostępu do baz danych w formie tzw. konektorów *IDC* (skr. ang. *Internet Database Connector*) [37]. Konektory to biblioteki *DLL ISAPI* tworzące kanał komunikacyjny pomiędzy serwerem *IIS* a bazami danych.



Rys. 8. Rola IDC w procesie dostępu do baz danych ze stron WWW

*IDC* jest częścią interfejsu programistycznego *ISAPI* używającego do komunikacji z bazami danych interfejsu programowego *ODBC* (skr. ang. *Open Database Connectivity*). *IDC* zawiera mechanizm bezpośredniego wiązania pól formularzy *HTML* z polami baz danych bez konieczności używania dodatkowych skryptów *CGI*.

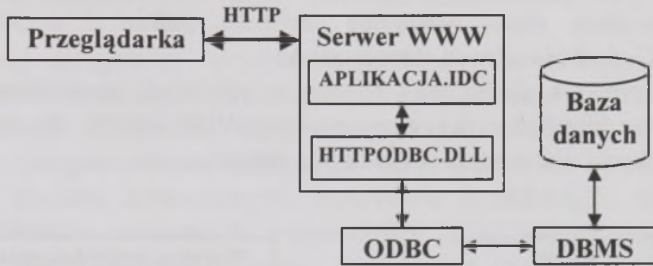
Za pomocą *IDC* można: publikować informacje w trybie „on-line”, administrować bazami danych, wydajnie wykonywać zapytania, budować internetowe i intranetowe aplikacje bazodanowe z dostępem w trybie bezpośrednim.

Dostęp do baz danych wykonywany jest z poziomu serwera *HTTP* serwera *IIS*, co widać na rysunku 8.

Do kontrolowania dostępu do bazy danych oraz prezentowania informacji w przeglądarce użytkownika *IDC* wykorzystuje dwa rodzaje plików *\*.idc* oraz *\*.htx* (skr. *HTML eXtension*). Plik *HTX* jest szablonem dla dokumentu *HTML* wyświetlanego przez przeglądarkę po pobraniu danych z bazy i połączeniu ich z przy użyciu *IDC* z szablonem. Plik *HTX* zawiera znaczniki określające formatowanie, wykonywanie prostych instrukcji oraz umiejscowienie danych i powiązań z plikiem *IDC*. Aplikacja *IDC* odwołuje się do plików programowych



(\**.idc*) na serwerze WWW poprzez adres URL. Szkic metody IDC pokazano na rysunku 9.



Rys. 9. Schemat działania aplikacji IDC

Metoda *IDC* jest prostym i łatwym w użyciu sposobem dostępu do baz danych poprzez *WWW*. Do programowania takiego dostępu wystarczy znać język *HTML*.

**Interfejs JDBC.** *JDBC* (skr. ang. *Java DataBase Connectivity*) [35,36] jest bazodanowym, „javowym” interfejsem *API*. Jest on zestawem metod umożliwiających funkcjonalny dostęp do zdalnych baz danych w oparciu o język zapytań *SQL* (ang. *Structured Query Language*). *JDBC* zawarto w pakiecie programistyczno-projektowym *Java Enterprise API*. *JDBC* rozszerza możliwości języka *Java* tworzenia aplikacji działających w środowiskach rozproszonych o dobrze zdefiniowany standard interfejsu programistycznego realizowania dostępu do baz danych alokowanych w środowiskach intranetów i w Internecie niezależnie od rodzaju systemu zarządzania bazami danych. *JDBC* można zastosować do budowy systemów w architekturze klient-serwer i trójwarstwowej.

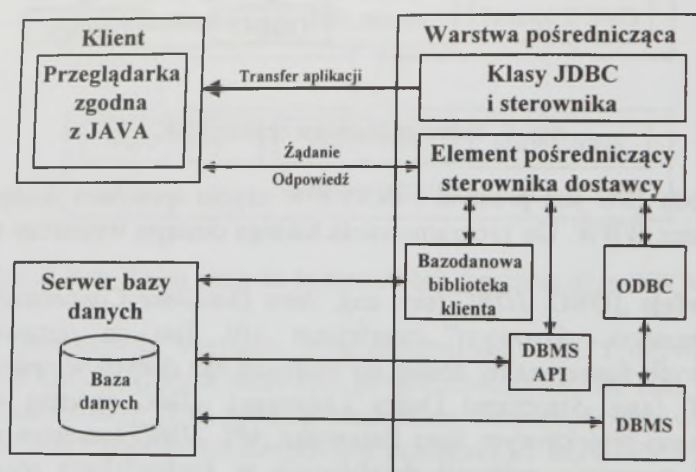
*JDBC* to zestaw klas *API* do obsługi bazy danych dla aplikacji i appletów napisanych w *Javie*. Oprogramowanie jest niezależne od platform sprzętowych i operacyjnych a funkcjonalnie jest podobne do istniejących już technologii i standardów, tzn. *X/OPEN SQL Call Level Interface* (skr. *CLI*) oraz *Microsoft Open DataBase Connectivity*. Funkcjonalnie *JDBC* jest bardzo podobny do *ODBC* i komunikuje się z bazami danych również na poziomie zapytań języka *SQL*. Sterownik bazy danych w *JDBC* jest zgodny ze standardem *ANSI SQL92 Entry Level*.

Interfejs *JDBC* włącza się na poziomie programistycznym przez dołączenie pakietu *java.sql* zawierającego interfejs dla programistów aplikacji i dla programistów sterowników. Obecnie są w użyciu dwie wersje: *JDBC 1.2* opracowana przez firmę *JavaSoft* oraz *JDBC 2.0* opracowana przez firmę *Sun*

Microsystems. Interfejs programowania aplikacji bazodanowych to zbiór interfejsów *Javy* pozwalający łączyć się z bazami, wykonywać zapytania w języku *SQL* i przetwarzać wyniki.

**Połączenia JDBC.** Istnieją cztery rozwiązania implementacyjne sterownika *JDBC* – typ *I*, *II*, *III*, *IV*. Z braku miejsca nieco więcej uwagi poświęcimy tylko sterownikowi typu *III*. Zastosowanie każdego ze sterowników tworzy połączenia nieco odmiennie architektonicznie i o zróżnicowanych możliwościach funkcjonalnych. Mamy zatem:

(1) sterownik „*JDBC* typu *I*”. Jest to sterownik opracowany przez firmy *JavaSoft* oraz *InterSolve* nazywany mostem *JDBC-ODBC*. Stosujemy go jeśli aplikacja ma korzystać z wielu różnych baz danych;



Rys. 10. Trójwarstwowy model dostępu przez protokół sieciowy - typ III

(2) sterownik „*JDBC* typu *II*”. W tym rozwiązaniu funkcjonalno-architektonicznym stosuje się rodzime *API* dostawców baz danych. Sterownik jest tylko częściowo zaimplementowany w *Javie*. W innych językach należy oprogramować bibliotekę tłumaczącą zapytania *JDBC* na zapytania bazodanowe;

(3) sterownik „*JDBC* typu *III*”. Tu stosuje się protokół sieciowy. Sterownik ten jest napisany całkowicie w języku *Java*. Istnieje warstwa pośrednicząca tłumacząca język protokołu sieciowego na zapytanie w konkretnym języku baz danych. To rozwiązanie architektoniczno-funkcjonalne do połączeń *JDBC* przedstawiono na rysunku 10. Widać, że model pokrywa się z architekturą trójwarstwową aplikacji i systemów bazodanowych w przeciwieństwie do pozostałych modeli *JDBC*, przystających do dwuwarstwowej architektury klient-serwer.

(4) sterownik „*JDBC* typu *IV*”. Jest podobny jest on do typu *II*, z tym że

nie używa się tu innych języków i API dostawców baz danych. Jest on całkowicie napisany w *Javie*.

Najnowszą wersją specyfikacji *JDBC API* jest wersja 2.0. Opublikowała ją firma Sun Microsystems 30.05.2000 [35].

## 5. Technologie i narzędzia programistyczne do prototypowania i budowy aplikacji wielowarstwowych i sieciowo zorientowanych

W celu zastąpienia oprogramowania klientów w rozproszonych lub scentralizowanych aplikacjach bazodanowych przeglądarkami *WWW* konieczne jest wykorzystanie możliwości funkcjonalnych odpowiednich serwerów oraz zastosowanie oprogramowania scalającego bazy danych z internetową siecią rozległą albo sieciami intranetowymi. Umożliwia to uniknięcie rozwiązywania licznych problemów związanych z tworzeniem klasycznego oprogramowania klient-serwer oraz wieloplatformowy dostęp do bazy danych. Ponieważ wykorzystuje się przy tym standardowe rozwiązania sieciowe, nieistotne jest na jakiej platformie sprzętowej i w jakim środowisku operacyjnym działa oprogramowanie aplikacji systemu bazodanowego.

Już obecnie istnieje dość dużo produktów w postaci obszernych i dość kompletnych pakietów oprogramowania integrujących bazy danych z sieciami intranetowymi i Internetem (źródła literaturowe i internetowe szacują ich ilość na ok. 50).

Najbardziej znane są pakiety oprogramowania integrującego oferowane przez duże firmy, np. Microsoft [7,12], Borland [36,40], Netscape [38] albo Oracle [32,33,47]. Kilka z nich omówimy niżej. Oprogramowanie do integracji baz danych dostarczają również mniejsze firmy, np. NetDynamics [40,42] (pakiet *NetDynamics*), Allaire [39] (oprogramowanie *ColdFusion*) oraz szereg firm małych i mniej znanych.

Daje się zauważyć spore różnice w budowie oferowanych produktów oprogramowania. W dalszej części tej pracy postaramy się je uchwycić.

W praktyce projektowej, programistycznej i projektowej wykorzystuje się wiele pakietów i środowisk prototypowania i wytwarzania aplikacji wielowarstwowych i sieciowo zintegrowanych baz danych. Trudno jednoznacznie stwierdzić, które z nich są najbardziej przydatne, wygodne i wydajne.

**Serwer ASP.** *Activer Server Pages* (skr. *ASP*) jest technologią opartą o skryptowe środowisko programistyczne zintegrowane z pakietem oprogramowania serwerowego *IIS* firmy Microsoft. *ASP* wykorzystuje się zarówno do tworzenia prostych dynamicznych stron *WWW*, jak i dużych interaktywnych aplikacji bazodanowych. Strony aplikacji *ASP* powstają jako mieszanka elementów języka *HTML*, tekstu i komend języka skryptowego. Wykonywane są w całości na serwerze *WWW* (tzw. *server-side scripting*). Są one aktywowane przez kliknięcie łącznika w przeglądarce. Serwer *WWW* przetwarza polecenia skryptu wysyłając do przeglądarki gotową stronę zawierającą jedynie format *HTML*. Przeglądarka nie



bierze udziału w przetwarzaniu i stacja robocza jest odciążona. Zastosowanie w skryptach mechanizmów *ActiveX* pozwala na sprawną komunikację z bazami danych i wykonywanie zadań wymagających dużej mocy obliczeniowej i centralnego przetwarzania.

*ASP* oparte jest na standardzie *ActiveX Scripting*, który nie definiuje konkretnej składni języka. Można tworzyć strony WWW np. w *Visual Basic* lub *Javie* ponieważ w *ASP* są dostępne ich interpretery (*VBScript* i *JScript*). Można też używać innych składni skryptowych, np. *Perl* lub *REXX*. Natomiast należy znać język *HTML*, ponieważ jest on nierozdzielnie związany z interfejsem przeglądarkowym i trzeba umieć tworzyć strony WWW w sposób standardowy.

Tworzenie aplikacji *ASP* polega na przygotowywaniu plików *.asp*, zawierających kompozycję *HTML* i kodu skryptowego (standardowo *VBScript*). Przeglądarka wysyła żądanie pobrania pliku *.asp*, serwer *IIS* przekazuje je do biblioteki *asp.dll* w celu przetworzenia. Zawarty w niej mechanizm powoduje wczytanie zawartości tego pliku jej analizę i wykonanie zawartych tam takich poleceń, jak połączenie z bazą danych, odczytaniem lub modyfikacją danych. Wygenerowany kod wynikowy *HTML* jest przekazywany z powrotem do *IIS*, który zwraca go do przeglądarki.

*ASP* jest częścią serwerową modelu *Active Platform* firmy Microsoft z technologiami: *HTML*, *VBScript* i *JScript*, komponentów (*ActiveX* i *Java*), usług systemowych (np. katalogowych, audio i video). Środowisko do programowania aplikacji *ASP* składa się z obiektów *Request*, *Response*, *Application*, *Session* oraz *Server* z odpowiednimi właściwościami, obsługą zdarzeń i metodami. Służą one do określania stanu i kontroli aplikacji. Są one dostępne z poziomu strony WWW.

Przykłady prostych bazodanowych aplikacji *ASP* można znaleźć w [11,19,27,50].

**Novell Script for NetWare.** W skład *Novell Script for NetWare* firmy Novell wchodzi trzy technologie: *Novell Script for NetWare* (skr. *NSN*), *Novell Script Pages* (skr. *NSP*) oraz *Novell Data Object* (skr. *NDO*). Umożliwiają one tworzenie dynamicznych stron WWW i operowanie z nich na bazach danych. *NSN* obsługuje język *VBScript* (stąd nazwa). *VBScript* natomiast jest częścią *Visual Basic* i jest używany do tworzenia makrodefinicji i funkcji rozszerzających działanie aplikacji *Microsoft Office* oraz jako serwerowy język skryptowy WWW. *VBScript* jest używany wraz z natywną technologią *ASP* firmy Novell.

Technologie *NSN* pozwoliły dodać do platformy *NetWare* sprawną technikę tworzenia skryptów WWW i dodawania nowych funkcji do systemu *NetWare*. *NSN* zawiera zestaw komponentów *UCX* (skr. ang. *Universal Component Extension*), realizujących dostęp do różnych usług systemu *NetWare*, np. *NDS* (skr. ang. *Novell Domain Services*).

**Novell Script Pages.** Za pomocą łączenia skryptów *NSN* i *HTML* można wytworzyć wydajne aplikacje WWW. Tą możliwość wykorzystuje się w *Novell Script Pages* (skr. *NSP*). Mamy tu do dyspozycji pięć obiektów: *Session*, *Application*, *Request*, *Server*, *Response*. Stosując je w skrypcie można wykorzystać usługi serwera WWW. Można np. zapewnić zapisywanie



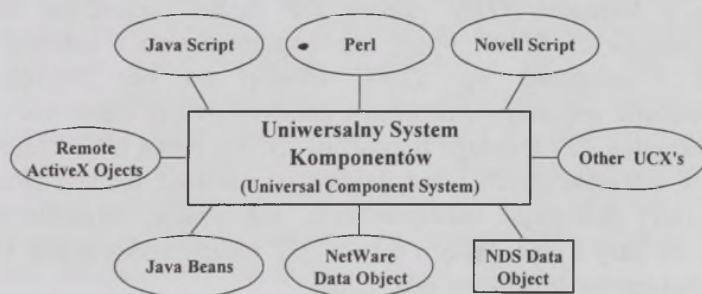
i odczytywanie różnych informacji dotyczących aplikacji WWW. Niektóre z tych akcji są bardzo ekspresywne, np. wysyłanie, otrzymywanie i zarządzanie znacznikami kontekstu klienta (ang. *cookies*). Diagram obiektów składowych *NSP* oraz zwięzły opis ich funkcjonalności można znaleźć w [23] oraz dokumentacji firmowej.

Za pomocą systemu tworzenia aplikacji WWW - np. standardowego pakietu *HTML Microsoft FrontPage 98/2000* można tworzyć statyczne części aplikacji *NSP*. Dynamikę aplikacji dodaje się przez wstawianie w skrypcie etykiety skryptowych kodu klienta i kodu *NSN*.

**NetWare Data Object.** *NetWare Data Object* (skr. *NDO*) jest trzecią z technologii firmy Novell do tworzenia dynamicznych stron WWW powiązanych z bazami danych. Wiązanie kodu skryptu z bazą danych zapewnia tu wspólny obiektowy interfejs dla różnych środowisk bazodanowych i systemów zarządzania bazami danych. *NDO* umożliwia łączenie stron z bazami *Oracle 7*, *Oracle 8*, *SQL Integrator* oraz bazami z dostępem zgodnym z *ODBC* a pracującymi pod kontrolą systemu *NetWare* lub na zdalnej stacji *Windows*.

Interfejs *NDO* zaimplementowano w modelu złożonym z siedmiu obiektów. Realizują one komplet funkcjonalności połączeń z bazami, wykonywania zapytań do baz, odczytywania atrybutów silników baz danych i atrybutów schematów baz [23,49]. *NDO* jest zaimplementowany, tak by był składnikiem rozszerzenia *UCX*. Można go zatem wywoływać w szerszym technologicznie systemie *UCS* (skr. ang. *Universal Component System*). Technologia *UCS* umożliwia używanie obiektów *NDO* przez różne języki skryptowe, np. *Perl* i *JavaScript*, w tym *NSN* (Rys. 11.).

Trzy omówione powyżej technologie składowe tworzą środowisko projektowo-implementacyjne do tworzenia internetowych i intranetowych systemów bazodanowych na platformę sieciową *Novell NetWare* oraz umożliwiają tworzenie wydajnych aplikacji bazodanowych w środowisku WWW. Można też przenieść za ich pomocą wiele istniejących aplikacji bazodanowych i różnych aplikacji pracujących w innych architekturach, do środowiska WWW. W ten sposób można realizować ideę skalowalności oraz znacząco zwiększać wydajność i dostępność tych systemów. *NSN*, *NSP* oraz *NDO* są dostępne za darmo na stronie internetowej poświęconej oprogramowaniu *Novell DeveloperNet* [23].



Rys. 11. Technologia UCS obiektowego interfejsu do dołączania się do baz danych z poziomu różnych języków i różnego oprogramowania dostępowego

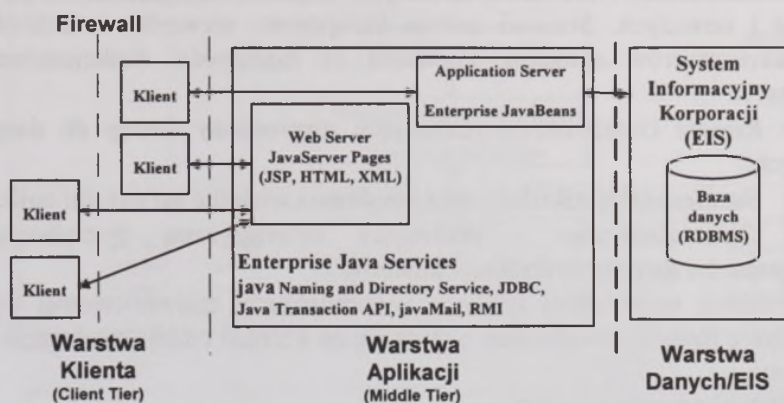
**Java Server Pages.** *Java Server Pages* (skr. *JSP*) jest technologią tworzenia dynamicznej zawartości stron WWW. Przyjęto w niej, że przetwarzanie realizuje warstwa pośrednia (*Middle Tier*). Dzięki temu proces tworzenia stron WWW jest uproszczony, bo logika aplikacji jest od nich oddzielona. Logika aplikacji jest enkapsulowana w przenośnych i komponentach *Javy* wielokrotnego użytku.

*JSP* jest rozwinięciem technologii tzw. serwletów. Technologia serwletów bazuje na języku *Java* i wspomaga tworzenie aplikacji po stronie serwera. *JSP* szybko się rozwija i coraz bardziej ułatwia proces budowy, dystrybucji i konserwacji aplikacji serwerowych komunikujących się z przeglądarkami WWW jako klientami sieciowymi. *JSP* zaimplementowano jako *Java API*, które jest częścią pakietu programistycznego *Java 2 Platform Enterprise Edition* (skr. *J2EE*). W tej technologii tworzone strony są kompilowane do postaci serwletów. Technologia *JSP* zawiera efektywną metodologię tworzenia i utrzymania aplikacji, gdyż rozdziela projekt i statyczną zawartość strony od logiki aplikacji generującej dynamikę tych stron. Do korzystania z *JSP* nie trzeba posługiwać się językami skryptowymi. Logika aplikacji jest automatycznie umieszczana na standardowej stronie *HTML* lub *XML*.

Wykorzystując *JSP* można skupić się na budowaniu komponent wywoływanych ze stron WWW. Komponenty stron można budować za pomocą komponent *JavaBeans* albo bibliotek *JSP*. Inną zaletą jest to, że zbudowane komponenty można używać wielokrotnie do budowy innych aplikacji. Technologia *JSP* pozwala stworzyć „kod logiki aplikacji nie zawierający treści stron oraz strony nie zawierające kodu logiki”.

*JSP* łatwo zastosować do budowy aplikacji i systemów o architekturze wielowarstwowej. Zilustrowano to na rysunku 12. Widać, że warstwa środkowa jest kompozycją dwóch podwarstw funkcjonalnych. Łatwo zauważyć, że strony zbudowane w technologii *JSP* pełnią rolę serwerowego interfejsu *Middle Tier* dla aplikacji wielowarstwowych. Warstwa ta jest elastycznym i skalowalnym środowiskiem dystrybucji aplikacji w sieciach intranetowych i w Internecie.

Strona *JSP* komunikuje się z klientami sieciowymi, którymi są przeglądarki WWW. Strona zarządza interakcją wysyłając i odbierając żądania i informacje w formacie *HTML*. Strony *JSP* można umieszczać na większości serwerów sieciowych WWW. Mogą one współpracować z różnymi interfejsami *Java API* i usługami, np. *JDBC* (dostęp do baz danych), *JavaMail* (oprogramowanie wspierające tworzenie aplikacji poczty elektronicznej w *Javie*), *Java Transaction API* (obsługa transakcji). Strony mogą też zarządzać dostępem do zasobów informacyjnych i baz danych w warstwie danych (*Back-End Tier*, *Data/EIS Tier*) obsługując zabezpieczenia, autoryzację, integralność transakcji, połączenia do bazy i inne. W tym celu w *JSP* można wykorzystać komponenty z biblioteki *Enterprise JavaBeans* (skr. *EJB*).



Rys. 12. Schemat strukturalno-funkcyjny architektury wielowarstwowej systemów bazodanowych zbudowanych za pomocą JSP

**Borland Delphi Client/Server Suite i technologia MIDAS.** *Borland Delphi* jest oprogramowaniem, które służy do tworzenia aplikacji uniwersalnych oraz bazodanowych pracujących w środowisku systemów *Windows 95/NT*. W skład pakietu wchodzi również narzędzia oraz technologie pozwalające tworzyć systemy relacyjnych baz danych we wszystkich architekturach, tzn. scentralizowanej, z serwerem plików, architekturach klient-serwer – dwuwarstwowej oraz wielowarstwowej. Nie można natomiast tworzyć systemów baz danych w architekturze host-terminal. Dzięki tym możliwościom *Delphi* zaliczana jest czasami do systemów zarządzania rozproszonymi relacyjnymi bazami danych.

Obecnie przedstawimy zwięźle budowę i własności funkcjonalne oprogramowania middleware *MIDAS* (skr. ang. *Muli-tier Distributed Application Services Suite*) zawartego w pakiecie *Delphi*. Za pomocą tego oprogramowania można tworzyć aplikacje wielowarstwowe. Technologia *MIDAS* jest oryginalną koncepcją działu *Open Software* firmy Borland zajmującego się technologiami systemów otwartych. *MIDAS* zawarto w pakiecie *Delphi 3 Client/Server Suite* [10] oraz w jego wersjach, które zawierają *Client/Server Suite*. (np. *Delphi 4 Client/Server* [21]). Oprogramowanie to jest też dołączane do innych pakietów oprogramowania narzędziowego firmy Borland [40], np. do *JBuilder*, *IntraBuilder*, *C++ Builder*. Firma Borland stworzyła również technologię middleware *Entera* [48], której produkty wchodzi w skład m.in. pakietu *Delphi 3 Enterprise*. *Entera* ma więcej funkcjonalności typu middleware oraz wspiera kilka różnych systemów operacyjnych, tzn. *Windows NT*, *Sun Solaris*, *HP-UX*, *IBM AIX*.



**Middleware MIDAS.** MIDAS jest włączony do pakietu *Delphi 3* oraz *Delphi 4* i nowszych. Stanowi zestaw komponent, serwerów i technologii do tworzenia serwerów aplikacji. Podstawą są możliwości funkcjonalne trzech brokerów:

- *Remote DataBroker* - realizujący rozproszony dostęp do danych baz danych,
- *Business ObjectBroker* - do zarządzania wieloma serwerami aplikacji,
- *ConstraintBroker* - realizujący automatyczną dystrybucję reguł integralności danych do aplikacji klientów.

Do utworzenia najprostszej aplikacji w architekturze trójwarstwowej wystarczy użyć jedynie *Remote DataBroker*. Składa się on z trzech oddzielnych części:

- (1) klienta,
- (2) środków komunikacji oraz
- (3) serwera.

Klient znajduje się w aplikacji klienta a serwer w serwerze aplikacji. Część stanowiąca klienta składa się z dwóch komponent *VCL*:

(1a) *ClientDataSet* oraz

(1b) *RemoteServer*, który realizuje nawiązywanie i utrzymuje połączenia z serwerem aplikacji oraz pobiera listę komponent dostarczających metody dla klientów.

Komponenta *ClientDataSet* umożliwia zdalne wywoływanie metod w serwerze aplikacji, dzięki czemu komponent ten może pobierać lub wysyłać dane do serwera.

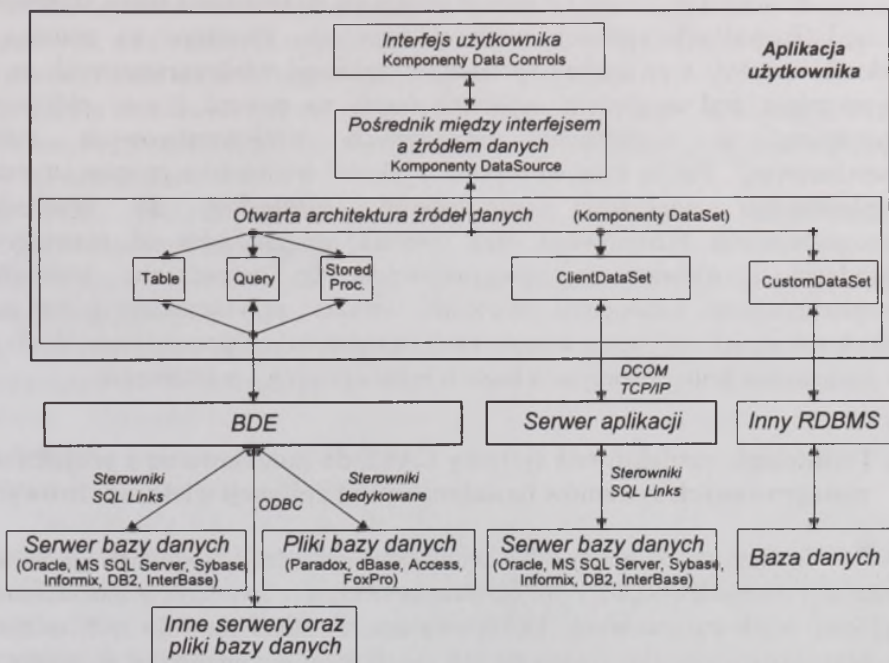
Wywoływanie metod (czyli komunikacja) odbywa się zgodnie ze standardem *DCOM*.

Część *Remote DataBroker* znajdująca się w serwerze aplikacji to *Remote DataModule*, czyli zdalny moduł danych, który zawiera: komponentę *Provider* (dostarczyciela usług), komponenty źródła danych oraz reguły przetwarzania danych. Moduł *Remote DataModule* jest serwerem *DCOM* udostępniającym interfejs dla klientów.

Metody serwera - operując na danych pochodzących z różnych źródeł danych - realizują przetwarzanie danych. Komponenty źródła danych poprzez *BDE* obsługują bazę danych na serwerze bazy lub bazy w postaci lokalnych plików.

Na rysunku 13 przedstawiono komponenty strukturalno-funkcjonalne wymagane do utworzenia w środowisku *Delphi* prostego systemu bazodanowego o architekturze trójwarstwowej. Dla uproszczenia przedstawiono tylko jeden serwer aplikacji, ale pojedyncza aplikacja klienta może również komunikować się z wieloma serwerami aplikacji.





Rys. 13. Komponenty strukturalno-funkcjonalne prostego systemu bazodanowego o architekturze trójwarstwowej w środowisku Delphi

Za pomocą *Business ObjectBroker* można zarządzać aplikacją wielowarstwową oraz równoważyć obciążenia serwerów. *ObjectBroker* zarządza globalnym rejestrem zawierającym spis serwerów aplikacji dostępnych w systemie. W rejestrze tym aplikacja klienta poszukuje serwera oferującego potrzebny komponent lub usługę.

*ConstraintBroker* umożliwia scentralizowanie zarządzania bazami danych. Umożliwia on utworzenie w serwerze aplikacji repozytorium reguł sprawdzania integralności i poprawności danych baz oraz przesyłanie ich do aplikacji klientów. Reguły integralności ładowane są z serwerów baz danych i przesyłane do klienta przy pierwszym połączeniu z serwerem aplikacji.

**Pakiet MS Transaction Server.** Przykładem oprogramowania do implementacji dobrej jakości bazodanowych aplikacji dwuwarstwowych i wielowarstwowych udostępniających bazy danych w intranetach i w Internecie jest pakiet programistyczny *Transaction Server 3.0* firmy Microsoft [13]. Oprogramowanie to posiada stosunkowo duże możliwości funkcjonalne i jest bardzo przydatne do tworzenia infrastruktury systemów bazodanowych ogólnego przeznaczenia. Posiada rozbudowane możliwości implementacji i rozpraszania aplikacji pomiędzy różne platformy serwerów. Oferuje mechanizmy komunikacyjne w standardzie *DCOM*.

*Transaction Server 3.0* dobrze nadaje się do tworzenia łatwo skalowalnych i modyfikowalnych aplikacji wielowarstwowych. Tworzone za pomocą tego pakietu warstwy wewnętrzne (pośrednie) aplikacji wielowarstwowych są mało wymagające pod względem zapotrzebowania na pamięć i moc obliczeniową wymaganą w warstwach zewnętrznych wielowarstwowych aplikacji bazodanowych. Pakiet stanowi wysokiej jakości środowisko programistyczne do implementacji większości infrastruktury niezbędnej do uruchamiania oprogramowania biznesowego oraz uwalnia programistów od rozwiązywania zagadnień i wytwarzania oprogramowania do zarządzania: komunikacją, bezpieczeństwem, katalogami, procesami, wątkami, przyłączaniem do baz danych oraz kontrolą ich spójności w systemach bazodanowych przeznaczonych do pracy w środowisku zintegrowanym w bazach intranetowych i w Internecie.

## **6. Technologie projektowe i systemy CASE do modelowania i projektowania zintegrowanych systemów bazodanowych i aplikacji wielowarstwowych**

Istnieje co najmniej kilka wiodących narzędzi i środowisk CASE nowej generacji do modelowania i projektowania sieciowych systemów bazodanowych i aplikacji wielowarstwowych. Dokonamy analizy stanu rozwoju tych technologii projektowych zwracając uwagę na ich możliwości automatyzacji w generowaniu prototypów lub szkieletów aplikacji intranetowych i internetowych systemów i środowisk bazodanowych oraz schematów scentralizowanych lub rozproszonych baz danych. Z konieczności będzie to analiza bardzo zwięzła.

Metodyki modelowania i projektowania aplikacji wielowarstwowych stosowane w kilku najnowszych, zaawansowanych technologicznie systemach projektowych i systemach CASE nowej generacji, dość istotnie różnią się od klasycznych metodyk projektowych i klasycznych systemów CASE.

Z przeprowadzonych badań studialnych [8,9] wynika, że największe możliwości posiadają cztery następujące systemy projektowe: *Sybase Power Designer 6.0* firmy Powersoft, *Select Systems Engineer* firmy SELECT (rozwiniecie produktu firmy Learmonth & Burchett Management Systems), *Oracle Designer (Designer/2000)* i *Oracle Developer/2000* firmy ORACLE oraz *Select Enterprise* firmy SELECT [29].

W celu porównania ekspresji projektowej tych systemów, zbadano je [4] pod względem: rodzaju modelowania, metodologii modelowania (modeli) procesów biznesowych i funkcji, wykorzystywania paradygmatu obiektowego do modelowania, stosowania modelowania zachowania (dynamiki), metodologii modelowania danych, możliwości modelowania interfejsu użytkownika, ekspresji w normalizacji związków w relacjach bazodanowych, możliwości generowania szkieletowych i szczegółowych baz danych, możliwości generowania reguł na danych, generowania aplikacji łącznie z interfejsem użytkownika, wsparcia dla aplikacji wielowarstwowych i aplikacji klient-serwer, możliwości tworzenia repozytoriów projektów, automatyzacji dokumentowania i raportowania,



możliwości symulowania zachowania projektowanych systemów, wymagań systemowych, wad i zalet, dostępności wersji testowych dla celów badawczych.

Przestudiowano też dość skąpe informacje nt. systemów:

- (1) *System Architect* firmy Popkins Systems & Software do modelowania danych, procesowego i obiektowego (system obsługuje różne metodologie i notacje), oraz
- (2) *Visible Analyst Workbench* firmy Visible Systems do modelowania procesowego i obiektowego.

Bardziej szczegółowe omówienie wyników tych badań wykracza poza wyznaczone ramy niniejszej pracy.

Szczególną uwagę poświęcono narzędziom programistycznym projektowania i wytwarzania systemów dwuwarstwowych klient-serwer i systemów trójwarstwowych przeznaczonych dla sieciowych środowisk intranetowych, ekstranetów integrowanych poprzez Internet oraz systemów bazodanowych udostępnianych w Internecie.

Na potrzeby tej pracy autorzy skoncentrowali się na dość szczegółowym zbadaniu technologii projektowo-implementacyjnych systemów bazodanowych firmy SYBASE (Powersoft). Nieco mniej uwagi poświęcono technologii firmy ORACLE, przy czym z braku miejsca pominięto całkowicie zagadnienia związane z metodykami i narzędziami firmy ORACLE do projektowania hurtowni danych.

Oracleowe technologie projektowo-implementacyjne systemów bazodanowych. Środowiska *Designer/2000* [46] i *Developer/2000* [18] należą do najbardziej zaawansowanego technologicznie oprogramowania projektowo-implementacyjnego do budowania intranetowych systemów informacyjnych i bazodanowych. Proces projektowania i implementacji za pomocą tego oprogramowania jest efektywny i stosunkowo łatwy. W Internecie można znaleźć wersje próbne *Oracle Designer/2000* i *Oracle Developer/2000 (2.0)* [43].

**Oracle Designer/2000.** Charakteryzujemy tu możliwości funkcjonalne wersji 2.1 [41,43,44,46] tego oprogramowania. Jest ono bardzo solidne i wszechstronnie sprawdzone w praktyce projektowej. Można je określić jako narzędzie projektowania danowo-procesowego. *Oracle Designer/2000* jest oprogramowaniem CASE o dużej sile ekspresji. Zawiera kilka oracleowych metodologii do całościowego projektowania systemów bazodanowych – z metodologią przepływu pracy (ang. *workflow*) włącznie.

*Oracle Designer/2000* implementuje oracleową metodologię *CASE\*Method* [2]. Do budowy modelu informacyjnego wykorzystuje się diagramy encji systemu (ang. *Entity Relationship Diagrams*). Transpozycja modelu informacyjnego na schemat bazy danych wspomagana jest przez narzędzie przetwarzające diagramy encji (*Server Generator*) w logiczny modela danych. *Designer/2000* pozwala generować schemat bazy danych w postaci skryptów *SQL* osadzających - bez trudności - fizyczną bazę na serwerze bazy danych z systemem zarządzania bazą danych *Oracle Server 8.05*. Działająca instancja bazy wraz z osadzonymi w niej obiektami przechowującymi dane stanowi podstawę informacyjną do funkcjonowania modułów aplikacji wytworzonych za pomocą bildera formatek (*Forms Builder*).



*Oracle Designer/2000* jest narzędziem, które pozwala zaprojektować kompletny schemat bazy tj. stworzyć strukturę tabel - określając nazwy i typy ich pól, określić ich początkowe wartości oraz ich dziedziny. Możliwe jest określenie i utworzenie czterech rodzajów ograniczeń realizujących więzy integralności bazy [2,46].

Oprogramowanie systemu *Oracle Designer/2000* zintegrowano z innymi narzędziami CASE oraz aplikacjami poprzez interfejs *API*.

System projektowy *Oracle Designer/2000* posiada bogate modelowanie danych i bardzo dobre modelowanie procesów z automatyczną synchronizacją diagramów. Stanowi to jego główną zaletę. Do innych zalet należy zaliczyć głęboką integrację oprogramowania *Oracle Designer/2000* z pakietem natywnym *Oracle Developer/2000* w zakresie generowania aplikacji oraz z oprogramowaniem serwerowym *Oracle Server 7* oraz *8* [1].

**Oracle Developer/2000.** *Oracle Developer/2000* stosuje się głównie do wytwarzania części aplikacyjnej projektowanego systemu bazodanowego - scentralizowanego lub w sieciowej architekturze klient-serwer. Część aplikacyjna wykonuje funkcjonalności systemu bazodanowego z bazą danych zarządzaną przez obiektowo-relacyjny system zarządzania bazami danych (skr. ang. *ORDBMS*) *Oracle 8.05* wytworzonymi za pomocą oprogramowania *Oracle Designer/2000*.

**Oracle Forms 6.0.** *Oracle Forms* to narzędzie służące do tworzenia części aplikacyjnej systemu bazodanowego realizującej akcje modyfikujące zawartość bazy danych - wprowadzania, usuwania i aktualizacji danych do bazy. Formularze - inaczej tzw. formatki (ang. *forms*) - tworzone za pomocą *Oracle Forms* realizują operacje manipulowania danymi gromadzonymi w bazie danych *Oracle*. *Oracle Forms* służy również do tworzenia i modyfikowania menu aplikacji i bibliotek programowych.

*Oracle Forms* działa w środowisku graficznym *GUI* systemów operacyjnych, np. *XWindows*, albo *MS Windows9.x* i *Windows NT 4.0/2000*. Składa się z: (a) *Nawigatora Obiektów*, (b) *Edytora Układu* oraz (c) *Arkusza Właściwości Obiektu*.

Tworzenie formularza polega na składaniu go z gotowych „klocków”-komponent. W ten sposób można łatwo i szybko zbudować większość formularzy. Gdy potrzebna jest bezpośrednia ingerencja w zachowanie się formularza, to programuje się wyzwalacze (ang. *triggers*), określających wykonywanie operacji uzależnionych od zachodzenia określonych zdarzeń. Do wielu zdarzeń przypisane są operacje domyślne.

*Nawigator Obiektów* ukazuje wszystkie elementy składowe tworzące dany moduł aplikacji - formularz lub menu. Możemy manipulować tymi elementami używając przycisków narzędziowych lub odpowiednich pozycji menu. Komponenty są zorganizowane i prezentowane w postaci drzewa hierarchii elementów formularza.

Elementami formularzy są:

(1) wyzwalacze w języku *PL/SQL* obsługujące akcje (zdarzenia) i warunki,

- (2) alarmy (ang. *alerts*) dostarczające informacje lub wiadomości wymagające odpowiedzi lub zatwierdzenia przez użytkownika,
- (3) dołączone biblioteki - moduły *Oracle Forms 4.5* zawierające procedury i funkcje wielokrotnego użytku,
- (4) bloki - odzwierciedlają encje (tabele, widoki, migawki) bazy danych oraz obiekty „nie bazowe” do przechowywania różnych informacji (np. kryteriów zapytań, liczników) związanych z więcej niż jednym rekordem bazy,
- (5) powiązania bloków (ang. *master* i *detail*),
- (6) kanwy (ang. *canvas*) - struktury wirtualne obiektów formularzy - wyskakujące i wypełniające, „*button bar*”, *toolbar*, kanwa pusta (pot. „*nullowa*”),
- (7) edytor - okno do przeglądania i modyfikacji danych w postaci opisów i notatek,
- (8) listy *LOV* (skr. ang. *List Of Values*) dla grup rekordów,
- (9) grupy obiektów (obektów w jednym „kontenerze”) - do ponownego użycia w innym formularzu,
- (10) parametrów (z danymi wejściowymi przekazywanych między formularzami),
- (11) jednostki programowe (procedury i funkcje *PL/SQL*), które można wywoływać z wyzwalaczy - do podziału kodu,
- (12) klasy właściwości obiektów,
- (13) grupy rekordów (wirtualne tabele w pamięci) do przekazywania danych pomiędzy modułami aplikacji lub wyświetlania w *LOV*,
- (14) okno - ramka, w którym pojawia się aplikacja (dla systemu *MS Windows* są to okna, w których pojawia się aplikacja oraz okna dialogowe).

Każda kanwa jest przypisana do wybranego okna w formularzu oraz kilka kanw może być przypisanych do jednego okna, formularz może zawierać jedno lub kilka okien.

*Edytor Układu* przedstawia kanwę, na której umieszczone są obiekty.

**Oracle Reports 6.0.** *Oracle Reports* jest narzędziem do projektowania raportów. W czasie uruchomienia raport przesyła do serwera bazy danych *Oracle* zapytanie, następnie przetwarza otrzymane z bazy dane, formatuje je i generuje wyniki. Raporty mogą być przeglądane na ekranie lub drukowane. Raporty mogą być także zapisywane do pamięci dyskowej.

*Oracle Reports 6.0*, jak wszystkie narzędzia *Developer/2000*, pracuje w środowisku graficznym *GUI*, np. *Microsoft Windows* lub *XWindows*. *Oracle Reports Designer* składa się z trzech głównych elementów:

- (1) *Nawigatora Obiektów*,
- (2) *Edytora Modelowania Danych*,
- (3) *Edytora Układu*.

*Nawigator Obiektów* działa na tej samej zasadzie jak *Nawigator Obiektów* w *Oracle Forms 6.0*. *Edytor Modelowania Danych* służy do definiowania zapytania, na podstawie którego są pobierane dane z bazy danych. Wynik



zapytania służy do generowania raportu. Zapytanie może być bardzo skomplikowane. Dlatego buduje się je w graficznie za pomocą *Edytora Modelowania Danych*.

*Edytor Układu* używany jest do wizualnej budowy formy raportu. Używany jest tu specyficzny system ramek. Ramki mają wiele różnych właściwości. Tak jak w *Edytorze Modelu Danych*, *Edytor Układu* obsługuje się za pomocą zestawu przycisków oraz poleceń wywoływanych z menu.

**Oracle Developer Server.** Przypomnijmy, że *Oracle Developer/2000* stosuje się głównie do wytwarzania części aplikacyjnej projektowanego systemu bazodanowego – scentralizowanego lub w sieciowej architekturze klient-serwer.

W celu wytworzenia części aplikacyjnej systemu bazodanowego działającego w architekturze klient-serwer lub architekturze trójwarstwowej - ale w środowisku intranetowym lub internetowym WWW - wykorzystuje się pakiet oprogramowania projektowo-programistycznego *Oracle Developer Server* (skr. *ODS*).

Do tworzenia modułów logiki aplikacji systemu bazodanowego działającej w architekturze klient-serwer wykorzystuje się

- (1) *Forms Developer 6.0* oraz
- (2) *Reaports Builder* z tego pakietu.

Aby aplikacja mogła działać w sieciowym środowisku intranetowym lub w Internecie, jej architektura musi być trójwarstwowa. W celu jej otrzymania należy do budowy „części formatkowej” i „części raportowej” aplikacji zastosować pary narzędzi z *ODS* - odpowiednio: (1a) *Forms Developer 6.0* w połączeniu z (1b) *Forms Server* oraz (2a) *Reaports Builder* w połączeniu z (2b) *Reports Server*. Utworzona aplikacja i baza danych może wtedy działać w środowisku WWW. Aby zbudowany system bazodanowy lub informacyjny mógł działać z przeglądarką sieciową jako klientem, należy zastosować dostępny w *ODS* oracłowy serwer WWW (3) *Oracle WebDB2.21*.

**Sieciowa Architektura Developer Server.** *ODS* to narzędzie programistyczne do dokonywania dystrybucji aplikacji bazodanowych i systemów informacyjnych w intranetach i w środowisku internetowym WWW. Wydajność obliczeniowa aplikacji jest tu zbliżona do osiąganą w architekturze klient-serwer. Pakiet programistyczny *Oracle Developer Server* jest zawiera dobrej jakości narzędzia do tworzenia oraz dystrybucji sieciowych aplikacji bazodanowych i informacyjnych w intranetach i w Internecie architekturze trójwarstwowej.

*ODS* jest przeznaczony do uruchamiania aplikacji bazodanowych w architekturze trójwarstwowej. W architekturze *ODS* można wyróżnić:

- (1) warstwę *Front-End* z klientami na stacjach roboczych,
- (2) warstwę *Middle* z serwerami aplikacji oraz
- (3) warstwę *Back-End* z serwerami baz danych.

Jak każda implementacja architektury trójwarstwowej, *ODS* wnosi znaczne ułatwienia do realizacji całego cyklu życia systemów bazodanowych oraz wiele innych korzyści i zalet. Najważniejsze z nich to:

- (a) zmniejszenie kosztów, czasu i złożoności procesu dystrybucji i



utrzymania rozproszonych sieciowo i terytorialnie systemów bazodanowych, ponieważ wystarczy to robić na serwerach warstwy *Middle*,

(b) łatwość migracji aplikacji bazodanowych z klasycznej architektury klient-serwer - ściślej aplikacje te można sieciowo rozpowszechniać bez zmian plików źródłowych oraz formularzy wykonywalnych ( *\*.fmb* oraz *\*.fmx* ) i nie jest do tego potrzebny żaden język programowania - nawet *Java* lub *JavaScript*,

(c) na stacjach roboczych można stosować „cienkich” klientów (ale trzeba zainstalować przeglądarkę obsługującą język *Java*),

(d) niezależność od platformy - językiem standardowym jest tu *Java*, jednak interfejs użytkownika sieciowych aplikacji bazodanowych jest nieco inny od stosowanego w aplikacjach sieciowych klient-serwer, np. na platformie *Windows* lub w standardzie *Motif*,

(e) zgodność z architekturą przetwarzania sieciowego *NCA* (skr. ang. *Network Computing Architecture*) - aplikacje są uruchamiane jako tzw. kartridże sieciowe,

(f) możliwość dynamicznego uruchamiania raportów w przeglądarce za pomocą specjalizowanego serwera raportów *Multi-tier Server* w powiązaniu z kartridżem *Reports Web Cartridge* z lub *Web CGI*, realizujących przekazywanie żądań do serwerów aplikacji i wyników *HTML* do klientów z przeglądarkami,

(g) wyposażenie w specjalizowane narzędzia: *Reports Builder* do publikacji sieciowych i zawarty w nim kreator publikacji sieciowych *Web Wizard*, narzędzie administrowania raportami w serwerze raportów - *Reports Queue Manager*.

**Budowa architektury ODS.** Sieciowa Architektura *Developer Server* [45] jest zbiorczą nazwą trzech wchodzących w jej skład elementów:

(1) Sieciowa architektura *Forms Builder* (*Forms Builder Web architecture*),

(2) Sieciowa architektura *Reports Builder* (*Reports Web architecture*) oraz

(3) Sieciowa architektura *Graphics Builder* (*Graphics Builder Web architecture*).

Omówimy krótko dwa pierwsze z nich, ponieważ są najczęściej stosowane.

**Klient aplikacji.** Klient aplikacji - ściślej klient formatek (*Forms Client*) to aplet *Javy* pobierany do wykonania z serwera aplikacji (serwera formatek) do przeglądarki sieciowej stacji roboczej. Przeglądarka realizuje funkcje interfejsu graficznego oraz zarządza interakcją między użytkownikiem a serwerem formatek. Klient formatek odbiera polecenia z serwera formatek i tłumaczy je na obiekty interfejsu graficznego. Polecenia z serwera formatek nadchodzą grupowo w pewnych „pakietach”.

Mechanizm wykonujący logikę biznesową formatki (silnik *Forms Runtime Engine*) w sieciowej implementacji intranetowej (architekturze trójwarstwowej) aplikacji, jest uwolniony od wykonywania takich akcji obsługi zdarzeń w interfejsie, które generują nadmierny ruch sieciowy. Również klient zachowuje się dynamicznie, wyświetlając postać interfejsu uzależnioną od stanu przetwarzającej

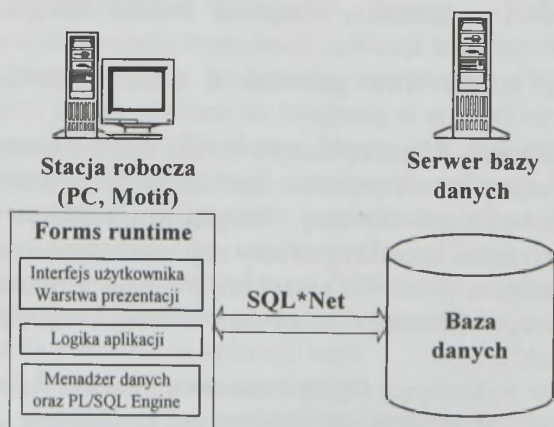
części biznesowej na serwerze aplikacji. Znika też potrzeba dystrybucji apletów *Javy* implementujących aplikację biznesową w rozproszonych i heterogenicznych środowiskach. W ten sposób aplikacja ma cechę ogólności, a klient nie wymaga mocnej stacji roboczej. Tylko w fazie inicjalizacji aplikacji klient pobiera pliki klas generujących używając ich do wygenerowania stanu początkowego aplikacji. Później pobieranie jest dynamiczne.

**Serwer aplikacji.** Serwer aplikacji (*Forms Server*) składa się z dwóch elementów:

- (1) procesu nasłuchującego (*Listener*), który inicjuje sesję serwera aplikacji (*Forms Server Runtime Engine*) i oraz ustala połączenie między tym serwerem a klientem,
- (2) silnika serwerowego (*Server Runtime Engine*). Silnik to zmodyfikowany silnik *Forms Runtime Engine*, w którym funkcje interfejsu użytkownika zostały przekierowane do klienta aplikacji. Zatem obsługuje on zarówno interakcję z użytkownikiem, jak i funkcjonalności biznesowe, ze zdarzeniami, zatwierdzaniem transakcji, zarządzaniem rekordami i całą interakcją z bazą danych włącznie.

**Architektury aplikacji generowanych przez Forms Builder.** Za pomocą narzędzia *Forms Builder* można budować dwa rodzaje implementacji aplikacji - aplikacje w architekturze klient-serwer (klasyczna) oraz tzw. aplikacje sieciowe (aplikacje o architekturze trójwarstwowej).

W pierwszej z nich (Rys. 14.) mechanizm uruchomieniowy wykonywalnych postaci formatek (*Forms Server Runtime Engine*) oraz całość logiki aplikacji (same formatki wykonywalne) są zainstalowane na stacji roboczej. Aplikacja może zawierać fragmenty logiki alokowane po stronie serwera, ale większość zdarzeń jest przetwarzana po stronie stacji roboczej.



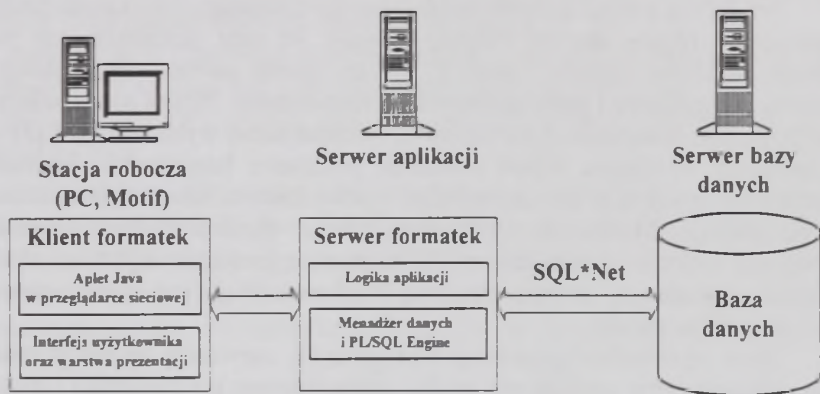
Rys. 14. Oracle Developer: schemat architektury klient-serwer

W aplikacjach wytworzonych w architekturze sieciowej mechanizm uruchomieniowy i logika biznesowa są zaalokowane i funkcjonują na serwerach aplikacji, przetwarzanie zdarzeń odbywa się w bazie danych i na serwerze aplikacji, a tylko przetwarzanie interfejsu użytkownika funkcjonuje na stacji roboczej z klientem. Ilustruje to rysunek 15.

Proces komunikacji serwera z klientem formatek jest inicjowany ustaleniem bezpośredniego połączenia między apletem (*Forms Client*) a serwerem aplikacji (*Forms Server*). Komunikują się one wzajemnie za pomocą żądań i odpowiedzi w postaci skompresowanych komunikatów. Żądania od klienta są zdarzeniami (np. naciśnięcie przycisku lub listy wartości *LOV*). Odpowiedzi z serwera aplikacji są serią aktualizacji na wyświetlanym interfejsie użytkownika.

Aplet (*Forms Client*) komunikuje się z serwerem wówczas, gdy użytkownik przeprowadza:

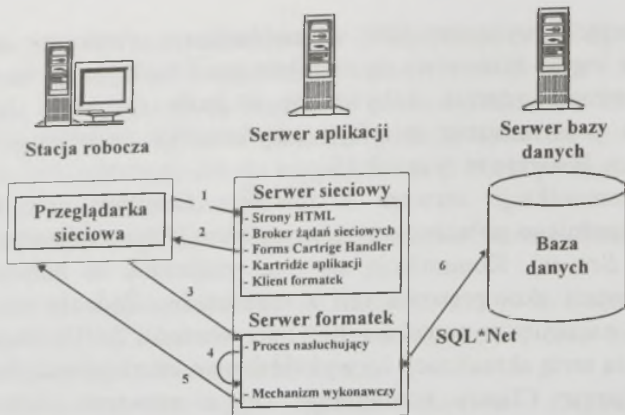
- (1) operacje wysokopoziomowe (np. akceptacja lub anulowanie dialogu),
- (2) operacje walidacji elementów lub powodujące uruchomienie innych wyzwalaczy użytkownika spowodowane np. zaznaczaniem przycisków wyboru i nawigacją pomiędzy polami.



Rys. 15. Oracle Developer: architektura sieciowa

Proces uruchamiania aplikacji sieciowej odbywa się wieloetapowo (Rys. 16.). Najpierw przeglądarka wysyła *URL* (1) odpowiadający stronie - statycznej *HTML* lub o postaci kartridża (ang. *application cartridge*), znajdującej się na serwerze *WWW*.





Rys. 16. Proces uruchamiania aplikacji Forms Builder w sieci

Strona pobierana jest z serwera WWW (2) - razem z klientem aplikacji w postaci apletu. Następnie klient wysyła żądanie do procesu nasłuchującego (*Forms Server Listener*) kontrolującego port komputera (3), z którego aplet został załadowany.

Ten proces inicjuje połączenie lub łączy się z istniejącym procesem mechanizmu wykonawczego (*Forms Server Runtime Engine*) (4) oraz przekazuje mu polecenie otrzymane od klienta (apletu) - może to być np. nazwa modułu zawierającego logikę biznesową, identyfikator i hasło użytkownika, nazwa menu. Proces nasłuchujący nawiązuje bezpośrednie połączenie (typu *socket*) z mechanizmem wykonawczym (5) i wysyła jego parametry do klienta. Klient nawiązuje bezpośrednią komunikację z mechanizmem i zwalnia proces nasłuchujący z pośrednictwa. Klient wyświetla interfejs w oknie przeglądarki. Mechanizm wykonawczy (*Server Runtime Engine*) komunikuje się bezpośrednio z oracłową bazą danych (6) za pomocą protokołu *SQL\*Net*, albo innych protokołów gdy nie jest to baza oracłowa. Ta komunikacja jest zaimplementowana w architekturze klient-serwer.

Dane wymieniane pomiędzy bazą danych, serwerem aplikacji oraz klientem są automatycznie szyfrowane przed i deszyfrowane po transmisji (opcja ta jest domyślnie włączona) za pomocą: (a) protokołu *RSA RC4* z kluczem 40 bitowym dla transmisji między klientem a serwerem, (b) protokołu *SQL\*Net SNS/ANO* dla transmisji między serwerem aplikacji i serwerem bazy danych.

**Sieciowa architektura Reports Builder.** *Reports Builder* (skr. *RB*) wykorzystuje się do publikowania informacji w środowiskach sieciowych - w szczególności w intranetach. Taki sposób publikowania raportów bazodanowych i różnych informacji jest bardzo wygodny i elastyczny. Realizuje się to dwoma sposobami. Pierwszy to generowanie raportów statycznych w postaci plików *HTML* albo *PDF*. Pliki te umieszcza się na serwerze i tworzy do nich linki na stronie internetowej. Metoda ta nadaje się dobrze do publikowania zmieniających się jedynie okresowo, w stałych odstępach czasu, albo przez czas nieokreślony.

Druga metoda publikowania polega na udostępnianiu najświeższych, zmieniających się dynamicznie informacji oraz danych przetworzonych, z aktywnych baz danych. Raporty tworzone są wtedy „on-line” za pomocą przeglądarki sieciowej przez wskazanie adresu *URL*. Po wykonaniu żądanego

raportu następuje jego zwrotne dostarczenie do przeglądarki.

Obie metody raportowania można zaprojektować i zaimplementować za pomocą oprogramowania narzędziowego *Reports Builder*. W pakiecie *RB* służą do tego trzy jego komponenty:

- (1) kreator publikacji sieciowych *Web Wizard*,
- (2) edytor *Report Editor* oraz
- (3) serwer *Reports Multi-Tier Server*, który pozwala raportom nadawać podstawowe właściwości i generować je w postaci plików *HTML* lub *PDF*.  
*RB* służy też do wykonywania raportowania. Do okresowego odświeżania publikowanego wyniku można wykorzystać czwarty komponent *RB* -
- (4) *Report Server* do planowania wykonywania żądanych raportów.

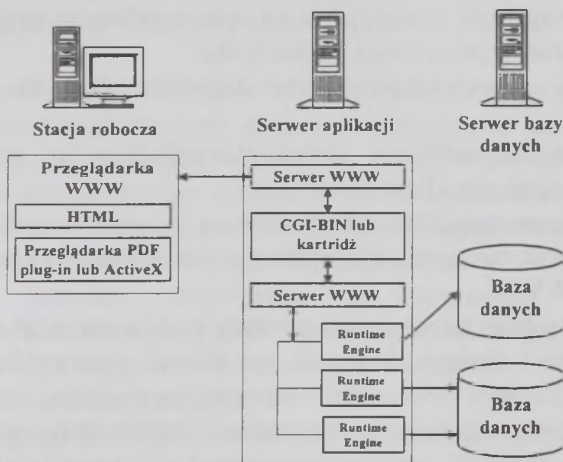
Natomiast do dynamicznego tworzenia raportów sieciowych należy wykorzystać *Reports Server* w powiązaniu z (4a) *Reports Web Cartridge* lub (4b) *Reports Web CGI*, które to po uaktywnieniu odpowiedniego linka przetwarzają adres *URL* i wysyłają żądanie wykonania zadania do serwera *Reports Server*. Zadanie jest następnie przekazywane do wykonania do wolnego mechanizmu wykonawczego (silnika *Runtime Engine*) a wyniki są zwracane przez serwer sieciowy do przeglądarki.

**Report Server (RS).** *Report Server* umożliwia uruchamianie raportów okresowych lub dynamicznych na zdalnym serwerze.

*Reports Web Cartridge* wysyła żądania z serwera *WWW* do serwera *RS*, lecz serwerem *WWW* musi być natywny *Oracle Application Server*.

*Reports Web CGI* wysyła żądania z serwera *WWW* do serwera *RS* zgodnie ze standardem *CGI*.

Dokładniejsze omówienie funkcjonowania i mechanizmów realizacyjnych zastosowanych w serwerze *RS*, kartridżu *Reports Web Cartridge* oraz interfejsie *Reports Web CGI* wykracza poza ramy tego artykułu. Można je znaleźć np. w [45]. Na rysunku 17 naszkicowano schemat sieciowej architektury modułu *Reports Builder* z pakietu oprogramowania projektowo-implementationalnego *Oracle Developer Server*.



Rys. 17. Oracle Developer Server: sieciowa architektura Report Builder

Oprogramowanie *Sybase PowerDesigner* firmy Powersoft to jest jednym z kilku wiodących narzędzi i środowisk CASE nowej generacji do modelowania i projektowania sieciowych systemów bazodanowych i aplikacji wielowarstwowych. Za jego pomocą można projektować i wytwarzać aplikacje architektury dwu oraz trzy warstwowej. Posiada on dużą siłę ekspresji projektowej i duże możliwości generowania prototypów klasycznych, sieciowych i internetowych systemów bazodanowych.

Możliwości narzędzia *Sybase PowerDesigner* ze składnikami *ProcessAnalyst*, *DataArchitect*, *AppModeler* i *MetaWorks* pozwalają stwierdzić, że *PowerDesigner* jest bardzo wydajnym i przyjaznym dla projektanta narzędziem, przy pomocy którego w krótkim czasie można wytworzyć w pełni funkcjonalną aplikację biznesową. Mimo, że narzędzie to wydaje się dość złożone, to po zapoznaniu się z nim staje się ono proste w użyciu. Dużą rolę odgrywa tutaj podobieństwo i konsekwencja z jaką zaprojektowano wygląd wszystkich jego składników. Dzięki temu łatwo poznaje się kolejne narzędzia pakietu *PowerDesigner*. Niewielkie wymagania sprzętowe sprawiają, że opisane w tej pracy narzędzie może z powodzeniem zastąpić już nieco przestarzałe systemy takie jak *EasyCASE*.

Z drugiej zaś strony *PowerDesigner* jest jakościowo zbliżony do bardzo złożonych systemów, na przykład do narzędzia projektowych firmy Oracle. Pakiet *Oracle Designer 6i* z repozytorium opartym na bazie danych *Oracle 8i* to jest bardziej rozbudowane jeżeli chodzi o pracę grupową. Jest to jednak okupione dużo większymi wymaganiami sprzętowymi. W ekspresji projektowej narzędzia te nie różnią się istotnie.

**Sybase PowerDesigner.** Oprogramowanie *Sybase PowerDesigner 6.0* [28,30] zajmuje znaczące miejsce wśród narzędzi CASE wśród których z jednej strony znajdują się narzędzia nieskomplikowane - jak *EasyCASE* firmy Evergreen, a z drugiej narzędzia bardzo złożone - jak *Oracle Designer* firmy Oracle.

Dla ustalenia uwagi opis nakierujemy na opis możliwości projektowania systemu oraz automatycznego generowania aplikacji dla:

- (1) systemu DBMS *Sybase Adaptive Server Anywhere 6.0.1* do osadzenia części danowej,
- (2) środowiska implementacyjnego *Sybase PowerBuilder 7.0* wytwarzania dwuwarstwowej aplikacji klient-serwer,
- (3) środowiska implementacyjnego i Web serwera *Sybase PowerDynamo 3.0*, dającego możliwość tworzenia trzywarstwowych aplikacji klient-serwer z dostępem przez WWW.

*Sybase PowerDesigner* stanowi zintegrowany zestaw narzędzi do analizy, projektowania, tworzenia i obsługi złożonych baz danych oraz aplikacji klient-serwer i internetowych. Zawiera mechanizmy wspomagające analizę i specyfikację oraz narzędzia do modelowania danych i procesów, umożliwiające generowanie bazy danych i obiektów aplikacji. Jego modułowa struktura umożliwia elastyczny



wybór rozwiązań w zależności od rozmiaru projektów i wymagań użytkowników. Wspiera on pracę grupową i zawiera intuicyjny interfejs użytkownika. Narzędzie to posiada budowę modułową. Możemy w nim wyróżnić następujące składniki:

- (A) *PowerDesigner DataArchitect*, umożliwiający modelowanie danych, generowanie, rewersowanie i dokumentowanie dla szerokiej gamy systemów baz danych,
- (B) *PowerDesigner ProcessAnalyst*, służący do analizy i modelowania procesów biznesowych,
- (C) *PowerDesigner AppModeler*, pozwalający automatycznie generować aplikacje klient-serwer,
- (D) *PowerDesigner MetaWorks*, wspomagający pracę grupową i zapewniający efektywne zarządzanie obiektami projektu przechowywanymi w centralnym słowniku.

*Sybase Adaptive Server Anywhere* to relacyjny, wielodostępny serwer bazodanowy o niewysokich wymaganiach sprzętowych. Jest on przedstawicielem rodziny korporacyjnych baz danych *Sybase Adaptive Server*, zoptymalizowanym pod kątem efektywności pracy użytkownika. Charakteryzuje go łatwość konfiguracji i zarządzania przy jednoczesnym zapewnieniu wysokiej wydajności i bezpieczeństwa danych. *Adaptive Server Anywhere* jest bazą łatwą do uruchomienia. Instalacja przebiega prawie automatycznie, podobnie przebiega konfiguracja. Graficzne narzędzie administracyjne, *Sybase Central*, umożliwia zastosowanie techniki "przenieś i upuść" łatwe zarządzania replikacją, tworzeniem tabel, indeksów i procedur wbudowanych.

*Sybase Adaptive Server Anywhere* dostępny jest dla wielu platform systemowych m. in. dla Windows 95/98, Windows NT/2000, Sun Solaris, HP-UX, IBM AIX, Novell Netware, OS/2, a także od niedawna dla systemu Linux. Jego architektura pozwala projektować aplikacje tak, aby bez modyfikacji mogły pracować zarówno na pojedynczym stanowisku, jak i w środowisku wielodostępnym na różnych platformach, z użyciem różnych protokołów sieciowych.

*Adaptive Server Anywhere* ułatwia tworzenie zaawansowanych aplikacji obsługując multimedialne typy danych pozwalające na przechowywanie w bazie danych obiektów o dużych rozmiarach (BLOB) jak grafika, filmy, dźwięki. Ponadto umożliwia obsługę alfabetów narodowych, łatwe wczytywanie danych zapisanych w innych formatach. Oferuje łatwy i szybki dostęp do danych zarówno przez interfejs ODBC jak i z aplikacji napisanych w C i C++. *Server Anywhere* umożliwia również współpracę z Internetem zapewnioną przez *Web server PowerDynamo*. Dzięki temu programiści mogą tworzyć strony WWW zawierające informacje z bazy danych w standardzie HTML z wbudowanymi instrukcjami oraz kodem w SQL.

*PowerSoft PowerBuilder* to środowisko RAD do budowy aplikacji klient-serwer, wielowarstwowych, rozproszonych i internetowych. Usprawnia i rozszerza możliwości tworzenia, zarządzania i uruchamiania biznesowych aplikacji dla Internetu i intranetów oraz umożliwia tworzenia standardowych komponentów

typu CORBA, COM/DCOM, JavaBean. Posiada dedykowane sterowniki do baz danych i jest dostarczany wraz z narzędziem *PowerDesigner AppModeler for PowerBuilder*.

*PowerBuilder 7.0* jest efektywnym narzędziem zapewniającym dużą wydajność pracy w środowisku 4GL RAD podczas tworzenia transakcyjnych, internetowych aplikacji dla dużych organizacji. Działa on na platformach Windows 95/98, Windows NT/2000 (Intel i Alpha), Macintosh, Sun Solaris (Motif), Win OS/2, HP/UX i IBM AIX. Używany język programowania czwartej generacji PowerScript jest prosty w użyciu i umożliwia dostęp do wielu wbudowanych funkcji. Można w nim pisać własne funkcje lub korzystać z istniejących napisanych w C++ lub innym języku, udostępnianych w plikach DLL. *PowerBuilder* posiada standardowe cechy narzędzia obiektowego: wielopoziomowe dziedziczenie, hermetyzacja danych i funkcji oraz polimorfizm. Jego kreatory automatycznie tworzą funkcjonalny szkielet komponentu, projekt aplikacji oraz dostosowywalną listę czynności "do zrobienia". Dołączone zaś biblioteki klas, *PowerBuilder Foundation Class*, oraz przykładowe aplikacje pomagają w szybkim prototypowaniu i konstrukcji oprogramowania.

*PowerBuilder* zapewnia wydajną współpracę z bazami danych. Zawiera on dedykowane sterowniki pozwalające na bezpośredni dostęp do najbardziej rozpowszechnionych systemów relacyjnych baz danych: *DB2*, *DB2/600*, *Informix*, *Microsoft SQL Server*, *Oracle*, *Sybase Adaptive Server Enterprise*, *Sybase Adaptive Server Anywhere*. Współpraca z pozostałymi systemami bazodanowymi możliwa jest za pomocą unowocześnionego mechanizmu ODBC.

*PowerDynamo* to samodzielny internetowy serwer aplikacji WWW, udostępniający "cienkie", dynamiczne aplikacje oparte o bazę danych w sieciach intranetowych i w Internecie. Serwer *PowerDynamo* dostępny jest na platformy Windows 95/98, NT/2000 oraz Sun Solaris. Uzupełnia on możliwości klasycznych serwerów Web przez dynamiczne wykonywanie szablonów i wysyłanie do klienta odpowiedzi w HTML i JavaScript. Są to „cienkie” aplikacje. Wykorzystanie szablonów zapewnia, że do klienta dociera tylko kod HTML dostępny za pomocą przeglądarki bez konieczności rezygnowania z implementacji logiki biznesowej, która jest tutaj zrealizowana za pomocą wbudowanych instrukcji SQL. Szablony HTML są przechowywane w bazie danych, co zapewnia ich łatwą replikację. Rozszerzenia języka, takie jak Java, ActiveX, C/C++ oraz możliwość wywoływania skompilowanych fragmentów kodu, zwiększają możliwości HTML. Dodatkowo, architektura *PowerDynamo* obsługuje CGI, NSAPI oraz ISAPI oferując pełną komunikację z popularnymi serwerami WWW. Dla komunikacji z bazami danych *PowerDynamo* obsługuje ODBC i Sybase CT-Lib.

*PowerDynamo* zawiera wiele udogodnień ułatwiających i przyspieszających tworzenie aplikacji WWW, m.in. przykładową, funkcjonalną aplikację. Kreatory wspierające tworzenie szablonów redukują czas i wysiłek konieczny do stworzenia i uruchomienia aplikacji. *PowerDynamo* zawiera graficzny składacz (ang. composer) zapytań SQL i kolorowy edytor, a dla bardziej zaawansowanych programistów, oferuje współpracę z wieloma innymi



środowiskami projektowymi. Pliki oraz istniejące strony WWW stworzone za pomocą innych narzędzi, łatwo mogą zostać zaimportowane, zachowane w bazie danych i zarządzane przez *PowerDynamo*. Do administracji służy graficzne narzędzie *Sybase Central*, wykorzystywane w wielu produktach Sybase m. in. w *Adaptive Server Anywhere*.

*PowerDynamo* oparte jest na języku DynaScript, zgodnym ze standardem przemysłowym PowerScript, co umożliwia wykorzystanie dotychczasowych doświadczeń programistów *PowerBuildera* oraz istniejących zasobów przy tworzeniu funkcjonalnych rozszerzeń do HTML. Skrypty DynaScript zawierają zdefiniowane obiekty dla powszechnie używanych funkcji, takich jak zarządzanie sesjami czy manipulacja rezultatami zapytań SQL.

**Składnik DataArchitect** jest narzędziem do projektowania baz danych, które daje użytkownikowi możliwość projektowania dwupoziomowego, tzn. tworzenia modelu konceptualnego i fizycznego. Używając narzędzia *DataArchitect* mamy dostęp do następujących funkcji:

- (a) modelowanie systemów informacyjnych przy pomocy diagramu związków encji, nazywanego modelem konceptualnym (ang. Conceptual Data Model - skr. ang. CDM),
- (b) generowanie odpowiadającego mu modelu fizycznego (ang. Phisical Data Model - skr. ang. PDM) dla konkretnego systemu DBMS biorąc pod uwagę jego specyfikę,
- (c) dostosowanie modelu PDM do wymagań fizycznych i wydajnościowych,
- (d) generowanie skryptu generującego docelową bazę danych,
- (e) generowanie wyzwalaczy integralności referencyjnej, jeżeli są one obsługiwane przez docelową bazę danych,
- (f) dostosowywanie i drukowanie raportów,
- (g) rewersowanie istniejących baz danych i aplikacji,
- (h) definiowanie rozszerzonych atrybutów dla obiektów PDM.

*Model konceptualny CDM* reprezentuje całkowitą strukturę systemu informacyjnego. Reprezentuje całość struktury logicznej bazy danych. Opisuje on koncepcyjne związki pomiędzy różnymi typami informacji, a nie ich fizyczną strukturę. CDM jest niezależny od poszczególnych systemów DBMS.

*Model fizyczny PDM* przystosowuje tworzony projekt do konkretnego DBMS i jest kolejnym krokiem do uzyskania fizycznej implementacji. Procedura generowania modelu PDM umożliwi przeniesienie całości struktury projektu CDM do modelu fizycznego.

Oba modele posiadają reprezentacje graficzne. Proces projektowania struktury danowej bazy za pomocą środowiska *PowerDesigner* rozpoczyna się od poziomu konceptualnego. Obejmuje interakcje pomiędzy obiektami takimi jak dziedziny, atrybuty, encje i relacje, oraz dziedziczenie - szczególną relację, która definiuje encje jako przypadek bardziej ogólny. Model konceptualny stanowi dane wejściowe do wygenerowania modelu PDM specyfikującego fizyczną implementację bazy danych oraz weryfikuje i waliduje on projekt danowy.

**Generowanie modelu fizycznego.** Model PDM generuje się dla



poszczególnych systemów zarządzania bazami danych DBMS. Przed wygenerowaniem modelu PDM należy wybrać jeden z oferowanych docelowych systemów DBMS dla docelowej bazy danych. *DataArchitect* transluje typy danych wyspecyfikowane w modelu CDM do fizycznych typów danych dostępnych w docelowej bazie danych. Odpowiedniość pomiędzy typami konceptualnymi i fizycznymi definiuje się w specjalnym pliku (DEF). Istnieje plik dla każdego typu w docelowej bazie danych.

Generując model PDM, *DataArchitect* transluje obiekty konceptualne w obiekty fizyczne.

**Model fizyczny.** Model PDM reprezentuje strukturę fizyczną danych, która zostanie zaimplementowana w systemie DBMS. Bierze się tu pod uwagę cechy i fizyczne ograniczenia poszczególnych systemów DBMS.

Za pomocą PDM można optymalizować charakterystyki bazy danych środkami dodefiniowywania tabel, kolumn, indeksów, integralności referencyjnej, perspektyw, wyzwalaczy i procedur pamiętanych. Model PDM specyfikuje fizyczną implementację bazy danych.

Model PDM spełnia następujące role:

- (a) reprezentuje fizyczną organizację danych w formie graficznej,
- (b) generuje skrypty tworzenia bazy danych i jej modyfikacji,
- (c) definiuje wyzwalacze i więzy integralności referencyjnej,
- (d) generuje atrybuty rozszerzone,
- (e) rewersuje istniejące bazy danych.

Model PDM reprezentuje interakcję obiektów takich typów jak tabele, kolumny, klucze, indeksy, referencje perspektywy.

Dla Modelu fizycznego można utworzyć jego użytkowników. Użytkownik modelu PDM to nazwa identyfikująca osobę lub grupę będącą właścicielem obiektów z modelu - tabel lub perspektyw.

**Procedury i wyzwalacze.** Wyzwalacze więzów integralności mogą być w modelu PDM generowane automatycznie w zależności od typu docelowej bazy danych. Użytkownik ma możliwość zdefiniowania dodatkowych wyzwalaczy.

*PowerDesigner* generuje wyzwalacze na podstawie wzorców szablonów. Każdy szablon wyzwalacza pobudza trzy rodzaje więzów integralności referencyjnej: wstawiania, modyfikacji oraz usuwania. Każdy docelowy DBMS posiada jeden wzorzec wyzwalacza każdego typu.

**Tworzenie bazy danych.** Bazę danych można wygenerować bezpośrednio z modelu PDM, albo można wygenerować tylko skrypt generujący bazę - do uruchomienia w docelowym środowisku DBMS. Skrypty generuje się dla poszczególnych docelowych baz danych.

Dostępne parametry do generacji zależą od wyboru docelowej bazy danych. Domyślną docelową bazą danych jest baza wybrana przez projektanta przy otwarciu modelu PDM, ale można to zmienić po wygenerowaniu skryptu. Połączenie z bazą danych zapewnione jest poprzez interfejs ODBC, który umożliwia dostęp do danych bazy za pomocą języka SQL stanowiącego standard dostępu dodanych.

Za pomocą narzędzia *PowerDesigner* można konfigurować przestrzenie tabel i pamięci. Przestrzeń tabel (ang. *tablespace*) i pamięć (ang. *storage*) wskazują na fizyczną lokację obiektów bazy danych (tabel i indeksów). Definicje przestrzeni i pamięci są specyficzne dla systemu DBMS. Dla różnych systemów DBMS stosuje różne określenia przestrzeni tabel i pamięci.

W różnych systemach bazodanowych stosuje się różne komendy do tworzenia przestrzeni tabel i pamięci. Bazę można utworzyć dla następujących systemów DBMS: *Allbase*, *DB2*, *Oracle 5*, *Oracle 6* lub *7*, *RDB 4* lub *5* lub *6*, *SQLBase*, *Sybase System 10* lub *System 11*, *SQL Server*, *SQL Anywhere Watcom 3* lub *4*.

**Generowanie danych testowych.** Dane testowe są próbką danych, którą definiujemy dla jednej więcej tabel w modelu PDM. Danych testowych można używać do weryfikowania wydajności bazy gdy zawiera ona dużo danych lub gdy jest ona udostępniana wielu różnym aplikacjom lub użytkownikom. *PowerDesigner* automatycznie generuje wiersze danych testowych i umieszcza je w tabelach.

**Inżynieria odwrotna.** *PowerDesigner* daje możliwości wygenerowania modelu PDM, albo poszczególnych obiektów z istniejącej bazy danych. Proces ten nazywamy inżynierią odwrotną (ang. *reverse engineering*).

Rewersowaniu mogą podlegać obiekty typów posiadających właścicieli, tzn. tablice, wyzwalacze, perspektywy, synonimy, tabele systemowe, oraz procedury.

Rewersowaniu podlegają również obiekty bazy, które nie posiadają właścicieli. Są to: typy abstrakcyjne, przestrzenie tabel, przestrzenie przechowywania, typy zdefiniowane przez użytkownika, oraz użytkownicy.

**Składnik ProcessAnalyst** jest narzędziem pakietu *PowerDesigner*, które umożliwia reprezentowanie procesów w systemie informacyjnym. Produktem pracy narzędzia *ProcessAnalyst* jest PAM – *ProcessAnalyst Model*. PAM oferuje funkcjonalną analizę systemu informacyjnego. Służy on do analizy procesów składających się na opisywany system. PAM jest uzupełnieniem modelu konceptualnego CDM i ukazuje dynamikę istniejącą pomiędzy elementami systemu. CDM natomiast opisuje statyczne fakty będące sercem tego systemu.

*ProcessAnalyst* służy do modelowania przepływów danych pomiędzy poszczególnymi elementami systemu. Spełnia on następujące role: służy analizie procesów w systemie, reprezentuje te procesy w formie modelu oraz tworzy hierarchiczną strukturę reprezentującą te procesy. Do realizacji tych celów wykorzystuje on cztery metodologie projektowe:

- (1) model funkcjonalny OMT (Object Modeling Technique),
- (2) Yourdon/DeMarco (diagramy przepływu danych),
- (3) Gene & Sarson (diagramy przepływu danych),
- (4) SSADM (Structured System Analysis and Design Methodology) (diagramy przepływu danych).

Model funkcjonalny OMT jest najnowszą i najbardziej ekspresywną wersją diagramów przepływu danych i dlatego przyjęto go w *ProcessAnalyst* jako metodologię domyślną.



Pozostałe zaimplementowane metodologie tj. Yourdon/DeMarco, Gane & Sarson i SSADM bazują na metodykach analizy strukturalnej. Stosuje się w nich diagramy przepływu danych (DFD).

**Definiowanie reguł biznesowych i danych.** Reguła biznesowa odwzorowuje semantykę modelowanego procesu. W *ProcessAnalyst* wyróżniamy cztery rodzaje reguł: definicja, fakt, formuła, wartościowanie.

Narzędzie *PowerDesigner ProcessAnalyst* umożliwia definiowanie danych na poziomie dziedzin i atrybutów. Dziedziny ułatwiają identyfikację typów informacji w projekcie. Stosując dziedziny atrybutów można łatwiej standaryzować charakterystykę danych.

W PAM, można skojarzyć z dziedziną następujące informacje: typ danych, długość, precyzję, oraz reguły biznesowe.

**Definiowanie procesów.** W modelu PAM występuje jeden wyróżniony model zwany modelem kontekstowym. Model kontekstowy to model PAM który posiada główny proces – rodzic poziomu 0. Każdy proces można zdekomponować, czyli podzielić na procesy niższego poziomu.

**Obiekty globalne i lokalne.** W modelu PAM wyróżniamy dwa typy obiektów: obiekty globalne i obiekty lokalne. Obiekty globalne to: składnice danych i terminatory (ang. external entity), a obiekty lokalne to: procesy, przepływy danych, przepływy przenoszone (ang. migrated flow), rozgałęźniki/koncentratory (ang. split/merge).

**Macierz CRUD.** Macierz CRUD (skr. ang. Create, Read, Update, Delete) jest tablicą, która pokazuje powiązania pomiędzy procesami i składnicami danych, albo pomiędzy procesami i atrybutami lub jednostkami danych. Gdy powiązanie istnieje, to pokazuje ono czy proces wykonuje operację tworzenia, czytania, aktualizacji lub usuwania na składnicy danych lub atrybutów albo jednostek danych.

W *ProcessAnalyst* istnieją dwa typy macierzy CRUD:

- (a) macierz CRUD proces/składnica danych, w której powiązania istnieją pomiędzy procesami a składnicami danych; macierz pokazuje operacje wykonywane przez proces na składnicy danych,
- (b) macierz CRUD proces/atribut (jednostka danych) pokazująca operacje wykonywane przez proces na atrybucie lub jednostce danych.

Macierz CRUD używana jest do przeglądania powiązań między procesami i używanymi przez nie danymi i wskazywania, do których danych z jakich składnic poszczególne procesy nie mają dostępu, oraz jakiego typu akcje są wykonywane na poszczególnych danych. Umożliwia to wykonywanie ekspresywnej weryfikacji modelu.

**Importowanie definicji danych z *DataArchitect*.** Narzędzie *ProcessAnalyst* daje możliwości importowania danych z modelu koncepcyjnego CDM. Ma to szczególne znaczenie dla utrzymania zgodności całego projektu. Importowaniu podlegają następujące elementy CDM: atrybuty, dziedziny, reguły biznesowe oraz encje, które w modelu PAM reprezentowane będą jako składnice danych. Ważną zaletą jest możliwość wyboru danych do importu. Funkcjonalność



ta przydaje się podczas aktualizowania modelu PAM, gdy przykładowo chcemy uzyskać zgodność jedynie na poziomie atrybutów.

**Składnik AppModeler.** W skład pakietu *PowerDesigner* wchodzi generator aplikacji AppModeler do generowania aplikacji dla następujących środowisk programistycznych: *PowerBuilder*, *Visual Basic*, *Power++*, *Delphi*, *Web* z zastosowaniem serwera WWW *PowerDynamo*, *Web* generującego strony *ASP (Active Server Pages)*.

*AppModeler* jest narzędziem do projektowania aplikacji klient-serwer. Zapewnia ono użytkownikowi wszystkie korzyści płynące z fizycznego projektowania danych i implementacji interfejsu klient-serwer.

Za jego pomocą można:

- (1) budować model fizyczny PDM dla docelowego systemu zarządzania bazą danych z uwzględnieniem specyfiki wybranego docelowego systemu zarządzania bazami,
- (2) przystosowywać model PDM do wymagań fizycznych i wydajnościowych,
- (3) generować skrypt kreujący bazę danych dla docelowego systemu DBMS,
- (4) generować wyzwalacze integralności referencyjnej jeśli mają one zastosowanie w docelowej bazie danych,
- (5) przystosowywać i drukować raporty o modelu,
- (6) rewersować istniejące bazy danych i aplikacje,
- (7) definiować atrybuty rozszerzone do generowania aplikacji 4GL.

Cechą narzędzia *AppModeler* jest to, że część funkcjonalności pokrywa się z narzędziem *DataArchitect*. Mowa tutaj o budowie modelu fizycznego i generowaniu schematu bazy danych. W zamierzeniu jednak *AppModeler* jest generatorem aplikacji klienckich.

Wszystkie generatory *AppModeler* posiadają podobny interfejs użytkownika. Zasady funkcjonowania generatorów są również podobne. Istnieją jednak różnice wynikające ze zróżnicowania środowisk docelowych. Omówimy krótko dwa generatory: *AppModeler for PowerBuilder* i *AppModeler for the Web*, tworzące aplikacje w zupełnie różnych środowiskach i architekturach.

### ***AppModeler for PowerBuilder***

**Wiadomości podstawowe.** Generator obiektów *PowerBuildera* (w skrócie *PBGen*) umożliwia generowanie aplikacji na podstawie obiektów i atrybutów zdefiniowanych w modelu PDM. *PBGen* wykorzystuje dane z modelu PDM do wytworzenia instancji z szablonów w celu wygenerowania aplikacji, okien, obiektów użytkownika i menu.

*AppModeler* transportuje aplikacyjne szablony generujące dla interfejsu aplikacji MDI (ang. Multiple Document Interface). *AppModeler* nie posiada szablonów dla aplikacji SDI (ang. Single Document Interface). W aplikacji MDI mamy dwa typy okien: okna ramek lub aplikacji, oraz okna formatek. *PBGen* tworzy instancje ramek okien z informacjami z modelu. Dotyczy to tabel, widoków i referencji. Podczas projektowania aplikacji za pomocą generatora aplikacji

*PowerBuilder* musimy specyfikować nazwy takich obiektów jak okna i menu.

Projekt w *PBGen* wykorzystuje informacje z bazy danych o aplikacji.

*AppModeler* stosuje autentykację sterowników ODBC używanych do połączenia z generowaną bazą danych albo do celów rewersowania. Można jednak zastosować sterowniki bez takich restrykcji, np. do przyłączenia generowanych aplikacji oraz aplikacyjnych baz danych.

**Budowa aplikacji.** Przed przystąpieniem do generowania aplikacji za pomocą *PBGen* należy wykonać następujące czynności: zdefiniować model atrybutów zewnętrznych, określić biblioteki *PowerBuilder*, określić szablony aplikacji, wskazać lokalizację i profil bazy danych.

Następnie można przystąpić do definiowania atrybutów rozszerzonych. Atrybuty rozszerzone modelu wiążą model PDM z cechami aplikacji i menu.

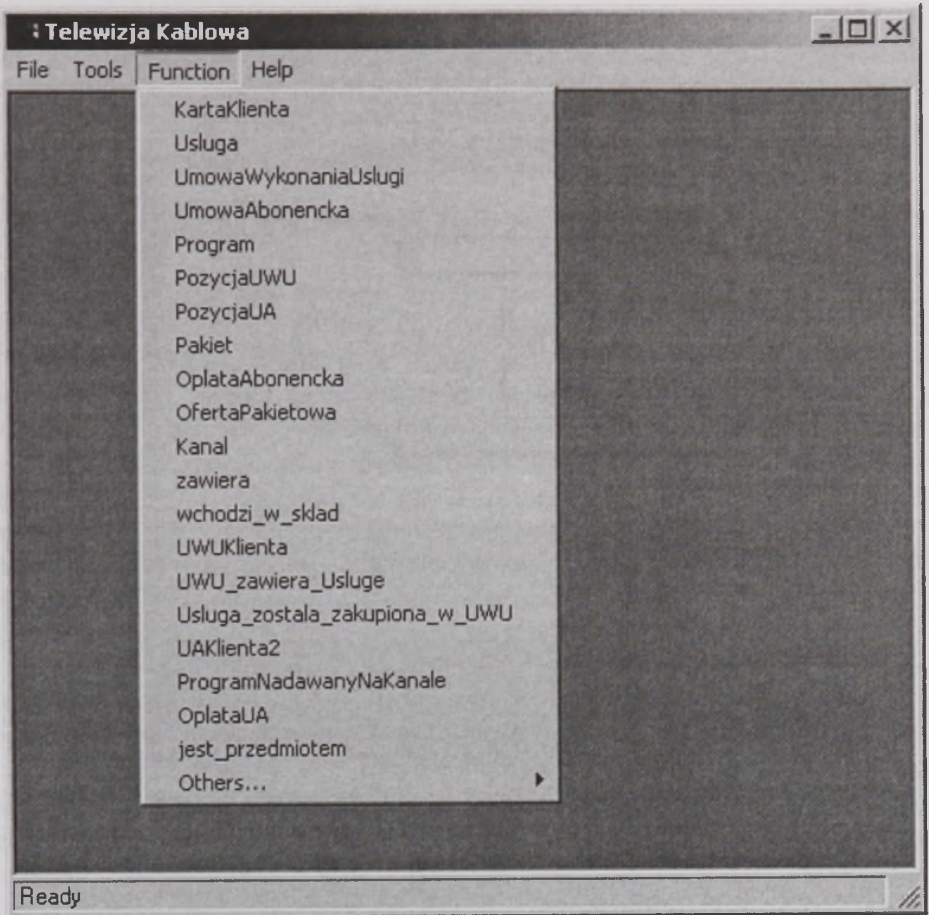
Na poziomie aplikacji można wybrać szablony aplikacji, okna aplikacji, menu oraz arkuszy okien.

**Dostrajanie aplikacji przed oraz po generowaniu.** Przed wygenerowaniem lub ponownym wygenerowaniem aplikacji można ją dostroić i przetestować z widoku drzewa okna „Application Generation” generowania aplikacji.

Można tutaj modyfikować właściwości takich elementów jak menu, okna, okna danych. Duży model PDM może zawierać setki jednostek zawartych w drzewie okna generowania aplikacji Application Generation. Dużą ilość informacji pomagają uchwycić tzw. filtry.

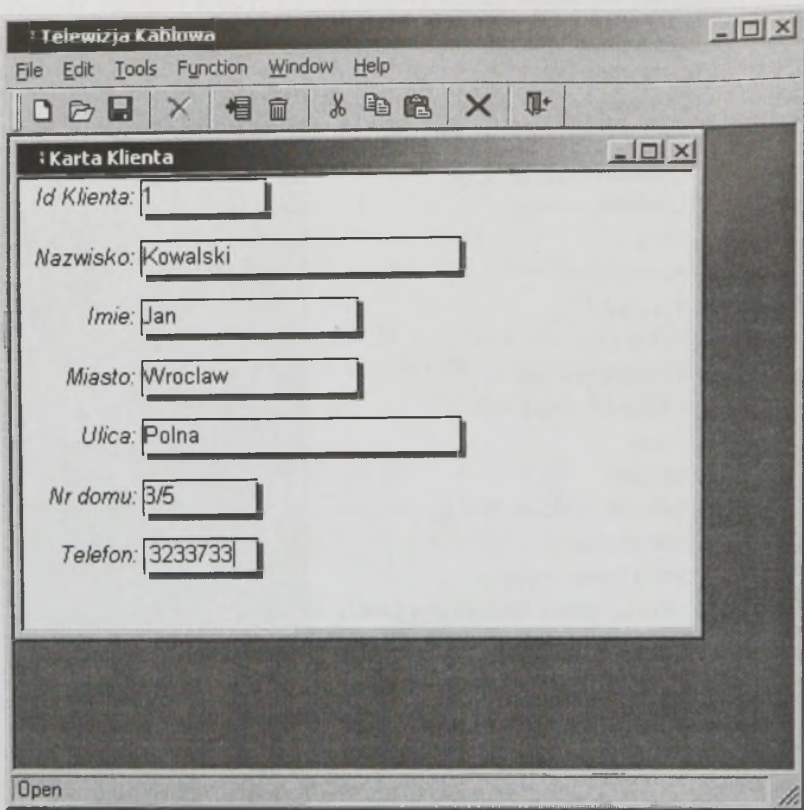
Z okna Application Generation generowania aplikacji można również otworzyć tzw. painter aplikacji. Za pomocą tego narzędzia możemy testować ostatnio wygenerowaną wersję aplikacji.

Na kolejnych dwóch rysunkach (Rys. 18 i Rys. 19) przedstawiono okno główne z rozwiniętym menu oraz okno danych „Karta klienta” wygenerowanej przez *AppModeler for PowerBuilder* przykładowej aplikacji bazodanowej.



Rys.18. Okno główne z rozwiniętym menu aplikacji





Rys.19. Otworzone okno danych „Karta Klienta”

**Wykorzystywanie szablonów *PBGen*.** *PBGen* zawiera dwie kompletne i niezależne biblioteki szablonów. Biblioteki szablonów *PBGen* zawierają osobne szablony dla okien z ramkami i osobne dla arkuszy okien menu.

**Tworzenie szablonów *PBGen*.** *PBGen* umożliwia generowanie obiektów *PowerBuilder'a* na podstawie szablonów własnych. Po zmianie nazwy szablonu i przestrzegając tej konwencji, można uzyskać do nich dostęp z okna dialogowego *PBGen* i generować aplikacje na podstawie tych szablonów. Stosując *PBGen* można przypisać wartości tym zmiennym dla aplikacji albo dla okna.

Jeżeli tworzymy własne szablony z pracujących aplikacji i nadajemy im nazwy stosując konwencje opisane wyżej, możliwe że zajdzie konieczność poprawienia błędów generacji. Zwykle łatwiej jest zdebugować wygenerowaną aplikację, zastosować zmiany do obiektów szablonów i ponownie wygenerować aplikację.

**Używanie repozytorium *PowerBuilder'a*.** Atrybuty katalogu są atrybutami rozszerzonymi z systemu tabel repozytorium *PowerBuilder'a*.

Repozytorium zawiera informacje o sposobie wyświetlania danych z bazy danych na ekranie i w raportach. Jeśli repozytorium *PowerBuilder'a* definiuje

wartości dla atrybutów katalogu, to można wyekstrahować je z repozytorium i użyć ich w bieżącym modelu PDM. Jest to typ rewersowy (inżynierii odwrotnej). Po zdefiniowaniu wartości dla atrybutów katalogowych można dołączyć je do domen, tabel i kolumn. Można również wygenerować je w repozytorium *PowerBuilder'a*.

### ***AppModeler for the Web. Web generator for PowerDynamo***

**Wiadomości podstawowe.** *AppModeler for the Web* (w skrócie *WebGen*) jest generatorem aplikacji WWW dla sieci Web. *WebGen* wykorzystuje dane z PDM do stworzenia obiektów z dostarczonych szablonów i wygenerowania aplikacji WWW. Projekt witryny internetowej generowany jest bezpośrednio do wskazanego folderu WWW lub bazy danych skonfigurowanej dla *PowerDynamo*.

Projekt witryny, który rozwija się z *WebGen* używa informacji z bazy danych o aplikacji. Analogicznie jak w przypadku *PBGen*, aby projekt mógł poprawnie działać, należy się najpierw upewnić czy baza danych istnieje i czy zawiera pola i typy danych zdefiniowane w modelu PDM. *WebGen* tak jak wszystkie inne generatory *AppModeler* komunikuje się z bazą danych poprzez sterowniki ODBC.

**Tworzenie projektu witryny.** *WebGen* generuje projekt strony głównej służący do zarządzania aplikacją sieci Web. Strona główna zbudowana jest na ramkach i zawiera indeks stron wygenerowanych dla tabel i widoków.

*WebGen* generuje skrypty serwera poprzez tworzenie instancji szablonów z wartościami pochodzącymi z modelu PDM. Użytkownik dołącza do modelu fizycznego szablon strony głównej i indeksu oraz definiuje pewne właściwości projektu. Wszystkiego tego dokonuje się w oknie dialogowym „PowerDynamo Model Extended Attributes”. Konfiguruje się tam: ogólne właściwości projektu, profil sieci (ang. Web profile), lokalację witryny internetowej, zbiór szablonów schematu.

Użytkownik musi wybrać typ dostępu do swojej witryny internetowej. Zależnie od typu dostępu (Web access type) można wybrać następujące parametry:

- (a) dla witryny umieszczonej strukturze katalogów na dysku możemy wybrać ścieżkę dostępu,
- (b) dla witryny umieszczonej w bazie danych wybieramy nazwę źródła danych, użytkownika i hasło.

Zbiór szablonów schematu zawiera główny szablon schematu, oraz oddzielne szablony dla każdego typu stron które generujemy. Globalny schemat definiuje zbiór ramek, pasków narzędziowych i tła które nie są specyficzne dla każdego typu strony.

Użytkownik może też utworzyć zmodyfikowany zbiór szablonów schematu. Może to jednak spowodować niezgodność projektu i tła strony.

**Dostrajanie aplikacji przed i po generowaniu.** Przed wygenerowaniem lub ponownym wygenerowaniem aplikacji można ją dostroić i przetestować z widoku drzewa okna Generate PowerDynamo Web Site generowania aplikacji

dokonując:

- (a) modyfikacji właściwości modelu,
- (b) modyfikacji właściwości atrybutów rozszerzonych,
- (c) wyboru kolumn do generowania,
- (d) generowania odnośników hipertekstowych pomiędzy stronami opartymi na tabelach nadrzędnych i podrzędnych (ang. parent i child tables),
- (e) otwarcia wygenerowanej aplikacji WWW w przeglądarce.

Możliwa jest modyfikacja właściwości wszystkich obiektów z widoku drzewa. Istnieje również możliwość modyfikowania z widoku drzewa właściwości aplikacji takich jak: nazwa, tytuł, szablon, biblioteka, profil bazy danych.

Jeżeli wygeneruje się witrynę osadzoną w bazie danych, można ją przetestować z widoku drzewa. Należy tylko uruchomić witrynę z *WebGen*, oczywiście pod warunkiem, że uruchomiony jest serwer *PowerDynamo*. Program instalacyjny *AppModelera* określa, które przeglądarki są powiązane z plikami HTML w rejestrze. Dla generowanych witryn *WebGen* otwiera przeglądarkę domyślną. Jeżeli rejestr nie wskazuje preferowanej przeglądarki, to należy wyznaczyć przeglądarkę, za pomocą której chcemy przeglądać generowane witryny.

**Wybór elementów do generowania.** Użytkownik może generować aplikacje WWW z obiektów, które wybiera się na podstawie obiektów zdefiniowanych w modelu PDM, podobnie jak dokonujemy tego w *PBGen*. Widok drzewa w oknie Generate PowerDynamo Web Site pokazuje elementy, które mogą zostać wygenerowane.

Drzewo zawiera takie elementy jak: obiekt strony głównej, obiekt strony indeksu, obiekty stron dla tabel i widoków, które wybiera się graficznie w modelu PDM lub obiekty stron dla tabel i widoków, które wybiera się za pomocą filtrów, pola obiektów oparte na kolumnach, związki pomiędzy tabelami rodzica i potomka.

Podobnie jak w *PBGen*, istnieją różne sposoby modyfikacji wstępnego wyboru obiektów w oknie Generate PowerDynamo Web Site.

Aby zmodyfikować obiekt, możemy: ustawić flagę generowania obiektów w widoku drzewa, dodawać lub usuwać obiekty z widoku drzewa, zastosować filtr.

**Definiowanie stron i pól.** Strony są głównym elementem interfejsu projektu witryny. Każda strona jest dokumentem STM zawierającym skrypt HTML wygenerowany przez *WebGen*. Dokumenty STM generuje się z poszczególnych tabel i widoków. Istnieje również możliwość generowania dokumentów STM jako plików dyskowych.

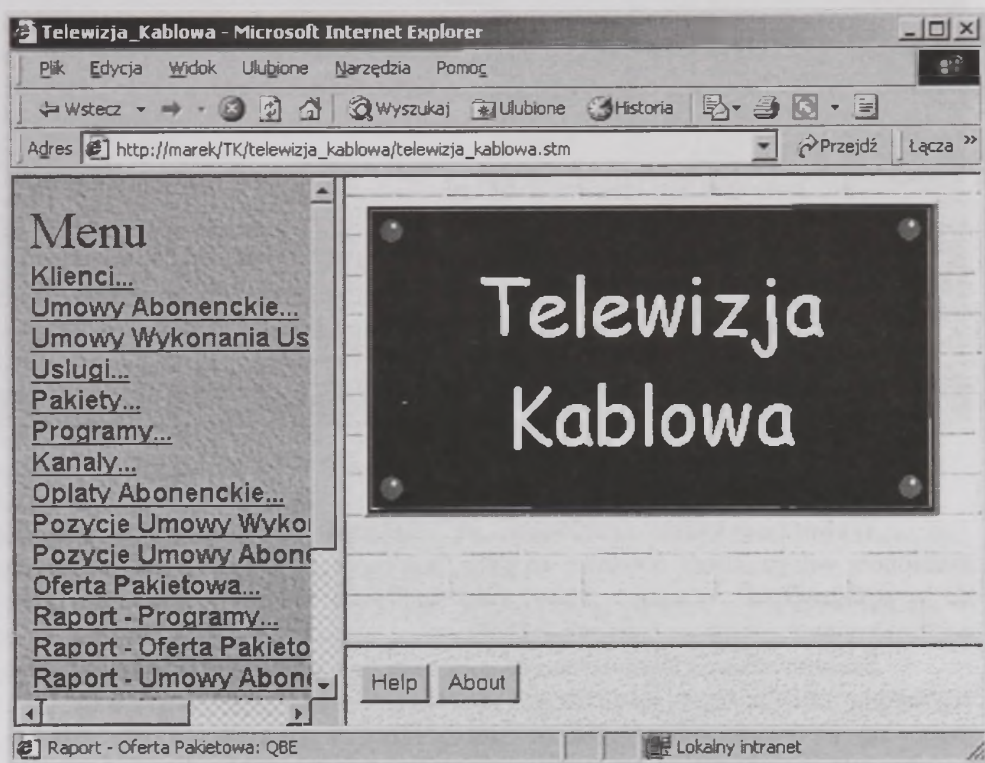
Pola są elementami stron, które odpowiadają kolumnom bazy danych. Szablony pól są używane do definiowania kontrolki lub zbioru kontrolki, które potrafią wyświetlić dane z wierszy bazy danych. Przykładami kontrolki w aplikacjach są pola tekstowe oraz przyciski. Dla każdej strony i widoku w modelu PDM można zdefiniować ogólne właściwości.



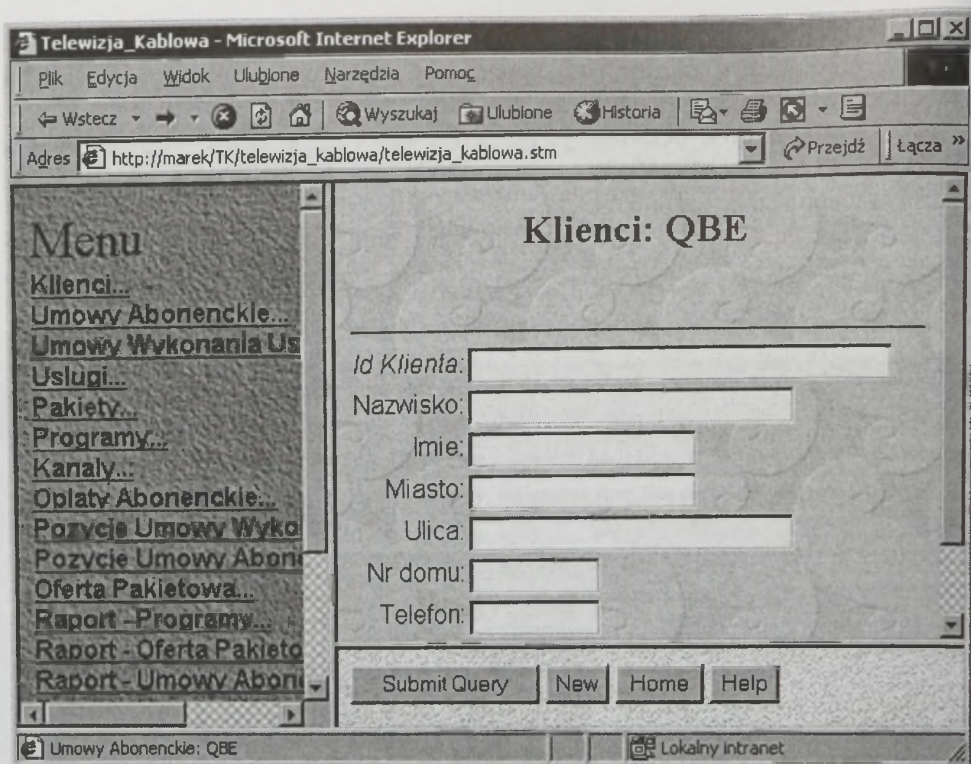
Dla każdej tabeli lub widoku, można używać standardowych właściwości strony, albo można zdefiniować specyficzne właściwości. Ponadto dla każdej tabeli lub widoku można wybrać szablon stron.

Szablon stron determinuje typ stron, które mają zostać wygenerowane z modelu fizycznego. *AppModeler* generuje następujące typy stron: strony QBE, strony tabelaryczne, formularze.

Na rysunkach 20 i 21 przedstawiono stronę główną z menu oraz stronę QBE „Klienci...” wygenerowanej przez *AppModeler for PowerBuilder* przykładowej aplikacji.



Rys.20. Wytworzona aplikacja PowerDynamo



Rys.21. Strona QBE „Klienci...”

**Wykorzystywanie szablonów.** W narzędziu *AppModeler* używa się szablonów witryn, stron, obiektów modelu fizycznego i atrybutów rozszerzonych do wygenerowania witryny i stron. Pola szablonów definiują kontrolki, które zostaną wygenerowane na stronach HTML.

Szablon strony głównej generuje pliki dla strony głównej i umieszcza je w wybranym folderze lub w bazie danych. Szablon strony indeksu generuje strony z menu zawierającym odniesienia do stron, które wygenerowano dla tabel i widoków. Szablon ten zapewnia także powiązania z zasobami kopiowanymi przez *WebGen* takimi jak bitmapy itp.

Szablony stron Query By Example są niezbędne do wygenerowania stron QBE dla tabel i widoków. Szablony pól definiują kontrolki wyświetlane na stronach HTML.

Szablon *CHKLIST.SCT* generuje standardowe kontrolki typu lista rozwijana. Szablon typu checkbox generuje standardowe dwustanowe pola typu „check” na stronach formularzy.

Za pomocą obiektów osadzonych można umieszczać na stronach formularzy wszystkie obiekty, które są wspierane przez przeglądarki stron HTML. Przykładowo, mogą to być kontrolki ActiveX.

Za pomocą *WebGen* można też wykreować własne szablony projektów i



szablony stron. W tym celu mogą być wykorzystane istniejące już projekty witryn. Informacje o ścieżkach szablonów zostaną wpisane do rejestru bezpośrednio w *WebGen*.

**Składnik *MetaWorks*.** *PowerDesigner MetaWorks* jest narzędziem zapewniającym koordynację pracy grupowej w złożonych projektach. Zawiera on centralny słownik (repozytorium) i tym samym oferuje scentralizowane zarządzanie modelowaniem informacji. Umożliwia on łatwe przeglądanie modeli procesów, danych, podmodeli, zapewniając spójność definicji obiektów. *MetaWorks* pozwala również administrować prawami dostępu użytkowników oraz blokować modele zapewniając bezpieczeństwo niezbędne dla dużych projektów.

Do przechowywania informacji o modelach *MetaWorks* używa bazy danych. Dlatego przed składowaniem słownika należy tą bazę najpierw utworzyć, a następnie zdefiniować źródło danych. Źródło danych definiuje połączenie modułu *MetaWorks* z bazą danych. Mimo, że *MetaWorks* może znaleźć istniejące źródło danych, należy zdefiniować nowe źródło danych dla utworzonej bazy danych. Dopiero teraz można przystąpić do definiowania tabel słownika, które zapewniają strukturę niezbędną do przechowywania informacji o modelach.

*MetaWorks* jest programem pracującym w architekturze klient-serwer. Serwer jest centrum aplikacji. Na serwerze składowany jest słownik, który gromadzi i wysyła informacje dotyczące projektów. Informacjami znajdującymi się w słowniku zarządza system zarządzania relacyjną bazą danych (RDBMS). Dostęp do słownika jest realizowany za pomocą klienta poprzez sterowniki ODBC. Klient zgłasza zapytania do serwera aby dokonać konsolidacji (składowanie modelu w słowniku) albo ekstrakcji (pobranie modelu ze słownika). Klient *MetaWorks* może działać na platformach Windows NT/2000, Windows 95/98. Moduły projektowe *PowerDesigner* są instalowane na komputerach klientów, z których mogą się one połączyć ze słownikiem.

Centralny słownika w architekturze klient-serwer ma liczne zalety. Umożliwia pracę grupową, zapewnia integralność danych i spójność danych oraz stwarza możliwość prawdziwej pracy grupowej.

Słownik *MetaWorks* centralizuje dane projektowe dla wszystkich systemów informacyjnych i przez to zapewnia spójność i kompatybilność różnych aplikacji. Rzadko zdarza się, aby w jakiejś organizacji aplikacje działały niezależnie. Niezależne operacje w dużej mierze prowadzą do niespójności danych, a to zwykle jest przyczyną załamywania się systemu informatycznego. Słownik *MetaWorks* pomaga zespołowo zarządzać definicjami danych i dziedzin.

**Zasady prowadzenia projektu.** *MetaWorks* wprowadza pewną strukturę składającą się z projektów, modeli i podmodeli. Aplikacja jest podzielona na projekty. Projekt jest globalnym szkieletem na którym można przechowywać model. Projekt może zawierać kilka modeli konceptualnych (CDM), modeli fizycznych (PDM), modeli *WarehouseArchitect* (WAM) oraz modeli *ProcessAnalyst* (PAM) wraz z ich podmodelami i to w wielu wersjach. Każdy model zawiera dane związane z konkretną częścią aplikacji i jest często dzielony na podmodele. Kierownik projektu lub jego administrator może przypisać



projektantom prawa dostępu do podmodeli. Poszczególne podmodele zawierają część danych w modelu. Projektant może być odpowiedzialny z jeden lub wiele podmodeli.

Słownik *MetaWorks* jest narzędziem do zarządzania procesem wytwarzania aplikacji w zespole. Dostarcza on następujących funkcjonalności:

- (1) tworzenie kont i profili dla użytkowników słownika,
- (2) tworzenie projektów, modeli, podmodeli i przypisywanie ich użytkownikom,
- (3) blokowanie i odblokowywanie modeli,
- (4) ekstrakowanie i konsolidacja modeli i podmodeli,
- (5) porównywanie definicji obiektów i zarządzanie konfliktami danych,
- (6) zadawanie zapytań do słownika,
- (7) modyfikowanie obiektów w słowniku.

W *MetaWorks* istnieją trzy poziomy interakcji ze słownikiem. Sposób w który użytkownik komunikuje się ze słownikiem zależy od tego, czy jest on administratorem, kierownikiem projektu czy też projektantem.

*MetaWorks* jest bardzo elastyczny co do odpowiedzialności za konsolidację, ekstrakcję i zarządzanie konfliktami danych. Elastyczność ta wymaga jednak rygorystycznej organizacji procesu projektowania, szczególnie w kwestii zadań i przydzielania odpowiedzialności. Zaleca się stosowanie jednej z dwóch przedstawionych struktur organizacyjnych:

- (a) jeden administrator konsoliduje i ekstrahuje modele i rozwiązuje konflikty danych,
- (b) każdy projektant konsoliduje i ekstrahuje modele samodzielnie, natomiast konflikty danych projektanci rozwiązują wspólnie.

Konsolidowane modele, które zawierają dane globalne składają się zazwyczaj z podmodeli i podprocesów.

Projektanci pracują nad podmodelami, które są ekstrakowane ze słownika. Kiedy praca jest kończona, zmodyfikowane podmodele są w słowniku konsolidowane. W ten sposób dane globalne są stale uaktualniane najnowszymi modyfikacjami podmodelu.

Istnienie danych globalnych wymaga jednak zarządzania modyfikacjami na nich dokonywanymi. Wymagane jest rozwiązywanie konfliktów modyfikacji tych samych danych. Istnieją dwa sposoby zarządzania modyfikacjami:

- (a) administrator definiuje wspólne dane i zarządza modyfikacjami,
- (b) administrator przypisuje odpowiedzialności dla różnych zbiorów globalnych danych różnym projektantom, zależnie od rozmiaru projektu.

Podczas procesu projektowania administrator albo kierownik projektu może modyfikować model globalny używając słownika.

Za pomocą modułu *PowerDesigner ProcessAnalyst* projektuje się struktury procesowe (przepływu informacji) w postaci diagramów przepływu danych. Modele *ProcessAnalyst* są proste w budowie i zarządzaniu oraz ułatwiają komunikację pomiędzy uczestnikami cyklu projektowego aplikacji. Dwukierunkowe połączenie pomiędzy *ProcessAnalyst* i *DataArchitect* pozwala na

koordynację modeli procesów i logicznych modeli danych. *ProcessAnalyst* umożliwia ponadto kompletną definicję przepływu danych za pomocą: procesów, składnic danych, terminatorów, przepływów danych, przepływów sterowania oraz definicji danych i atrybutów. Analizy funkcjonalnej można dokonywać używając metodologii (i notacji) OMT, Yourdon/DeMarco, SSADM, oraz Gane&Sarson z możliwością dekompozycji procesów na podprocesy.

*DataArchitect* jest podstawowym modułem zestawu narzędzi *PowerDesigner* umożliwiającym modelowanie danych i optymalizację struktury bazy danych. Projektowanie dwupoziomowe umożliwia podział modelowania danych na poziom konceptualny i fizyczny. W konceptualnym modelu danych CDM projektant używa encji oraz zależności pomiędzy nimi do opisanie bazy danych w sposób nietechniczny. Po stworzeniu modelu konceptualnego *DataArchitect* automatycznie przekształca CDM na fizyczny model danych PDM, oparty na platformie bazy danych wybranej przez projektanta. Podczas tej operacji automatycznie przekształcane są wszystkie zależności zdefiniowane w modelu logicznym. Fizyczny model danych można znormalizować. Duże modele mogą być rozkładane na mniejsze podmodele, aby łatwiej zarządzać projektem. Każda modyfikacja danych na poziomie podmodelu jest przenoszona do globalnego modelu danych i na odwrót. *Data Architect* daje także możliwość rewersowania. Polega ono na stworzenie modelu z istniejącej, fizycznej bazy danych, umożliwiając reorganizację struktury bazy danych, tworzenie dokumentacji oraz przeprowadzanie migracji do innych systemów DBMS.

*PowerDesigner AppModeler* współpracuje z innymi modułami pakietu umożliwiając szybkie prototypowanie aplikacji klient-serwer. Generuje obiekty aplikacji na podstawie fizycznego modelu danych oraz wybranej biblioteki klas. Zapewnia ścisłą integrację pomiędzy bazą danych i aplikacją kliencką. Definicja rozszerzonych atrybutów takich jak: style edycji, reguły sprawdzania poprawności, maski formatu dodatkowo zwiększa efektywności pracy programistów pracujących z narzędziami 4GL. *AppModeler* promuje używanie wspólnych dla danej organizacji bibliotek klas, co pomaga zapewnić standardową konwencję kodowania i umożliwia wykorzystywanie doświadczeń bardziej zaawansowanych programistów.

Koordynacja pracy grupowej w złożonych projektach może być zapewniona przez *PowerDesigner MetaWorks*. Składnik ten współpracuje ze wszystkimi wymienionymi powyżej modułami, a poprzez specjalne właściwości centralnego słownika, oferuje scentralizowane zarządzanie modelowaniem informacji i umożliwia łatwe przeglądanie modeli procesów, danych, podmodeli, zapewniając spójność definicji obiektów. Dzięki mechanizmom administrowania prawami dostępu użytkowników i blokowania modelu można zapewnić bezpieczeństwo niezbędne dla dużych projektów.

Jakość pakietu *PowerDesigner* łatwo jest ocenić tworząc model fizyczny PDM dla systemu DBMS *Sybase Adaptive Server Anywhere* i generując następnie schemat bazy danych. Po zmodyfikowaniu modelu fizycznego za pomocą narzędzi *AppModeler for PowerBuilder* i *AppModeler for the Web* przez dodanie definicji



atrybutów rozszerzonych dla środowisk *PowerBuilder* i *PowerDynamo* i tym samym dostrojeniu wygenerowanych później aplikacji klienckich, otrzymujemy w pełni funkcjonalne systemy biznesowe.

## 7. Podsumowanie

Integracja baz danych w sieciach intranetowych i w Internecie ma kluczowe znaczenie praktyczne w nowoczesnych systemach z zasobami informacyjnymi. Zagadnienia związane z integrowaniem baz danych w bazodanowe systemy rozproszone ciągle jeszcze należą do technologicznych nowości. Niniejszy artykuł jest próbą całościowej analizy standardów i technologii integrowania baz danych z sieciami intranetowymi i Internetem. Wykonanie tych badań miało duży walor poznawczy i pozwoliło zebrać informacje o standardach integrowania baz danych z sieciami korporacyjnymi i Internetem, a w konsekwencji, dokonać analizy metodologii i metodyk projektowania intranetowych i internetowych aplikacji bazodanowych w tych technologiach.

## Literatura

1. Austin, D., *Poznaj Oracle 8*, MIKOM, 1998.
2. Barker R., Longman, C., *CASE\*Method, Modelowanie związków encji*, WNT, Warszawa 1996.
3. Berg, C., Corba w Javie. *Software*, 3 (1997), 40-41.
4. Bernardyn, J., Gruber J., Jasiński, M., Mirończuk, P., Józwiak, I., Integracja centralizowanych i rozproszonych systemów DBMS i systemów baz danych z sieciami intranetowymi i Internetem, *Raporty Wydziałowego Zakładu Informatyki*, Politechnika Wrocławska, SPR, nr 1, Wrocław 1998.
5. Campbell, T., Why Intranets?, *Intranet Design Magazine* 1998.
6. Comer, D.E., *Sieci komputerowe TCP/IP, T. 1-3, Projektowanie w trybie klient-serwer, wersja BSD*, WNT, Warszawa 1997.
7. Frey, A., Web-To-Database Communication with API-Based Connectivity, *NetWorld-OnLine*.
8. Gruber J., Mazur Z., Środowiska projektowe i programistyczne dla zintegrowanych sieciowo systemów bazodanowych, *Pro Dialog – Programowanie i zastosowania komputerów*, Wydawnictwo Nakom, Poznań 2001.
9. Gruber J., Mazur Z., Technologie projektowe i programistyczne dla aplikacji wielowarstwowych i zintegrowanych sieciowo systemów bazodanowych, *Prace Naukowe Wydziałowego Zakładu Informatyki*, nr 2, *Bazy danych*, Wydawnictwo Politechniki Wrocławskiej, Wrocław 2000.
10. Henederson, K., *Bazy danych w architekturze klient-serwer*, Robomatic, Wrocław 1998.



11. Hettihewa, S., *Active Server Pages 2.0 dla każdego*, Wydawnictwo Helion, Gliwice 1999.
12. Jacobs, J., Narzędzia Microsoftu do integracji baz danych z WWW, *Software*, 9 (1996), 72-73.
13. Kramer, I.M., Microsoft Transaction Server, *Software*, 4 (1997), 46-47.
14. Patchett, C., Wright, G., *CGI / Perl - książka kucharska*, Wydawnictwo Helion, Gliwice 1999.
15. Siegel, J., *CORBA. Fundamentals and programming*, OMG Group, John Wiley & Sons, Inc., Wiley Computer Publishing Group, New York, 1996.
16. Sipe, S., RPC w Windows 95 i Windows NT, *Software*, 5 (1997), 25-26.
17. Stawowski, M., OMG CORBA, *Software*, 8 (1996), 86-88.
18. Sobczyk, Ł., Budowanie Aplikacji w języku Java dla Internetowej Platformy Oracle przy użyciu narzędzia Jdeveloper, *Software*, 2 (2000).
19. Suleiman, L., Ramesh, Ch., *ActiveX - Biblioteka Programisty*, Zakład Nauczania Informatyki MIKOM, Warszawa 1997.
20. Schneider, A., Davis, Ch., Zeit, W., Intranet Architecture, *Intranet Design Magazine*, New York 1999.
21. Teixeira, P., Pacheco, X., *Delphi 4 – podręcznik profesjonalisty*, Helion, Gliwice, 1998.
22. Tittel, E., Stewart, J., *Intranet. Biblia*, Wydawnictwo PLJ, Warszawa 1999.
23. Nicholes, B., *Building Web Database Applications Using Novell Script for NetWare*, <http://www.developer.novell.com/>, 8 (1999).
24. Douglass, C., *The Dominance of the Intranet*, *Intranet Journal*, <http://www.intranetjournal.com/>, 1999.
25. Lang C., Chow, J., *Database Publishing on the Web and Intranets*, <http://www.itknowledge.com/>, (1) (1996).
26. Rodley, J., *Developing Databases for the Web and Intranets*, <http://www.itknowledge.com/>, 3 (1997).
27. Williams, A., Barber, K., Newkirk, B., *Active Server Pages Black Book*, <http://www.itknowledge.com/>, 5 (1998).
28. *Pliki pomocników oraz dokumentacja firmowa w postaci elektronicznej dołączona do następujących pakietów oprogramowania firmy Powersoft: InfoMaker for Windows, PowerDesigner, Power Builder Enterprise, Power++ Enterprise.*
29. *Pliki pomocników oraz dokumentacja firmowa w postaci elektronicznej dołączona do oprogramowania SELECT Enterprise V6.0e firmy SELECT Software Tools, Inc., [www.selectst.com/Products](http://www.selectst.com/Products).*
30. *Pliki pomocników oraz dokumentacja firmowa w postaci elektronicznej dołączona do następujących pakietów oprogramowania firmy Powersoft (Sybase): PowerDesigner, PowerBuilder Enterprise, Software 2.0, 4 (2000) wraz z dołączonym pakietem Sybase Pack.*

## Źródła internetowe i dokumentacja firmowa

- [2831] *Enterprise Edition Application Programming Model*,  
<http://java.sun.com/j2ee/apm/>.
- [2932] *Informacje techniczne o Oracle WebServer*,  
<http://www.us.oracle.com/products/webserver>.
- [3033] *Informacje o Oracle WebServer*,  
<http://www.oracle.com/products/websystem/webserver>.
- [3134] *JavaServer Pages Whitepaper - A Simplified Guide*,  
<http://java.sun.com/>, 12 (1999).
- [3235] *JDBC API Documentation*,  
<http://java.sun.com/>.
- [3336] *Ogólne informacje opakiecie IntraBuilder*,  
<http://www.borland.com/intrabuilder/papers>.
- [3437] *Opis IDC*,  
<http://www.windows.com/IDC/default.asp>.
- [3538] *Opis LiveWire 1.01*,  
[http://www.netscape.com/comprod/server\\_central/product/livewire/index.html](http://www.netscape.com/comprod/server_central/product/livewire/index.html)
- [3639] *Opis ogólny ColdFusion 3.0*,  
<http://www.allaire.com/Products/ColdFusion/30/Overview/index.cfm>.
- [3740] *Opis oprogramowania narzędziowego firmy Borland*,  
<http://www.borland.com/papers/interde.html>.
- [3841] *Opisy oprogramowania firmy Oracle*,  
[http://www.com.pl/index\\_produkty.htm](http://www.com.pl/index_produkty.htm).
- [3942] *Opis techniczny NetDynamics 4.0*,  
[http://www.netdynamics.com/product/overview/nd400\\_overview.htm](http://www.netdynamics.com/product/overview/nd400_overview.htm).
- [4043] *Oprogramowanie demonstracyjne Oracle*,  
<http://www.oracle.com/products/trial/html/trial.html>.
- [4144] *Oracle CDM Advantage, Release 1.0 for Windows 3.1/Windows 95/Windows NT*, Oracle Corporation, Release 2.0.1, November, 1996.
- [4245] *Oracle Developer Guidelines for Building Applications - Release 6.0*, Oracle Corporation.
- [4346] *Oracle Method - Custom Development - Requirements Modelling using Designer/2000*, CDM Standards and Guidelines Library, Volume 1, Release 2.1.0, October, 1996.
- [4447] *Oracle Web Application Server*, Oracle Press, 1995.
- [4548] *Strategia firmy Borland - dostęp do baz danych w sieciach rozległych*,  
<http://www.borland.com/about/papers/bgg>.
- [4649] *Technologia NDO dynamicznych stron WWW Novell*,  
<http://www.developer.novell.com/>.
- [4750] *The ASP Developer's Site*, BitShop, Inc.,  
<http://www.aspdeveloper.net/>.

## DESIGN TECHNOLOGIES AND SOFTWARE TOOLS FOR BUILDING DATABASE SYSTEMS IN THE NETWORK ENVIRONMENT

Diversity in hardware and software is a fact of life, and network environment is becoming more diverse, not less as computers evolve. In the information systems powerful servers working in teams serve massive databases and surprisingly capable desktop machines provide an increasingly complex array of services, and interact with each other and with bigger systems to compute information in complex information systems. Varied types of computer and network hardware and operating systems are needed to get work done, but the complexity is difficult to deal with. Developing information and database systems by the mean of multi-layer application technologies is making this possible. This paper covers methodologies, techniques and tools that will be fundamental to programmers developing significant database applications in distributed environments. Also, the paper presents some of the typical problems found in software engineering, that can be solved by an approach based on middleware technologies, giving two examples of the technology.

Jacek GRUBER, Zygmunt MAZUR  
Politechnika Wrocławska, Wydział Informatyki i Zarządzania,  
Wydziałowy Zakład Informatyki  
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław  
e-mail: gruber@ci.pwr.wroc.pl, mazur@ci.pwr.wroc.pl





# EKSPLORACJA DANYCH JAKO ELEMENT SYSTEMU WSPOMAGAJĄCEGO PODEJMOWANIE DECYZJI

Iwona ISKIERKA

**Streszczenie:** W pracy przedstawiono sposoby wyboru właściwego systemu baz danych ze szczególnym uwzględnieniem hurtowni danych jako elementu systemu wspomagającego zarządzanie firmą.

## Wstęp

W latach siedemdziesiątych firma Ashton Tate wprowadziła na rynek komputerowy aplikację dBase, która stała się przełomem dla użytkowników komputerów. Mianowicie dano użytkownikowi komputera możliwość wykonywania własnoręcznie operacji na dość dużych zbiorach danych. Osoba korzystająca z bazy danych mogła porządkować dane, wyszukiwać potrzebne informacje oraz porządkować je. Z biegiem czasu zaczęto wykorzystywać to oprogramowanie przy informatyzacji firm. Początki były skromne. Na każdym komputerze instalowano kompletne oprogramowanie, aby możliwe było pobieranie informacji z dysku serwera plików przez sieć lokalną w trakcie działania programu. Takie rozwiązanie miało wady w sytuacji gdy dane tego samego pliku podlegały przetwarzaniu równocześnie u kilku użytkowników. Uaktualnienie zawartości zbioru następowało dopiero po zakończeniu jednej z sesji. Szukano rozwiązań tego problemu. Małe komputery połączone w lokalne sieci z niewielkimi serwerami nie zapewniały również bezpieczeństwa danych. Dynamiczny rozwój firm wymuszał rozwiązania bardziej elastyczne. Takim rozwiązaniem okazała się architektura typu klient/serwer. Pomysł był prosty – przeniesiono narzędzia do wyszukiwania, uaktualniania i opracowywania danych do serwera.[5] Dzięki temu w firmach możliwe stały się rozwiązania informatyczne w których obsługuje się wielu użytkowników pracujących równocześnie z tymi samymi danymi. Można powiedzieć więc, że wszelkie systemy do zarządzania przedsiębiorstwami są w pewien sposób „nakładkami” na serwery baz danych.

## 1. Hurtownia danych jako element systemu wspomagającego zarządzanie firmą

W pewnych sytuacjach, gdy firma dynamicznie rozwija się, prowadząc wiele rodzajów działalności, staje przed koniecznością zaimplementowania znacznie bardziej wydajnego niż zwykła baza danych rozwiązania. W każdej firmie gromadzi się dane które odpowiednio wykorzystane stają się cennym jej zasobem. Skuteczne kierowanie firmą wiąże się z umiejętnością podejmowania właściwych strategicznych decyzji. Aby takie decyzje były trafne i przynosiły firmie wymierne korzyści powinno mieć się szybki dostęp do bardzo dużej ilości

danych, a także dostęp do danych przekrojowych, bowiem jakość podejmowanych decyzji w dużej mierze zależy od jakości dostępnej informacji.

System wspomagający podejmowanie decyzji jest systemem dostarczającym informacje, w oparciu o którą użytkownicy są w stanie dokonać analizy danej sytuacji i w zależności od wyniku analizy, podjąć odpowiednią decyzję.[1] Może to być długoterminowa decyzja strategiczna dotycząca wprowadzenia na nowy rynek danego produktu bądź krótkoterminowa decyzja taktyczna. Jednak nie wszystkie potrzebne informacje można uzyskać bezpośrednio z operacyjnych systemów przetwarzania danych. W firmach systemy do zarządzania przedsiębiorstwem korzystają z transakcyjnych baz danych, w których są gromadzone informacje niezbędne do obsługi bieżących procesów zachodzących w firmie. Część danych staje się produktem ubocznym. W takiej sytuacji aby optymalnie wykorzystać dane tak cenne dla danej firmy należy je przekształcić do postaci ułatwiającej analizę i równocześnie umożliwić przechowywanie danych „historycznych”. W takiej sytuacji często stosuje się elastyczne rozwiązanie informatyczne takie jak hurtownia danych. Jest to baza danych specjalnie zaprojektowana pod kątem wykonywania analiz danych, przechowująca informacje pochodzące z wielu różnych źródeł, a w efekcie dająca przekrojowy obraz działalności firmy. Okazuje się, że w hurtowni danych właśnie informacje „historyczne” są najbardziej wartościowe i to one właśnie są wykorzystywane przez analityków. Ze względu na możliwość łączenia informacji ze źródeł wewnętrznych i zewnętrznych oraz systematyzowania danych, hurtownia danych może stać się ważną częścią systemu wspomagającego zarządzanie firmą.

Dane do hurtowni danych dostarczane są z baz operacyjnych. Aby na podstawie zgromadzonych informacji odkryć pewne zależności oraz wydobyć szczególnie rodzaj informacji już na etapie wprowadzania danych powinno nastąpić ich porządkowanie, a następnie ich transformacja i załadowanie do hurtowni danych. Dlatego też architekci systemów informatycznych stosują dla hurtowni danych trójstopniową strukturę składającą się z: narzędzi do zbierania informacjami, samej bazy danych oraz rozbudowanych narzędzi analitycznych.

Ważnym składnikiem architektury hurtowni danych jest integrowanie danych z wielu źródeł przed ich wprowadzeniem do hurtowni danych.[3] Jeśli dane pochodzą z wielu systemów, baz danych oraz platform to pewne formy integracji są konieczne.

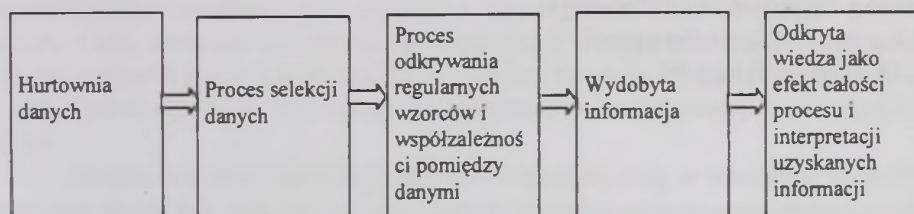
Wynikiem przetwarzania wspomagającego podejmowanie decyzji są informacje analityczne, które następnie wykorzystuje się w procesach podejmowania taktycznych i strategicznych decyzji. W procesach tych wymagane są dane historyczne. Dane takie są wykorzystywane do analiz porównawczych aby możliwe było śledzenie trendów i wzorców informacji w czasie. Narzędzia analityczne umożliwiają zamianę zgromadzonych informacji na wiedzę przydatną do efektywnego zarządzania przedsiębiorstwem.



## 2. Analiza danych biznesowych

Coraz większą popularność jako narzędzie do analizy danych biznesowych zdobywa technologia zwana eksploracją baz danych. Stosując tradycyjne metody analizy danych czasami nie można odnaleźć pewnych specyficznych zależności, które mogą występować pomiędzy danymi. Zależności te „ukryte” w ogromnych zbiorach danych nie są rozpoznawane znanymi metodami. [4] Aby je odkryć są używane specjalne algorytmy w odpowiedni sposób analizujące i przeszukujące bazy danych. Odkrywanie wiedzy i stosowanie odpowiedniego oprogramowania analitycznego tradycyjnie używane było do przetwarzania danych finansowych. Obecnie wykorzystywane jest również do analizy relacji z klientami i dostawcami oraz w e-biznesie. Odkrywanie wiedzy nie jest procesem jednoetapowym. Jest to proces złożony, którego efekty zależą od poprawnego zorganizowania wielu etapów.

Proces odkrywania wiedzy składa się z kilku kroków (Rys.1). W pierwszym z nich należy przygotować dane tak, aby efektywność algorytmu poszukującego zależności była optymalna. Po przygotowaniu danych następuje etap poszukiwania współzależności pomiędzy danymi. Na tym etapie stosuje się wiele różnych technik i narzędzi w zależności od oczekiwań i potrzeb użytkownika. Następnie następuje weryfikacja i prezentacja odkrytej wiedzy.



Rys. 1. Proces eksploracji danych

Eksploracja danych wymaga użycia specjalistycznych narzędzi, pozwalających szybko zauważać współzależności pomiędzy danymi. Do czołowych programów używanych do eksploracji danych zaliczyć należy Discovery Server, Intelligent Miner, Clementine firmy SPSS, Darwin (Oracle). Firmy te dostarczają narzędzi służących do przygotowania danych, ich przetwarzania, sporządzania raportów oraz dystrybucji. Jest możliwość przygotowania danych dostarczanych przez użytkowników z wykorzystaniem tzw. inteligentnych formularzy. Część programów posiada zestaw narzędzi służący do określenia niezbędnej wielkości próby przy założeniu określonej mocy statystycznej wyników. Do przetwarzania danych wykorzystuje się narzędzia służące do rozpoznawania brakujących danych i uzupełnianiu ich wartościami przybliżonymi statystycznie.[6] Do dystrybucji wyników służą narzędzia np. Clementine Solution Publisher, które pozwalają przekształcić model analityczny z procesu data mining w gotowe do użycia aplikacje. Dostępne są także liczne narzędzia uzupełniające np. moduł do raportów tabelarycznych, model regresji oraz inne modele matematyczne i analizy danych. Oracle Darwin udostępnia trzy

podstawowe i najczęściej stosowane metody eksploracji danych. Są nimi: sieci neuronowe, drzewa decyzyjne, automatyczna detekcja klastrów.

Łatwość wykonywania analiz przyspiesza zdobywanie wiedzy o kliencie, zachowaniach rynku oraz sytuacji wewnątrz firmy.

## Literatura

1. Banachowski L.: Bazy danych. Tworzenie aplikacji. Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1998.
2. Hernandez M.J.: Bazy danych dla zwykłych śmiertelników. Edu-Mikom, 1998.
3. Poe Vidette, Klauer P, Brobst S.: Tworzenie hurtowni danych WNT, Warszawa 2000.
4. Głowiński Cezary.: Sztuka wysokiego składowania, PCKurier 2000.
5. Grochowski Tomasz.: Od SAS-a do SAP-a, Obszary zastosowań baz danych, PCKurier 2001
6. Rzewuski Marek.: Programy do analizy danych, PCKurier 2001

Dr inż. Iwona ISKIERKA  
Instytut Ekonometrii i Informatyki  
Politechnika Częstochowska  
ul. H. Dąbrowskiego 69

# ROZDZIELANIE OBOWIĄZKÓW W PROCESACH BIZNESOWYCH

Stanisław KĘDZIERSKI

**Streszczenie:** Znaczna część systemów informatycznych stosowanych w przedsiębiorstwie obsługuje procesy biznesowe (workflow). Ze względu na dużą ilość osób (agentów) definiowane są abstrakcyjne role (pełnione przez osoby lub maszyny), którym przydzielane są obowiązki. Istotne znaczenie dla bezpieczeństwa ma rozdzielanie obowiązków. W referacie przedstawiona zostanie teoria i sposoby rozdzielania ról w systemach baz danych jak i również w procesach biznesowych.

## 1. Bezpieczeństwo informacji w systemach workflow

Warto sobie uprzytomnić znaczenie powiedzenia „żaden system nie jest w stu procentach bezpieczny”. Aby zwiększyć bezpieczeństwo informacji należy określić niebezpieczeństwa na jakie narażone są informacje organizacji. Procedury bezpieczeństwa powinny być rozwijane aby zminimalizować utratę cennych danych. Cena ochrony informacji w organizacji uwzględnia zwykle zależności wartości cennych danych w stosunku do ryzyka ich utraty. Czyli że koszt ochrony danych powinien być równoważony kosztem niewłaściwego wykorzystania danych.

Bezpieczeństwo informacji ogrywa szczególną rolę w otoczeniu workflow zwłaszcza wtedy gdy olbrzymie ilości danych transferowane są codziennie z biurka na biurko i od osoby do osoby. Duża ilość danych w systemach workflow sprawia, że są one podatne na możliwość utraty przez nie danych. Aby temu zapobiec system bezpieczeństwa powinien zapewnić to, że dane są:

- zdatne do użycia,
- integralne,
- użytkowe (mają wartość użytkową),
- autentyczne,
- poufne dla właściwych osób.

Wymienione atrybuty możliwe są do osiągnięcia poprzez implementację usług bezpieczeństwa: legalizacja, kontrola dostępu, poufność, integralność danych oraz spójność całej transakcji [5].

## 2. Strategie kontroli dostępu

Strategie kontroli dostępu składają się z dwu oddzielnych części: taktyki oraz mechanizmów kontroli dostępu. Strategie można rozumieć jako zbiór reguł wysokiego poziomu, które opisują w jaki sposób podejmowane są decyzje o kontroli dostępu [8]. Strategia kontroli dostępu określana jest dla każdego



użytkownika lub grupy użytkowników, czynności które mogą oni wykonywać na obiektach z systemu informacyjnego. Strategia nie określa w jaki sposób reguły powinny być przestrzegane, jest to zadaniem mechanizmów kontroli dostępu [1].

Mechanizmy kontroli dostępu mogą być uważane za podstawowy poziom działań sprzętowych lub programowych ustanowionych dla wypełnienia wymagań poziomu taktyki. Mechanizmy kontroli dostępu ulegają ciągłemu rozwojowi aby sprostać wymaganiom z poziomu taktyki. Przedstawione zostaną trzy sposoby kontroli dostępu do danych: dyskrecjonalny (dowolny), ustawowy oraz zorientowany na role (RBAC).

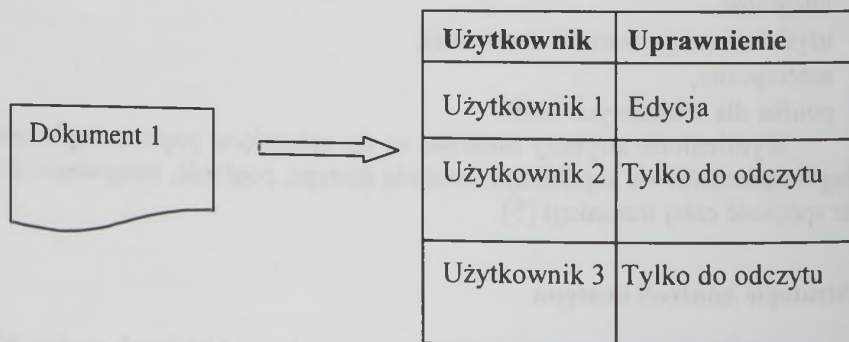
## 2.1. Dyskrecjonalna kontrola dostępu

Dyskrecjonalna (uznaniowa) kontrola dostępu administruje dostępem użytkownika do cennych informacji w organizacji. Dzieje się to zgodnie z poświadczoną identyfikacją i uprawnieniami dostępu (np.: czytania, pisania lub wykonywania) przydzielonymi w odniesieniu do poszczególnych obiektów [8].

Tablica 1. Macierz dostępu

Użytkownicy		Obiekty		
		Dokument 1	Dokument 2	Dokument 3
Użytkownik 1	Edycja	Edycja		
Użytkownik 2	Tylko do odczytu	Tylko do odczytu	Edycja	
Użytkownik 3	Tylko do odczytu		Edycja	

Jednym z rodzajów mechanizmu kontroli dostępu jest macierz dostępu. Wadą macierzy dostępu są jej rozmiary w przypadku dużej organizacji oraz to, że wiele pól może być pustych. Dlatego też macierz dostępu bardzo rzadko implementowana jest w postaci macierzy ale jako zbiór list kontroli dostępu. Powstają one poprzez wycinanie kolumn z macierzy i zawierają uprawnienia dostępu użytkowników do obiektu [3].



Rys. 1 Lista kontroli dostępu

Listy kontroli dostępu są łatwiejsze do obsługi i przechowywania uprawnień użytkowników względem określonych obiektów. Jeśli nowy użytkownik

potrzebuje uzyskać dostęp do obiektu jego nazwa dopisywana jest do listy. Problemem może być obsługa przypadku sprawdzania dużej liczby obiektów.

## 2.2. Ustawowa kontrola dostępu

W swojej najprostszej postaci ustawowa (obowiązkowa) kontrola dostępu przydziela każdemu użytkownikowi i obiektowi poziom bezpieczeństwa. Im bardziej cenne informacje obiekt przechowuje tym wyższy poziom bezpieczeństwa posiada, zaś im bardziej godny zaufania użytkownik tym wyższy poziom jego bezpieczeństwa. Obiekty posiadają poziomy klasyfikujące bezpieczeństwo, zaś użytkownicy przydzielani są do poziomów nazywanych *certyfikatem bezpieczeństwa*.

Poziomy bezpieczeństwa przydzielane są obiektom i użytkownikom. Obowiązywać może przykładowa hierarchia: ściśle tajne (ST), tajne (T), poufne (P) i nie klasyfikowane (N) [8]. Zgodnie z tą hierarchią uprawnienia użytkownika są mu przydzielane tylko wtedy gdy jego certyfikat bezpieczeństwa oraz poziom bezpieczeństwa obiektu spełniają określoną regułę ustawowej kontroli dostępu.

Reguła *czytania w dół* oznacza dla użytkownika możliwość odczytu obiektu tylko wtedy jeśli jego certyfikat bezpieczeństwa jest wyższy lub co najmniej równy poziomowi bezpieczeństwa czytanego obiektu. Przykładowo tylko użytkownicy z certyfikatami ST lub T mogą czytać obiekty o poziomie T. Reguła *pisania w górę* mówi, że użytkownik może zapisywać tylko obiekty których poziom bezpieczeństwa jest większy lub równy certyfikatowi bezpieczeństwa użytkownika. Przykładowo użytkownik z certyfikatem P może zapisywać obiekty o poziomie P lub wyższym.

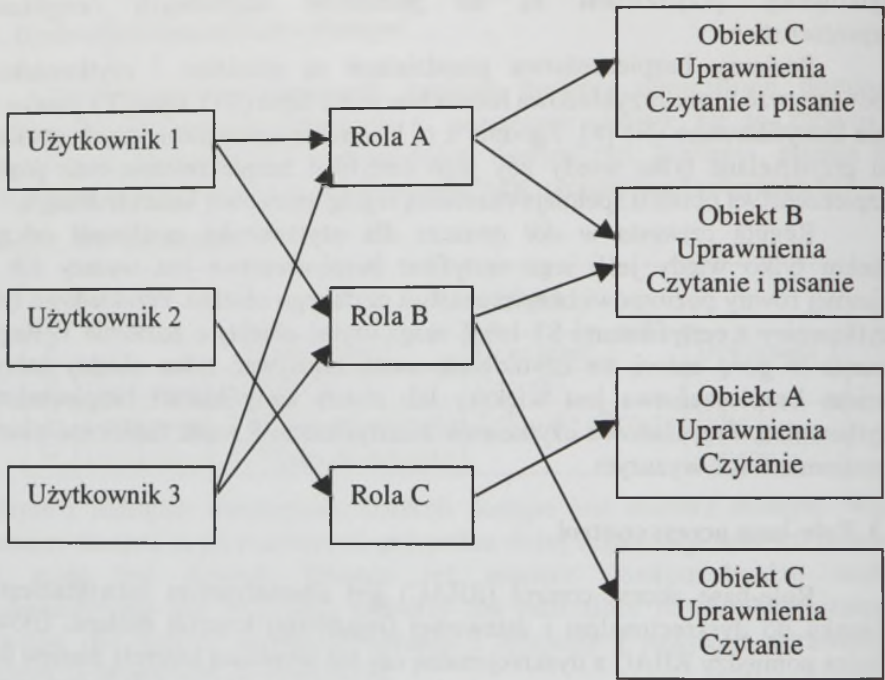
## 2.3. Role-base access control

Role-base access control (RBAC) jest alternatywnym rozwiązaniem w stosunku do dyskrecjonalnej i ustawowej (regułowej) kontroli dostępu. Główna różnica pomiędzy RBAC a dyskrecjonalną czy też ustawową kontrolą dostępu tkwi w tym, iż RBAC łączy uprawnienia dostępu z rolami pełnionymi przez użytkownika. Aby użytkownik mógł otrzymać uprawnienie musi być przydzielony do roli, która posiada określone uprawnienia. Strategie dyskrecjonalnej i ustawowej kontroli dostępu były ściśle określone natomiast strategia RBAC zależy od skonfigurowania jego elementów (ról, hierarchii, ograniczeń). RBAC jest bardziej elastyczny; możliwa jest zmiana przydziału agenta z jednej roli do drugiej w zależności od obowiązków agenta w organizacji. Tak samo uprawnienia dostępu mogą być przydzielane i cofane rolom w miarę rozwoju systemu informatycznego.

Najprostszy model RBAC składa się z trzech części: użytkowników, ról oraz uprawnień. Użytkownik jest osobą korzystającą z systemu. Rola definiowana jest jako stanowisko w organizacji, które uczestniczyć może w procesach biznesowych oraz odpowiada za pewne czynności. Uprawnienia oznaczają pewien szczególny zbiór przywilejów dostępu do zbioru obiektów z danymi w systemie. Jeden użytkownik może pełnić wiele ról i tak samo do jednej roli może być

przydzielonych wielu użytkowników. Tak samo rola może posiadać wiele uprawnień oraz jedno uprawnienie może być przydzielone wielu rolam. Te powiązania wiele do wiele stanowią o elastyczności systemu RBAC. Zastosowanie roli jako pośrednika pomiędzy użytkownikiem a uprawnieniem stwarza możliwość większej kontroli nad konfigurowaniem uprawnień użytkowników. Rysunek 2 przedstawia związki pomiędzy użytkownikami a uprawnieniami.

Większość organizacji posiada zhierarchizowaną strukturę. Tak jak role przydzielane są użytkownikom (osobom) będącym częścią organizacji tak możliwe jest istnienie hierarchii pomiędzy rolami. Wykorzystanie hierarchii ról pozwala systemowi RBAC na lepsze odwzorowanie struktury organizacyjnej.



Rys. 2 relacje użytkownik – uprawnienie w systemie RBAC

Role-based access control (RBAC) jest obecnie najbardziej popularnym sposobem administrowania kontrolą dostępu w systemach workflow. RBAC odgrywa także podstawową rolę w badaniach nad rozdzielaniem obowiązków.

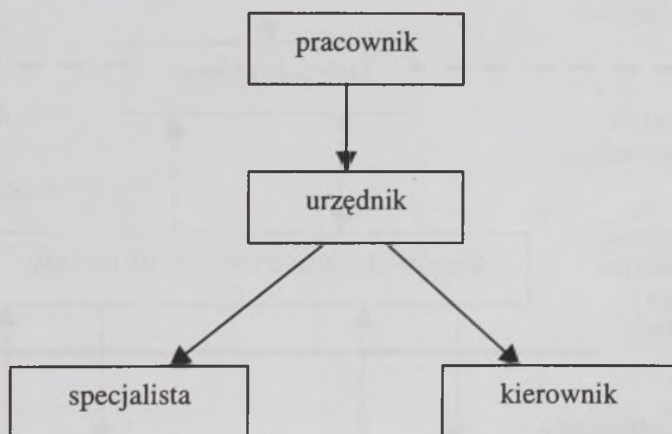
Role-based access control (RBAC) stosuje metodę abstrakcji do pojęcia roli aby móc później określić wymagania odnośnie dostępu. Podczas administrowania rolą uprawnienia przydzielane są do ról i agentów (agenci przyporządkowywani są do ról). Upewnienie mówi o zdolności do wykonania pewnych czynności. Przykładami uprawnień mogą być; złożenie zamówienia, zaaprobowanie zamówienia. W danej organizacji agent przydzielony do roli otrzymuje uprawnienia i obowiązki do spełnienia. Administrowanie rolami może się wtedy ograniczyć do właściwego ustawienia roli w hierarchii ról. Hierarchia ról



porządkuje role według relacji starszeństwa zachodzącej pomiędzy rolami. Dzieje się to zgodnie ze strukturą organizacyjną danej instytucji. Rysunek. 3 pokazuje hierarchię ról.

### 3. Zasada rozdzielania ról

Podstawowym celem zasady rozdzielania obowiązków jest zapobieganie kradzieżom lub możliwości popełnienia błędu. W określonym procesie biznesowym cele te można osiągnąć poprzez rozdział zadań i skojarzonych z nimi uprawnień pomiędzy wielu uczestników tego procesu. Klasycznym przykładem, na którym prezentowana jest zasada to konieczność złożenia dwu podpisów na czeku. Automatyzowanie prac biurowych staje się coraz to łatwiejsze wraz z wprowadzaniem sieci komputerowych. W latach dziewięćdziesiątych ubiegłego stulecia komputerową implementacją procesów biznesowych stały się systemy workflow. Systemy zarządzania workflow umożliwiają definiowanie, zarządzanie oraz wspieranie wykonywania procesów biznesowych za pomocą komputera. Wzrost zastosowań mediów elektronicznych w zarządzaniu przedsiębiorstwem powoduje istotne podniesienie jakości i szybkości przetwarzania informacji. Niestety (tak jak i w wielu innych dziedzinach) sprawia to możliwość poniesienia strat a w szczególności zwiększenia ryzyka bezpieczeństwa danych lub pieniędzy.



Rys. 3. Przykładowy fragment hierarchii ról

Bezpieczeństwo informacji wymaga systemu, który dostarczałby pięć podstawowych usług: legalizacja (legalizowanie), sterowanie dostępem, integralność, poufność oraz nie usuwalność.

Integralność systemu może być rozumiana na wiele sposobów:

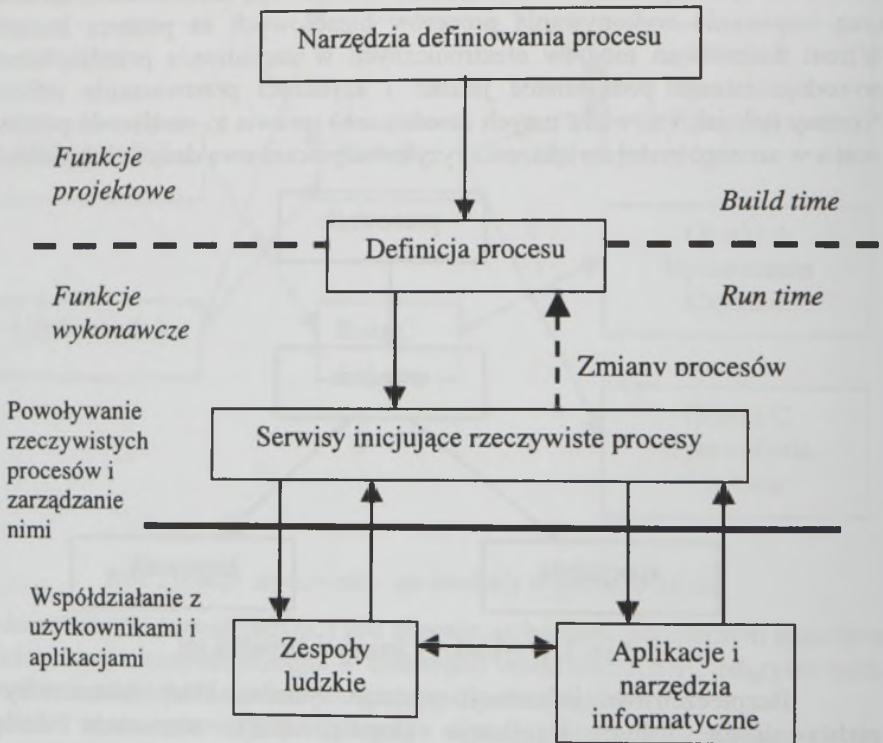
- operacyjna, odnosząca się do równoczesnego dostępu do danych i którą można uzyskać poprzez kontrolę współbieżności,
- fizyczna, gwarantująca ochronę przed utratą danych, uzyskiwana przez stosowanie sum kontrolnych, technik kryptograficznych,

- semantyczna, określająca zgodność danych z odpowiednimi regułami biznesowymi (gospodarczymi), osiągana podczas próby zmiany danych co powinno być zgodne z przyjętymi regułami.

Przykładową regułą biznesową jest reguła wymagająca separację (rozdzielanie) obowiązków. Pierwotnym celem stosowania tej reguły było zapobieganie kradzieżom i błędom czyli zapewnienie integralności semantycznej. Często reguły biznesowe formułowane są następująco: "na czeku muszą widnieć dwa różne podpisy". Reguła ta wymusza współpracę systemu w potwierdzaniu autentyczności klienta oraz ogranicza prawa dostępu w danym czasie. Ażeby móc mówić o regule separacji obowiązków należy wprowadzić pojęcie roli.

### 3.1. Środowisko systemów workflow oparte o role

Na rysunku 4 przedstawione są trzy główne obszary systemów workflow: administracyjny, zarządzania i interakcji.



Rys. 4. Model działania systemu workflow

Obszar administracyjny odpowiada za definiowanie procesów biznesowych. Proces biznesowy składa się z zadań rozumianych jako pewne umowne (małe) elementy pracy. Ludzie mogą wykonywać zadania jak i również zautomatyzowane zadania mogą być realizowane przez komputery. Przyjmuje się, iż za wykonanie zadania odpowiedzialny jest pojedynczy agent (osoba, dział,

oddział). Definicją zadania jest wyrażenie przedstawione w języku akceptowalnym przez komputer.

Proces biznesowy zdefiniowany jest jako zadania połączone w graf skierowany, w którym warunki wykonania zadań występują na łukach. Odpowiednio definicją procesu biznesowego jest wyrażenie przedstawione w języku akceptowalnym przez komputer. Podczas wykonywania procesu biznesowego warunki są sprawdzane oraz podejmowana jest decyzja, które zadania są wykonywane w następnej kolejności.

Separacja obowiązków definiowana jest w obszarze reguł biznesowych. Przykładowa reguła biznesowa to: „nikt nie może aprobować podpisanego przez siebie czeku”.

Obszar zarządzania systemów workflow odpowiada za interpretowanie zdefiniowanych procesów biznesowych i tworzenie wystąpień procesów w tych momentach w których proces ma być wykonany. Wystąpienie procesu można rozumieć jako obiekt przedstawiający sobą określony przebiegający proces. Każde wystąpienie procesu będzie więc przechowywało informacje typowe dla tego specyficznego wystąpienia procesu. Przykładowymi informacjami mogą być: wyzwalacz procesu, czas rozpoczęcia, zadania do wykonania, informacje mogące mieć wpływ na wykonanie procesu. Podczas wykonywania procesu badany jest jego stan szacowane są warunki, które będą wyznaczać ścieżkę zadań do wykonania. Każde wystąpienie procesu składa się z jednego lub więcej wystąpień zadań. Na poziomie zarządzania należy również zdecydować, którzy agenci będą mogli wykonać zadanie. Ażeby można podjąć taką decyzję na poziomie zarządzania oceniane są wymagania odnośnie separacji obowiązków określonych na poziomie administracyjnym. Określają one nie tylko co będzie wykonane jako następne zadanie ale także który z agentów mógłby je wykonać. W przypadku procesu zlecenia zakupu wyznaczony byłoby tylko jeden agent do zadania aprobaty zamówienia. Co więcej zostałby on wyłączony z grupy ewentualnych wykonawców następnych zadań w tym wystąpieniu procesu.

Poziom interakcji koncentruje się głównie na faktycznym wykonywaniu zadania przez agenta. Implementacja wykorzystuje listę zadań w celu przedstawienia agentowi zadań, które będzie musiał wykonać. Lista ta może być stałą lub też może się zmieniać dynamicznie w miarę postępu prac. Zawsze kiedy agent próbuje wykonać zadanie z listy środowisko interakcji musi potwierdzić czy agent jest autoryzowany do wykonania danego zadania.

Środowisko interakcji może wywoływać inne aplikacje (zadania) które wymagane są do zrealizowania zadania. System powinien sprawdzić czy wywoływana aplikacja posiada w danym momencie stosowne prawa dostępu. Prawa dostępu nie powinny być nadużywane poza kontekstem zadania. Przykładowo należy zapewnić to, iż autoryzacja „aprobaty zlecenia” albo dozwolenia nie jest możliwa dla osoby w momencie zainicjowania zlecenia.



## 4. Rodzaje separacji obowiązków

Ogólnie stosowaną zasadą jest rozdział obowiązków, co oznacza, że nikt w systemie nie posiada wystarczających pełnomocnictw, aby móc samemu przygotować i popełnić kradzież. Zasadę tę realizuje się to poprzez rozdzielanie większych czynności (zadań) na małe kroki wykonywane przez różnych agentów (osoby). Powszechnie przyjmuje się, że istnieje podział na dwie kategorie zasady separacji obowiązków: statyczną i dynamiczną [2]. Podstawą takiego podziału było to, iż decyzje odnośnie statycznej separacji obowiązków podejmowane są podczas budowy systemu zaś decyzje dynamicznej separacji obowiązków podejmowane są podczas wykonywania procesu. Pojęcie „czas budowy” implikuje, że zasada statycznej separacji obowiązków jest stała w przydzielaniu uprawnień do poszczególnych ról. Zasada dynamicznej separacji obowiązków steruje dostępem do uprawnień ról przydzielanych podczas realizacji procesu (runtime).

### 4.1. Statyczna separacja obowiązków

Statyczna separacja obowiązków jest prostszą formą separacji obowiązków i może być wzmocniona przez przydzielanie agentów do wielu wzajemnie wykluczających się ról [9]. Statyczna separacja obowiązków polega na tym, że agent nie może pełnić dwu wzajemnie wykluczających się ról. Role można uważać za wzajemnie wykluczające się jeśli osoba nigdy nie może równocześnie pełnić obu ról. Zaletą statycznej separacji obowiązków jest łatwość jej implementacji. Do słabych stron statycznej separacji obowiązków zaliczyć można trudność w spełnieniu potrzeb sterowania dostępem organizacji gospodarczych co jest przyczyną pojawienia się szeregu postaci dynamicznej separacji obowiązków.

### 4.2. Dynamiczna separacja obowiązków

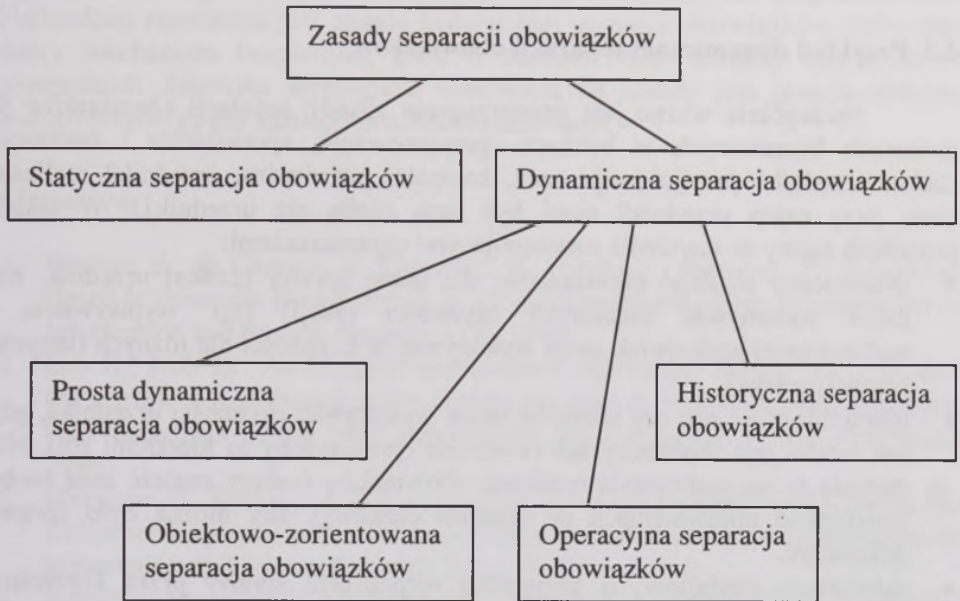
Dynamiczna separacja obowiązków umożliwia większe rozróżnienie oraz dokładniejsze zebranie sposobów sterowania poprzez branie pod uwagę agentów, którzy już uczestniczyli w realizacji procesu workflow przed podjęciem decyzji o separacji obowiązków. Ponieważ dynamiczna separacja obowiązków jest bardziej elastyczna dlatego lepiej nadaje się do zastosowania w organizacjach.

Rysunek 4 przedstawia najbardziej znane rodzaje dynamicznych separacji obowiązków.

*Prosta dynamiczna separacja obowiązków* jest najstarsza i najprostsza. Umożliwia ona agentom pełnić wzajemnie wykluczające się role. Zasada ta nie daje jednak pełnej kontroli nad agentami. Agent pełniący dwie wykluczające się role może obejść tę zasadę kończąc jedną rolę (wylogując się) i rozpocząć następną (logując się).

*Obiektowo zorientowana separacja obowiązków* umożliwia agentowi skorzystać z wielu wzajemnie wykluczających się ról w tym samym procesie. Obiektowo zorientowana separacja obowiązków nie jest skoncentrowana na tym którą rolę agent pełni lecz na możliwej ilości dostępu do obiektów. Jeśli agent może mieć

dostęp do obiektu tylko raz w danym procesie nie jest możliwe aby mógł on nadużyć kombinacji posiadanych uprawnień.



Rys. 4 Postacie zasady separacji obowiązków

**Operacyjna separacja obowiązków** powinna uniemożliwić agentowi wypełnienia dwu wzajemnie wykluczających się ról w procesie biznesowym jeśli ich uprawnienia mogłyby pozwolić agentowi na ukończenie procesu biznesowego. Ten rodzaj separacji obowiązków pokazuje częściowo statyczny charakter albowiem decyzje nie są podejmowane na dynamiczny sposób w czasie wykonywania procesu. Operacyjna separacja obowiązków pozwala na unikanie kradzieży poprzez zapewnienie tego, że konieczna jest obecność co najmniej dwu agentów do zakończenia procesu [4]. Ten rodzaj zasady separacji obowiązków sprawdza się w dużych procesach gdzie występuje wiele uprawnień.

Ostatnie dwie zasady separacji obowiązków nie zapewniają wystarczająco elastycznej ochrony systemom workflow. Obiektowo zorientowana separacja obowiązków nie pozwala agentowi na wykonanie drugiej czynności na obiekcie w danym procesie zaś operacyjna separacja obowiązków nie zezwala agentowi na pełnienie wszystkich ról w procesie biznesowym w stosunku do różnych obiektów występujących w workflow. Aby rozwiązać wymienione problemy wprowadza się historyczną separację obowiązków.

**Historyczna separacja obowiązków** daje możliwość agentowi wypełniać wzajemnie wykluczające się role. Uprawnienia przydzielone do tych ról mogą rozciągać się na akcje całego procesu biznesowego. Jedynym ograniczeniem nałożonym na historyczną separację obowiązków jest to, iż żaden agent nie może wykonywać wszystkich czynności procesu biznesowego na tym samym obiekcie

[9]. Określenie uprawnień agenta w historycznej separacji obowiązków polega na indywidualnych historiach obiektów workflow.

### 4.3. Przykład dynamicznej separacji obowiązków

Szczególnie ważne jest przestrzeganie zasady separacji obowiązków w procesach biznesowych w bankach (przygotowanie, sprawdzenie i realizacja czeku; urzędnik1 przygotowuje czek, kontroler potwierdza, urzędnik2 realizuje czek, przy czym urzędnik2 musi być inną osobą niż urzędnik1). W takich procesach mamy do czynienia z następującymi ograniczeniami:

- dynamiczny rozdział obowiązków; dla danej sprawy (czeku) urzędnik<sub>1</sub> nie może wykonywać niektórych czynności (akcji) (np. wypisywanie i realizowanie) aczkolwiek może wykonywać te czynności dla różnych (innych) spraw (czeków),
- hierarchiczność ról; czy kontroler może wykonywać czynności urzędnika, gdy ten ostatni jest nieobecny lub odwrotnie (przeciwnie to hierarchii ról). Nie zwalnia to od zachowania rozdziału obowiązków (należy znaleźć inną osobę [również o uprawnieniach do kontroli czeków]), aby można było sprawę dokończyć.
- substytucja atrybutów; w przypadku rozpoczęcia sprawy przez kontrolera powinien on znaleźć inną osobę o uprawnieniach kontrolera albo też w momencie obecności urzędnika rozpocząć rozpatrywanie sprawy od nowa.

Zapis dynamicznego sprawdzania rozdziału obowiązków:

Czynność	Rola
Przygotowanie:	urzędnik;
Potwierdzenie	kontroler;
Wydanie	urzędnik;

Wykonanie konkretnej sprawy (obsługi czeku) przebiegać będzie następująco:

Krok 1

Przygotowanie:	Abacki;
Potwierdzenie	kontroler;
Wydanie	urzędnik;

Krok 2

Przygotowanie:	Abacki;
Potwierdzenie	Babacki;
Wydanie	urzędnik;

Krok 3

Przygotowanie:	Abacki;
Potwierdzenie	Babacki;
Wydanie	Cabacki;

Na zakończenie należy podkreślić istotną rolę jaką odgrywa zasada separacji obowiązków we współczesnych systemach workflow. Spośród opisanych zasad zasada statycznej separacji obowiązków oraz zasada prostej dynamicznej separacji obowiązków wydają się zbyt ograniczające współczesne systemy workflow. Zasady obiektowo-zorientowanej separacji obowiązków oraz operacyjnej separacji



obowiązków próbują skompensować niedostatki poprzednich poprzez wprowadzenie ograniczeń pozwalających lepiej zarządzającym przydzielać ról. Najbardziej rozwiniętą jest zasada historycznej separacji obowiązków, która daje dobry mechanizm bezpiecznej kontroli dostępu będąc bardziej elastyczna od poprzednich. Istotnym wymogiem stosowania tej zasady jest przechowywanie historii przebiegu dla każdego procesu biznesowego.

## Literatura

1. Bertino, E., de Capitani di Vimercati, S., Ferrari E., Samarati, P.: Exception-Based Information Flow Control. in: Object-Oriented Systems. Transactions on Information and System Security, 1 (1). 1998.
2. Lee H., Noh B.: An integrity enforcement application design and operation framework in Role-based access control systems: A Session-oriented approach. in: Proceedings of the 1999 International Workshops on Parallel Processing. 1999.
3. Ferraiolo, D., Barkley, J., Kuhn, D.: A Role-Based Access Control Model and Reference Implementation Within a Corporate Internet. ACM Transactions on Information and System Security, 2 (1). 1999.
4. Ferraiolo D., Cugini J., Kuhn D.: Role-Based Access Control (RBAC): Features and Motivations. in: Proceedings of 1995 Computer Security Applications Conference. 1995.
5. International Standards Organization Information processing systems - Open System Interconnection - Basic Reference Model - Part 2: Security Architecture (7498-2). ISO/IEC. 1989.
6. Sandhu R.: Separation of duties in computerized information systems. Proceedings of the IFIP WG 11.3 Workshop on Database Security, Halifax UK, September 18 - 21, 1990.
7. Sandhu R., Coyne E., Feinstein H., Youman C.: Role-based access control models. Computer. February 1996.
8. Sandhu, R. Samarati, P.: Access Control: Principles and Practice. IEEE Communications Magazine, 32 (9). 1994.
9. Simon R. Zurko M.: Separation of duty in role-based environments. in: Proceedings of IEEE Computer Security Foundations Workshop. 1997.

Stanisław Kędziński

AE Katowice

1-go Maja 50

e\_mail: [kedziers@ae.katowice.pl](mailto:kedziers@ae.katowice.pl)

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

... ..  
... ..  
... ..  
... ..  
... ..

# MECHANIZM PERSPEKTYW MATERIALIZOWANYCH W EKSPLOKACJI DANYCH

Mikołaj MORZY, Marek WOJCIECHOWSKI

**Streszczenie:** Eksploracja danych to proces interaktywny i iteracyjny. Użytkownik definiuje zbiór interesujących go wzorców określając eksplorowany zbiór danych i wybierając konkretne wartości parametrów eksploracji. Jest bardzo prawdopodobne, że w celu uzyskania satysfakcjonujących go wyników użytkownik wielokrotnie dokona eksploracji, za każdym razem nieznacznie zmieniając eksplorowany zbiór danych lub modyfikując parametry algorytmu. Aktualnie dostępne algorytmy eksploracji danych charakteryzują się długim czasem przetwarzania, wprost proporcjonalnym do rozmiaru analizowanych danych. Ponieważ eksploracja odbywa się najczęściej w środowisku magazynu danych, długie czasy przetwarzania są nie do przyjęcia z punktu widzenia interaktywnej eksploracji. Z drugiej strony wyniki kolejnych, następujących po sobie zapytań użytkownika są bardzo zbliżone. Jednym z rozwiązań problemu długich czasów przetwarzania zapytań eksploracyjnych jest wykorzystanie zmaterializowanych wyników wcześniejszych zapytań. W tym artykule przedstawiamy koncepcję materializowanych perspektyw eksploracyjnych i sposoby wykorzystania takich perspektyw w przetwarzaniu zapytań eksploracyjnych. Pokazujemy, w jaki sposób mechanizm ten może wydatnie przyspieszyć proces odkrywania reguł asocjacyjnych lub wzorców sekwencji. Wskazujemy też dalsze kierunki badań w tym zakresie.

## Wstęp

Eksploracja danych, zwana także odkrywaniem wiedzy w bazach danych (ang. *data mining, knowledge discovery in databases*) to proces odkrywania nowych, nieznanych, pożytecznych i zrozumiałych wzorców w dużych wolumenach danych [12]. W ostatnich latach obserwujemy wyraźne odchodzenie od dedykowanych i wyspecjalizowanych systemów eksploracyjnych i dążenie do integracji tych systemów z istniejącymi systemami zarządzania bazami danych. Integracja ta przebiega przede wszystkim w ramach magazynów danych (ang. *data warehouses*), które stanowią doskonałe źródło danych dla różnych technik eksploracyjnych. Z punktu widzenia użytkownika wykonanie algorytmu i odkrycie zbioru wzorców to rodzaj odpowiedzi na zaawansowane zapytanie do bazy danych. Użytkownik określa zbiór eksplorowanych danych (np. za pomocą standardowego zapytania wyrażonego w języku SQL) oraz wyznacza wartości współczynników sterujących danym algorytmem odkrywania wzorców. W odpowiedzi system wykonuje odpowiedni algorytm i prezentuje użytkownikowi uzyskany zbiór wzorców.



Użytkownik z reguły nie zna dokładnego celu eksploracji, lecz dochodzi do interesujących i satysfakcjonujących go wyników w wielu następujących po sobie krokach. W każdym kroku użytkownik weryfikuje zbiór uzyskanych wzorców i stosownie do swych potrzeb i oczekiwań zmienia zbiór eksplorowanych danych (modyfikując odpowiednie polecenie języka SQL), lub dostraja algorytm odkrywania wzorców przez zmiany wartości parametrów. Praktyka wykazuje, że w typowym procesie odkrywania wzorców użytkownik wykonuje wiele razy ten sam algorytm z nieznacznie zmienionymi parametrami. Z drugiej strony często zdarza się, że użytkownik wydaje dane zapytanie eksploracyjne okresowo, np. raz w tygodniu, w celu znalezienia najbardziej aktualnych wzorców. W takim wypadku system powinien móc przechować wyniki poprzedniej eksploracji i spróbować udzielić odpowiedzi w sposób przyrostowy, bazując na wcześniejszych wynikach i uwzględniając zmiany, jakie zaszły w bazie danych od czasu ostatniej eksploracji. W przypadku magazynu danych wolumen zmian stanowi zazwyczaj znikomą część oryginalnej bazy danych.

Podstawowym problemem, jaki napotyka się podczas eksploracji danych, jest czas przetwarzania typowego zapytania eksploracyjnego. Algorytmy odkrywania reguł potrzebują minut i godzin aby odpowiedzieć na stosunkowo proste zapytanie. Często też rozmiar odpowiedzi przekracza rozmiar eksplorowanej bazy danych. Taka charakterystyka procesu odkrywania wiedzy czyni go zupełnie niezdatnym do zastosowań interaktywnych i iteracyjnych.

Jednym z rozwiązań powyższego problemu jest zastosowanie perspektyw materializowanych. Materializacja wyników zapytań eksploracyjnych może się odbywać automatycznie, lub na żądanie użytkowników. System eksploracyjny powinien umożliwiać wykorzystanie tych wyników i włączenie ich do algorytmu odkrywania wzorców. Mechanizm perspektyw materializowanych został dokładnie zbadany i z powodzeniem wykorzystany w tradycyjnych systemach relacyjnych baz danych. Proponujemy, aby podobne rozwiązanie zastosować w przypadku systemów odkrywania wiedzy.

W artykule pokazujemy, w jaki sposób można wykorzystać wyniki wcześniejszych zapytań eksploracyjnych do skrócenia czasu przetwarzania w przypadku odkrywania zbiorów częstych, reguł asocjacyjnych i wzorców sekwencyjnych. Eksperymenty dowodzą, że wykorzystanie wcześniejszych wyników wielokrotnie skraca czas wykonania zapytania. Jednakże, w przypadku zapytań eksploracyjnych określenie perspektyw materializowanych, które mogą być wykorzystane do odpowiedzi na dane zapytanie, nie jest proste. Perspektywa eksploracyjna może się różnić od danego zapytania nie tylko wartościami współczynników wykonania algorytmu, ale również schematem eksplorowanej bazy danych. Poniżej pokazujemy, w jakich przypadkach można wykorzystać perspektywę materializowaną do odpowiedzi na zapytanie eksploracyjne i jakie dodatkowe czynności są konieczne, aby zwrócona odpowiedź była poprawna. Przykłady przedstawione w artykule zostały wyrażone za pomocą języka MySQL, deklaratywnego języka do eksploracji danych opartego na języku SQL i rozwijanego od kilku lat w Instytucie Informatyki Politechniki Poznańskiej [14].

## 1. Podstawowe definicje

### 1.1. Perspektywy zwykle i materializowane

Perspektywa to wywiedziona tabela, zdefiniowana w oparciu o tabele bazowe. Perspektywa definiuje funkcję ze zbioru tabel bazowych do tabeli wywiedzionej. Funkcja ta jest zazwyczaj obliczana przy każdym odwołaniu do perspektywy. Perspektywa może zostać zmaterializowana poprzez składowanie krotek perspektywy w bazie danych. Na materializowanej perspektywie można zakładać indeksy, przez co dostęp do tak zapisanych danych może być dużo szybszy niż ponowne wyliczenie perspektywy. Perspektywa materializowana w pewnym sensie przypomina pamięć podręczną - jest kopią danych którą można szybko odczytać. Materializacja perspektywy eliminuje potrzebę ponownego obliczania i rozwijania perspektywy przy każdym dostępie do niej.

Zawartość materializowanej perspektywy staje się nieaktualna w momencie modyfikacji tabel bazowych, z których wywiedziono perspektywę. Proces modyfikowania perspektywy materializowanej w odpowiedzi na zmiany zachodzące w tabelach bazowych nazywamy pielęgnacją perspektywy. Często zmiany w tabelach bazowych powodują zmianę tylko w części perspektywy. Pielęgnacja perspektywy przez wyliczanie całej jej zawartości byłaby w takiej sytuacji marnotrawstwem. Znacznie prościej i szybciej jest wyznaczyć zmianę zachodzącą w perspektywie materializowanej tylko na podstawie zmian jakie zaszły w tabelach bazowych. Taką pielęgnację nazywamy pielęgnacją przyrostową (inkrementalną). Należy jednak pamiętać, że w przypadku wielu typów perspektyw materializowanych pielęgnacja przyrostowa jest niemożliwa.

### 1.2. Zbiory częste

Niech  $L = \{l_1, l_2, \dots, l_n\}$  będzie zbiorem literałów zwanych elementami. Niech  $D$  będzie kolekcją transakcji, gdzie każda transakcja jest dowolnej długości i  $\forall T \in D \ T \subseteq L$ . Mówimy, że transakcja  $T$  wspiera element  $x$  jeśli  $x \in T$ . Mówimy, że transakcja  $T$  wspiera zbiór  $X$ , jeśli  $T$  wspiera każdy element  $x \in X$ . *Wsparciem* (ang. support) zbioru  $X$  nazywamy stosunek liczby transakcji wspierających  $X$  do liczby wszystkich transakcji.

$$\text{support}(X, D) = \frac{|\{T \in D : T \text{ wspiera } X\}|}{|D|}$$

Problem odkrywania zbiorów częstych polega na znalezieniu w danej bazie danych  $D$  wszystkich zbiorów, których wsparcie jest wyższe od zdefiniowanej przez użytkownika wartości, zwanej minimalnym wsparciem (*minsup*). Zbiór, którego wsparcie jest wyższe niż *minsup* nazywamy **zbiorem częstym** (ang. *frequent itemset*).

### 1.3. Reguły asocjacyjne

Reguła asocjacyjna to implikacja postaci  $X \rightarrow Y$ , gdzie  $X \subset L$ ,  $Y \subset L$  i  $X \cap Y = \emptyset$ . Zbiór  $X$  nazywamy głową reguły a zbiór  $Y$  ciałem reguły. Z każdą regułą asocjacyjną związane są dwie miary wyznaczające statystyczne znaczenie i siłę reguły. **Wsparciem** (ang. *support*) reguły  $X \rightarrow Y$  w bazie danych  $D$  nazywamy stosunek liczby transakcji wspierających regułę do liczby wszystkich transakcji. Innymi słowy reguła  $X \rightarrow Y$  ma w bazie danych  $D$  wsparcie  $s$ , jeśli  $s\%$  transakcji w bazie danych wspiera  $X \cup Y$ .

$$\text{support}(X \rightarrow Y, D) = \frac{|\{T \in D : T \text{ wspiera } X \cup Y\}|}{|D|}$$

**Ufnością** (ang. *confidence*) reguły  $X \rightarrow Y$  w bazie danych  $D$  nazywamy stosunek liczby transakcji wspierających regułę do liczby transakcji wspierających głowę reguły. Innymi słowy reguła  $X \rightarrow Y$  ma w bazie danych  $D$  ufność  $c$ , jeśli  $c\%$  transakcji wspierających  $X$  wspiera również  $Y$ .

$$\text{confidence}(X \rightarrow Y, D) = \frac{|\{T \in D : T \text{ wspiera } X \cup Y\}|}{|\{T \in D : T \text{ wspiera } X\}|}$$

Problem odkrywania reguł asocjacyjnych polega na znalezieniu w danej bazie danych  $D$  wszystkich reguł asocjacyjnych, których wsparcie i ufność są wyższe od zdefiniowanych przez użytkownika wartości minimalnego wsparcia i minimalnej ufności (*minsup* i *minconf*).

### 1.4. Wzorce sekwencyjne

Niech  $L = \{l_1, l_2, \dots, l_n\}$  będzie zbiorem literałów zwanych elementami. Sekwencją nazywamy uporządkowaną listę zbiorów elementów. Sekwencję oznaczamy przez  $\langle X_1, X_2, \dots, X_n \rangle$ , gdzie  $X_i$  jest zbiorem elementów,  $X_i \subset L$ . Zbiory  $X_i$  nazywamy wyrazami sekwencji. Rozmiarem sekwencji nazywamy liczbę występujących w niej elementów. Długością sekwencji nazywamy liczbę występujących w niej wyrazów. Z każdym wyrazem sekwencji związany jest znacznik czasowy. Pomijając ograniczenia czasowe mówimy, że sekwencja  $\langle X_1, X_2, \dots, X_n \rangle$  zawiera się w sekwencji  $\langle Y_1, Y_2, \dots, Y_m \rangle$ , jeśli istnieją liczby całkowite  $i_1 < i_2 < \dots < i_n$  takie, że  $X_1 \subset Y_{i_1}, X_2 \subset Y_{i_2}, \dots, X_n \subset Y_{i_n}$ . Sekwencję  $\langle Y_{i_1}, Y_{i_2}, \dots, Y_{i_n} \rangle$  nazywamy wystąpieniem sekwencji  $X$  w sekwencji  $Y$ . Odkrywając wzorce sekwencyjne posługujemy się następującymi ograniczeniami czasowymi: minimalnym i maksymalnym przedziałem pomiędzy kolejnymi elementami wystąpienia sekwencji (odpowiednio *min-gap* i *max-gap*), oraz rozmiarem okna czasowego, które pozwala na łączenie tych samych pozycji sekwencji w jedną pozycję (o ile znaczniki czasowe tych pozycji mieszczą się w określonym oknie). Wsparciem sekwencji  $\langle X_1, X_2, \dots, X_n \rangle$  w bazie danych  $D$  nazywamy stosunek liczby sekwencji zawierających  $X$  do liczby wszystkich sekwencji.

Problem odkrywania wzorców sekwencyjnych polega na znalezieniu w danej bazie danych  $D$  wszystkich sekwencji, których wsparcie jest wyższe od



zdefiniowanej przez użytkownika wartości minimalnego wsparcia (*minsup*). Sekwencję, której wsparcie jest wyższe niż *minsup*, nazywamy **wzorcem sekwencyjnym**.

## 2. Aktualny stan badań

Prace nad perspektywami materializowanymi rozpoczęły się w latach 80-tych. Początkowo były one narzędziem do przyspieszenia wykonywania zapytań i udostępnienia starszych kopii danych. Opracowano wiele algorytmów pielęgnacji perspektyw materializowanych. Dalsze badania dotyczyły między innymi tworzenia modeli szacowania kosztów pielęgnacji perspektyw materializowanych oraz określania wpływu obecności perspektyw na efektywność przetwarzania zapytań. Kolejne prace dotyczyły zastosowania perspektyw do narzucania ograniczeń integralnościowych. W pracy [8] można znaleźć podsumowanie i klasyfikację różnych technik pielęgnacji perspektyw. Obszerne przedstawienie tematu znajduje się w [9].

Problem odkrywania reguł asocjacyjnych po raz pierwszy sformułowano w [1]. W [2] wprowadzono pojęcie zbioru częstego i zaproponowano algorytm Apriori, który stał się podstawą bardzo wielu różnych metod odkrywania wzorców. Działanie algorytmu opiera się na następującej obserwacji: zbiór elementów może być częsty wtedy i tylko wtedy, gdy wszystkie jego podzbiory są częste. Apriori generuje zbiory potencjalnie częste (zwane zbiorami kandydującymi) tylko na podstawie dotychczas znalezionych zbiorów częstych. W pierwszym kroku znajdowane są wszystkie zbiory częste o rozmiarze 1, następnie wszystkie zbiory częste o rozmiarze 2. We wszystkich kolejnych krokach tworzone są zbiory  $n$ -elementowe na podstawie  $(n-1)$ -elementowych zbiorów częstych. Wsparcie zbiorów kandydujących o danym rozmiarze określa się podczas pełnego odczytu bazy danych. Podstawową wadą algorytmu Apriori jest właśnie fakt, że potrzebuje on  $(k+1)$  pełnych odczytów bazy danych aby znaleźć wszystkie  $k$ -elementowe zbiory częste.

W [6] zaproponowano algorytm FUP, który odkrywał zbiory częste w oparciu o wcześniej znalezione wyniki. Algorytm ten skracał wydatnie czas działania algorytmu Apriori poprzez przetwarzanie tylko zmodyfikowanej części bazy danych. Kolejną propozycją był algorytm zaprezentowany w [17], który minimalizował czas odkrywania zbiorów częstych w zarówno zwiększonej, jak i zmniejszonej bazie danych. Algorytm ten odkrywał zbiory częste korzystając z pojęcia negatywnej granicy, wprowadzonego w [18].

Idea odkrywania wzorców sekwencyjnych została po raz pierwszy przedstawiona w [3] i [4]. Zaproponowano wówczas algorytm GSP odkrywający szeroką klasę wzorców sekwencyjnych (tzw. uogólnione wzorce sekwencyjne) i wykorzystujący ograniczenia czasowe. W [16] zaproponowano materializację wzorców o obniżonych kryteriach wsparcia i wykorzystanie kolekcji zmaterializowanych wzorców do znalezienia odpowiedzi na dane zapytanie eksploracyjne. Większość prac dotyczących wzorców sekwencyjnych

koncentrowała się jednak na ulepszaniu algorytmu odkrywania wzorców [10,11]. W pracy [15] po raz pierwszy rozważano koncepcję interaktywnego i iteracyjnego odkrywania zbiorów częstych. Autorzy zaproponowali stworzenie podręcznej pamięci wiedzy (ang. *knowledge cache*), która przechowywałaby ostatnio odkryte zbiory częste wraz z ich wsparciem. Taka pamięć podręczna może być współdzielona przez wielu użytkowników i wiele aplikacji, dzięki czemu użytkownicy mogą nawzajem wykorzystywać wyniki swoich zapytań eksploracyjnych. Poza opracowaniem samej koncepcji autorzy zaproponowali kilka różnych schematów zarządzania zawartością pamięci podręcznej.

Idea wcześniejszego obliczania wsparcia zbiorów częstych w partycjach bazy danych pojawiła się po raz pierwszy w [19]. Zaproponowana metoda wykorzystywała fakt, że dany zbiór może być częsty wtedy i tylko wtedy, gdy jest częsty w którejkolwiek z partycji. Algorytm dokonywał wcześniejszego odkrywania zbiorów częstych w małych, mieszczących się w całości w dostępnej pamięci operacyjnej partycjach bazy danych i wykorzystywał te zbiory do znajdowania zbiorów, które były częste w całej bazie danych.

W [12] przedstawiono ideę Systemu Zarządzania Wiedzą i Danymi (ang. *Knowledge Data Management System*), który powinien być następcą współczesnych systemów zarządzania bazami danych. Autorzy po raz pierwszy zdefiniowali pojęcie zapytań eksploracyjnych oraz podkreślili potrzebę ścisłej integracji systemów odkrywania wiedzy z istniejącą infrastrukturą informatyczną, przede wszystkim z bazami danych i magazynami danych.

## 2.1. Streszczenie artykułu

Artykuł zorganizowany jest następująco. W rozdziale 4 przedstawiono koncepcję zapytań eksploracyjnych i zaprezentowano przykłady takich zapytań. Rozważono związki zachodzące między wynikami różnych zapytań i zdefiniowano pojęcie perspektywy eksploracyjnej. Rozdział 5 dotyczy optymalizacji zapytań eksploracyjnych z wykorzystaniem perspektyw materializowanych. W rozdziale 6 przedstawiono nierozwiązane problemy i naszkicowano zarys dalszych kierunków badań.

## 3. Zapytania eksploracyjne i relacje między zapytaniami

### 3.1. Zapytania eksploracyjne

W [14] przedstawiono deklaratywny język eksploracyjny MineSQL. Służy on do wyrażania problemów odkrywania wiedzy za pomocą **zapytań eksploracyjnych** (ang. *data mining queries*). Język ten oddziela aplikację użytkownika od używanego algorytmu odkrywania wiedzy. Składnia MineSQL przypomina składnię języka SQL i pozwala na ścisłą integrację zapytań eksploracyjnych z tradycyjnymi zapytaniami do bazy danych. Język MineSQL pozwala aktualnie na wyrażanie poleceń służących do odkrywania zarówno zbiorów częstych, jak i reguł asocjacyjnych i wzorców sekwencyjnych. MineSQL

definiuje zbiór dodatkowych typów danych (SET, ITEMSET, RULE) oraz zbiór operatorów i funkcji operujących na tych typach danych (np. CONTAINS, BODY(x), HEAD(x)).

Poniżej przedstawiono przykładowe zapytanie eksploracyjne, odkrywające w podanym zbiorze danych zbiory częste o wsparciu powyżej 20% i zawierające element 'mleko'.

```
MINE ITEMSET, SUPPORT(ITEMSET)
FOR ITEMS FROM (
    SELECT SET(PURCHASED_ITEM) AS ITEMS
    FROM PURCHASES
    WHERE DATE_OF_PURCHASE > '01.07.2001'
    AND DATE_OF_PURCHASE < '31.12.2001'
    GROUP BY TRANSACTION_ID )
WHERE SUPPORT(ITEMSET) > 0.2
AND ITEMSET CONTAINS TO_SET('mleko');
```

Analogicznie można w języku MineSQL wyrazić polecenie odkrywania reguł asocjacyjnych, których wsparcie jest większe niż 10%, ufność większa niż 30% i których głowa zawiera element 'masło'.

```
MINE RULE r, BODY(r), HEAD(r)
FOR ITEMS FROM (
    SELECT SET(PURCHASED_ITEM) AS ITEMS
    FROM PURCHASES
    GROUP BY TRANSACTION_ID )
WHERE SUPPORT(r) > 0.1
AND CONFIDENCE(r) > 0.3
AND HEAD(r) CONTAINS TO_SET('masło');
```

### 3.2. Relacje zachodzące między wynikami zapytań eksploracyjnych

W [5] określono trzy rodzaje relacji, jakie zachodzą między dwoma zapytaniami eksploracyjnymi  $Q_1$  i  $Q_2$  odkrywającymi wzorce w tej samej bazie danych. Są to równoważność, zawieranie się oraz dominacja.

- Dwa zapytania eksploracyjne są **równoważne**, jeśli dla każdego zbioru danych zwracają ten sam zbiór odkrytych wzorców i dla każdej pary wzorców wartości współczynników statystycznych (np. wsparcia i ufności) są identyczne.
- Zapytanie eksploracyjne  $Q_2$  **zawiera** zapytanie  $Q_1$  jeżeli dla każdego zbioru danych każdy wzorec odkryty przez zapytanie  $Q_1$  jest też odkryty przez  $Q_2$  i wartości współczynników statystycznych są identyczne w obu przypadkach.
- Zapytanie eksploracyjne  $Q_2$  **dominuje** zapytanie  $Q_1$  jeżeli dla każdego zbioru danych każdy wzorec odkryty przez zapytanie  $Q_1$  jest też odkryty



przez  $Q_2$  i wartości współczynników statystycznych wyznaczone przez  $Q_1$  są nie mniejsze niż wartości współczynników wyznaczone przez  $Q_1$ .

Równoważność zapytań eksploracyjnych jest więc szczególnym przypadkiem relacji zawierania, zaś relacja zawierania jest szczególnym przypadkiem relacji dominacji.

Relacje te zachodzą między wynikami zapytań i mogą być wykorzystane do zidentyfikowania sytuacji, w których można efektywnie udzielić odpowiedzi na zapytanie eksploracyjne  $Q_1$  wykorzystując wynik innego zapytania  $Q_2$ . Relacje te mają charakter ogólny i można je zastosować do wielu typów wzorców (zbiorów częstych, reguł asocjacyjnych) oraz wielu modeli ograniczeń.

Jeżeli dla danego zapytania eksploracyjnego  $Q_1$  istnieją zmaterializowane wyniki równoważnego mu zapytania  $Q_2$ , wówczas żadne przetwarzanie nie jest konieczne (ponieważ zapytania mają ten sam wynik). Jeśli dostępne są wyniki zapytania  $Q_2$  zawierającego oryginalne zapytanie, konieczny jest jeden pełny odczyt zmaterializowanych wyników i odrzucenie wzorców nie spełniających ograniczeń nałożonych na  $Q_1$ . Jeśli dostępne są wyniki zapytania  $Q_2$  dominującego oryginalne zapytanie, konieczny jest jeden pełny odczyt bazy danych i określenie statystycznych współczynników wzorców zmaterializowanych w  $Q_2$ . Dodatkowo trzeba z  $Q_2$  odfiltrować te wzorce, które nie spełniają ograniczeń nałożonych na  $Q_1$ .

### 3.3. Perspektywy eksploracyjne

Tradycyjne użycie perspektyw ma na celu przede wszystkim ukrycie przed użytkownikami skomplikowanych konstrukcji zapytań i uproszczenie dostępu do często odczytywanych danych. Dodatkowo perspektywa zapewnia niezależnienie aplikacji od bieżącej struktury bazy danych. Wszelkie zmiany zachodzące w bazie danych muszą zostać uwzględnione tylko w definicji perspektywy. Każdy odczyt danych z perspektywy powoduje ponowne wykonanie zapytania definiującego tę perspektywę.

Ponieważ eksploracja danych jest czynnością iteracyjną i powtarzalną, zaś zapytania eksploracyjne mogą być skomplikowane, w [14] wprowadzono pojęcie perspektyw eksploracyjnych. Poniżej przedstawiono polecenie tworzące perspektywę eksploracyjną V\_ASSOC\_RULES.

```
CREATE VIEW V_ASSOC_RULES AS
MINE RULE, BODY(RULE), SUPPORT(RULE)
FOR ITEMS FROM (
  SELECT SET(PURCHASED_ITEM) AS ITEMS
  FROM PURCHASES
  WHERE TRANS_DATE BETWEEN '01.01.2002' AND '31.01.2002'
  GROUP BY TRANSACTION_ID
  HAVING COUNT(*) >= 3)
WHERE SUPPORT(RULE) > 0.2
AND HEAD(RULE) CONTAINS TO_SET('chleb');
```

W powyższej definicji można wyróżnić dwie klasy ograniczeń: ograniczenia

bazodanowe (klauzula WHERE w zapytaniu SELECT) oraz ograniczenia eksploracyjne (klauzula WHERE w zapytaniu MINE). Ograniczenia bazodanowe definiują zbiór danych, w którym następuje odkrywanie wzorców. Ograniczenia eksploracyjne definiują warunki, jakie muszą spełnić odkrywane wzorce.

Dzięki zastosowaniu perspektyw eksploracyjnych aplikacje nie muszą być ściśle powiązane z algorytmem odkrywania wzorców. Zmiany parametrów algorytmu lub zmiany zbioru, w którym odbywa się eksploracja, mogą być wprowadzane do definicji perspektywy, separując tym samym aplikację od szczegółów implementacji algorytmu. Podobnie jak w przypadku tradycyjnych perspektyw, każde odwołanie do perspektywy eksploracyjnej powoduje wykonanie odpowiedniego zapytania, czyli wykonanie algorytmu odkrywania wzorców.

Ponieważ algorytmy odkrywania wzorców są bardzo czasochłonne, wykonanie zapytania do perspektywy eksploracyjnej mogłoby trwać zbyt długo z punktu widzenia interaktywnego procesu odkrywania wiedzy. Rozwiązaniem tego problemu jest materializacja wyników uzyskanych we wcześniejszych zapytaniach eksploracyjnych. Pomysł materializowanych perspektyw eksploracyjnych został po raz pierwszy sformułowany w [14]. Materializowana perspektywa eksploracyjna to obiekt w bazie danych, który przechowuje wzorce (zbiory częste, reguły asocjacyjne, wzorce sekwencyjne) odkryte podczas wykonywania zapytania eksploracyjnego. Wzorce przechowywane w takiej perspektywie mają związany z sobą znacznik czasowy, określający moment ich odkrycia (i ważności). Z każdą perspektywą materializowaną może też być związany przedział czasowy, po którym następuje automatyczne odświeżenie zawartości perspektywy. Poniżej przedstawiono polecenie tworzące perspektywę materializowaną MV\_ASSOC\_RULES.

```
CREATE MATERIALIZED VIEW MV_ASSOC_RULES  
REFRESH 7 AS  
MINE RULE, SUPPORT(RULE), CONFIDENCE(RULE)  
FOR ITEMS FROM (  
    SELECT SET(PURCHASED_ITEM) AS ITEMS  
    FROM PURCHASES  
    WHERE ITEM_GROUP = 'nabiał'  
    GROUP BY TRANSACTION_ID )  
WHERE SUPPORT(RULE) > 0.3  
AND CONFIDENCE(RULE) > 0.5;
```

Odświeżanie perspektywy materializowanej może odbywać się automatycznie lub na żądanie użytkownika. W większości przypadków perspektywy takie można odświeżać za pomocą jednego z efektywnych algorytmów odświeżania przyrostowego [6,7,17], zamiast wykonywać kosztowny algorytm od podstaw. Na korzyść perspektyw materializowanych przemawia dodatkowo fakt, że eksploracja odbywa się najczęściej w środowisku magazynu danych, w którym zmiany w relacjach bazowych nie mają ciągłego charakteru, lecz następują wszystkie w jednym momencie, podczas ładowania lub odświeżania zawartości magazynu. Tak

więc reguły odkryte i przechowywane w perspektywie materializowanej przez długi czas pozostają poprawne, zaś ich weryfikacja powinna się odbyć podczas odświeżania całego magazynu.

#### 4. Optymalizacja zapytań eksploracyjnych za pomocą perspektyw materializowanych

##### 4.1. Zbiory częste i reguły asocjacyjne

W wielu przypadkach zawartość perspektywy materializowanej może posłużyć do odpowiedzi na zapytanie eksploracyjne, które jest podobne do zapytania definiującego perspektywę. Jeśli np. zapytanie definiujące perspektywę  $Q_v$  dominuje lub zawiera dane zapytanie eksploracyjne  $Q$ , to w celu udzielenia odpowiedzi na zapytanie  $Q$  wystarczy odczytać zawartość perspektywy i odfiltrować te wzorce, które nie spełniają warunków sformułowanych w  $Q$ . W celu wykorzystania perspektyw materializowanych do optymalizacji zapytań eksploracyjnych trzeba jednak najpierw określić warunki, jakie muszą być spełnione, aby można było udzielić poprawnej odpowiedzi na zapytanie  $Q$  przy użyciu zawartości perspektywy. W tym celu należy najpierw zdefiniować relacje, jakie mogą wystąpić między dwoma zapytaniami eksploracyjnymi.

- Zapytanie  $Q$  **rozszerza** ograniczenia bazodanowe zapytania  $Q_v$ , jeżeli:
  - dodaje do ograniczeń bazodanowych zapytania  $Q_v$  dodatkowe klauzule WHERE lub HAVING
  - dodaje koniunkcję nowego warunku do warunków bazodanowych w klauzulach WHERE lub HAVING
  - usuwa warunek z alternatywy warunków bazodanowych w klauzulach WHERE lub HAVING
- Zapytanie  $Q$  **redukuje** ograniczenia bazodanowe zapytania  $Q_v$ , jeżeli:
  - usuwa z ograniczeń bazodanowych zapytania  $Q_v$  klauzule WHERE lub HAVING
  - usuwa warunek z koniunkcji warunków bazodanowych w klauzulach WHERE lub HAVING
  - dodaje alternatywę nowego warunku do warunków bazodanowych w klauzulach WHERE lub HAVING
- Zapytanie  $Q$  **rozszerza** ograniczenia eksploracyjne zapytania  $Q_v$ , jeżeli:
  - dodaje do ograniczeń eksploracyjnych zapytania  $Q_v$  dodatkowe klauzule WHERE lub HAVING
  - dodaje koniunkcję nowego warunku do warunków eksploracyjnych w klauzulach WHERE lub HAVING
  - usuwa warunek z alternatywy warunków eksploracyjnych w klauzulach WHERE lub HAVING
  - zastępuje ograniczenie eksploracyjne występujące w  $Q_v$  ograniczeniem bardziej restryktywnym (np. wyższy próg minimalnego wsparcia)



- Zapytanie Q **redukuje** ograniczenia eksploracyjne zapytania  $Q_v$ , jeżeli:
  - redukuje ograniczenia eksploracyjne zapytania  $Q_v$  o klauzule WHERE lub HAVING
  - usuwa warunek z koniunkcji warunków eksploracyjnych w klauzulach WHERE lub HAVING
  - dodaje alternatywę nowego warunku do warunków eksploracyjnych w klauzulach WHERE lub HAVING
  - zastępuje ograniczenie eksploracyjne występujące w  $Q_v$  ograniczeniem mniej restryktywnym (np. niższy próg minimalnego wsparcia)

Rozszerzenie ograniczeń bazodanowych oznacza zawężenie zbioru danych, w którym ma przebiegać eksploracja. Redukcja ograniczeń bazodanowych oznacza rozszerzenie eksplorowanego zbioru. Rozszerzenie ograniczeń eksploracyjnych oznacza zawężenie zbioru wzorców zaś redukcja ograniczeń eksploracyjnych oznacza rozszerzenie wynikowego zbioru wzorców.

W zależności od okoliczności można wykorzystać jeden z czterech sposobów eksploracji. **Pełna eksploracja** (ang. *full mining*) polega na wykonaniu całego algorytmu odkrywania wzorców, bez wykorzystania zawartości perspektywy. Ta sytuacja następuje wówczas, gdy dane zapytanie Q rozszerza ograniczenia bazodanowe zapytania definiującego perspektywę  $Q_v$ . **Eksploracja przyrostowa** (ang. *incremental mining*) polega na wykonaniu któregoś z algorytmów przyrostowego odkrywania wzorców (np. [6]) w rozszerzonym zbiorze danych. Tę metodę można zastosować w przypadku redukcji ograniczeń bazowych względem zapytania  $Q_v$ . **Eksploracja uzupełniająca** (ang. *complementary mining*) polega na odkrywaniu wzorców na podstawie wcześniej znalezionych wzorców. Eksploracja uzupełniająca znajduje zastosowanie w przypadku, gdy zapytanie Q redukuje ograniczenia eksploracyjne zawarte w perspektywie (wszystkie wzorce zawarte w perspektywie znajdują się również w odpowiedzi na zapytanie Q). Wreszcie **eksploracja weryfikująca** (ang. *verifying mining*) polega na odfiltrowaniu tych wzorców przechowywanych w perspektywie, które nie spełniają rozszerzonych ograniczeń eksploracyjnych.

Poniżej zamieszczono przykład ilustrujący zastosowanie zawartości materializowanej perspektywy eksploracyjnej do udzielenia odpowiedzi na zapytanie eksploracyjne.

Dana jest następująca definicja perspektywy  $Q_v$ :

```

MINE ITEMSET, SUPPORT (ITEMSET)
FOR ITEMS FROM (
    SELECT SET (PURCHASED_ITEM) AS ITEMS
    FROM PURCHASES
    GROUP BY TRANSACTION_ID
HAVING COUNT (*) > 5 )
WHERE SUPPORT (ITEMSET) > 0.3;

```

oraz następujące zapytanie Q:

```
MINE ITEMSET
FOR ITEMS FROM (
    SELECT SET(PURCHASED_ITEM) AS ITEMS
    FROM PURCHASES
    GROUP BY TRANSACTION_ID )
WHERE SUPPORT(ITEMSET) > 0.5
AND ITEMSET CONTAINS TO_SET('mleko','masło');
```

Zapytanie Q rozszerza ograniczenia eksploracyjne  $Q_v$  (większa wartość współczynnika minimalnego wsparcia oraz dodatkowa klauzula z ograniczeniami eksploracyjnymi) i jednocześnie redukuje ograniczenia bazodanowe  $Q_v$  (usunięta klauzula HAVING z ograniczeń bazodanowych  $Q_v$ , zatem wykonywana jest eksploracja weryfikująca (odrzućenie zbiorów częstych o wsparciu niższym niż 0.5 i nie zawierających zbioru  $\{mleko, masło\}$ ), a następnie eksploracja przyrostowa, wyszukująca zbiory częste w transakcjach składających się z mniej niż 5 elementów.

#### 4.2. Wzorce sekwencyjne

Podobnie jak w przypadku odkrywania zbiorów częstych lub reguł asocjacyjnych, materializacja wyników wcześniejszych zapytań może wydatnie przyspieszyć odkrywanie wzorców sekwencyjnych. W celu wykorzystania zawartości perspektywy materializowanej do odpowiedzi na zapytanie o wzorce sekwencyjne Q należy najpierw określić rodzaj relacji pomiędzy oboma zapytaniami (zapytaniem eksploracyjnym i zapytaniem definiującym perspektywę). Relacja ta jest uzależniona od związków między poszczególnymi klasami ograniczeń zawartymi w obu zapytaniach.

W podstawowym ujęciu problem odkrywania wzorców sekwencyjnych jest zdefiniowany za pomocą trzech klas ograniczeń:

- o ograniczenia bazodanowe: wykorzystywane do zawężenia eksplorowanej bazy danych do interesującego podzbioru
- o ograniczenia eksploracyjne: parametry algorytmu odkrywania wzorców sekwencyjnych, aktualnie stosuje się tylko współczynniki minimalnego wsparcia poszukiwanych wzorców (ang. *minsup*)
- o ograniczenia czasowe: wykorzystywane do określania rozmiaru okna przetwarzania sekwencji, aktualnie stosuje się współczynniki minimalnej i maksymalnej odległości między elementami oraz szerokości okna (ang. *min-gap*, *max-gap*, *window-width*)

Dla dwóch zapytań eksploracyjnych  $Q_1$  i  $Q_2$  można określić następujące wzajemne relacje w odniesieniu do wyżej wymienionych ograniczeń:

- o zapytanie  $Q_2$  **rozszerza** ograniczenia eksploracyjne zapytania  $Q_1$  jeżeli ograniczenia eksploracyjne  $Q_1$  mogą być uzyskane poprzez dodanie do ograniczeń eksploracyjnych obecnych w  $Q_2$  nowych predykatów

elementarnych, lub przez zastąpienie predykatów  $Q_2$  przez silniejsze (bardziej restrykcyjne) predykaty.

- o zapytanie  $Q_2$  **rozszerza** ograniczenia czasowe zapytania  $Q_1$  jeżeli zaostrza któryś z współczynników *min-gap*, *max-gap* lub *window-width*, jednocześnie nie łągdując innych współczynników.

Powyższe uwagi dotyczą składni zapytań eksploracyjnych. Wpływ różnic składniowych na występowanie jednej z ogólnych relacji między zapytaniami (równoważność, dominacja i zawieranie) został zbadany w [20]. Wykazano tam, że w przypadku zapytań eksploracyjnych dotyczących wzorców sekwencyjnych zachodzą następujące zależności:

- o Niech zapytania  $Q_1$  i  $Q_2$  operują na tym samym zbiorze danych (czyli posiadają te same ograniczenia bazodanowe) i posiadają te same ograniczenia czasowe. Jeżeli  $Q_2$  rozszerza ograniczenia eksploracyjne  $Q_1$ , wówczas  $Q_1$  zawiera  $Q_2$ .
- o Niech zapytania  $Q_1$  i  $Q_2$  operują na tym samym zbiorze danych i posiadają te same ograniczenia eksploracyjne. Jeżeli  $Q_2$  rozszerza ograniczenia czasowe  $Q_1$ , wówczas  $Q_1$  dominuje  $Q_2$ .
- o Niech zapytania  $Q_1$  i  $Q_2$  operują na tym samym zbiorze danych. Jeśli  $Q_2$  rozszerza ograniczenia eksploracyjne i czasowe  $Q_1$ , wówczas  $Q_1$  dominuje  $Q_2$ .

Te zależności stanowią podstawę algorytmów wykorzystujących zmaterializowane wyniki wcześniejszych eksploracji do udzielenia odpowiedzi na dane zapytanie. Poniżej przedstawiono krótki opis możliwości wykorzystania poszczególnych technik eksploracji z wykorzystaniem perspektyw materializowanych. Pełny opis tych algorytmów i analiza kosztów ich wykonania znajduje się w [20]. We wszystkich przykładach  $Q$  oznacza zapytanie eksploracyjne odkrywające wzorce sekwencyjne, zaś  $Q_v$  oznacza zapytanie stanowiące definicję perspektywy materializowanej MV.

Jeżeli  $Q$  i  $Q_v$  operują na tym samym zbiorze danych (ograniczenia bazodanowe są takie same) i posiadają te same ograniczenia czasowe i eksploracyjne, to między zapytaniami zachodzi równoważność. Wyniki obu zapytań są identyczne i żadne dodatkowe przetwarzanie nie jest konieczne, cały wynik znajduje się w perspektywie materializowanej MV.

Jeżeli  $Q$  i  $Q_v$  operują na tym samym zbiorze danych i mają te same ograniczenia czasowe, zaś  $Q$  rozszerza ograniczenia eksploracyjne  $Q_v$ , wówczas można udzielić odpowiedzi przez odrzucenie z perspektywy tych wzorców, które nie spełniają ograniczeń  $Q$  ( $Q_v$  zawiera  $Q$ ). Algorytm dokonuje pojedynczego odczytu całej zawartości perspektywy materializowanej MV i dla każdego wzorca sprawdza, czy wzorzec ten spełnia dodatkowe ograniczenia eksploracyjne nałożone przez  $Q$ .

Jeżeli  $Q$  i  $Q_v$  operują na tym samym zbiorze danych i mają te same ograniczenia eksploracyjne, zaś  $Q$  rozszerza ograniczenia czasowe  $Q_v$ , wówczas można udzielić odpowiedzi poprzez sprawdzenie wsparcia wzorców przechowywanych w perspektywie z użyciem ograniczeń czasowych  $Q$  (zgodnie z definicją  $Q_v$  dominuje  $Q$ ). Oczywiście w odpowiedzi znajdują się tylko te wzorce z perspektywy MV, które posiadają odpowiednio wysoką wartość współczynnika wsparcia.



Jeżeli  $Q$  i  $Q_v$  operują na tym samym zbiorze danych i  $Q$  rozszerza zarówno ograniczenia eksploracyjne jak i czasowe  $Q_v$  (czyli  $Q_v$  dominuje  $Q$ ), wówczas można udzielić odpowiedzi poprzez sprawdzenie wsparcia wzorców przechowywanych w perspektywie MV z użyciem ograniczeń czasowych  $Q$  i uwzględnieniem ograniczeń eksploracyjnych  $Q$ . Algorytm w pierwszym kroku odczytuje zawartość perspektywy MV i usuwa z niej wszystkie wzorce nie spełniające ograniczeń eksploracyjnych  $Q$ . Następnie podczas pełnego odczytu zbioru danych źródłowych określana jest wartość wsparcia pozostałych wzorców przy użyciu ograniczeń czasowych  $Q$ . W ostatnim kroku algorytm usuwa raz jeszcze te wzorce, których wartość wsparcia jest niższa niż ograniczenie eksploracyjne  $Q$ .

Powyższe metody umożliwiają udzielenie odpowiedzi na zapytanie eksploracyjne dotyczące odkrywania wzorców sekwencyjnych bez konieczności wykonywania kosztownego algorytmu. Wykorzystują one wzorce odkryte w poprzednich sesjach i zmaterializowane w postaci perspektyw. Jak pokazują wyniki eksperymentów, czas odpowiedzi z wykorzystaniem wyników wcześniejszych jest wielokrotnie krótszy niż czas wykonania pełnego algorytmu odkrywania wzorców.

## 5. Podsumowanie

W niniejszym artykule przedstawiono aktualny stan wiedzy dotyczący optymalizacji zapytań eksploracyjnych przy użyciu perspektyw materializowanych. Przeanalizowano możliwości wykorzystania rezultatów wcześniejszych eksploracji do odkrywania zarówno zbiorów częstych, jak i reguł asocjacyjnych i wzorców sekwencyjnych. W tej nowej i fascynującej dziedzinie pozostaje wciąż bardzo wiele otwartych kwestii. Większość przedstawionych algorytmów zakłada, że eksploracja odbywa się na tym samym zbiorze danych, z którego została wyprowadzona perspektywa. Dodatkowo zakłada się, że kształt analizowanych transakcji pozostaje stały. Nierozstrzygnięte pozostają pytania o efektywne metody pielęgnacji materializowanych perspektyw eksploracyjnych. Brakuje wreszcie modeli kosztów dla zapytań eksploracyjnych.

Najbliższa praca autorów skupiać się będzie na rozszerzeniu zakresu stosowalności opisanych metod do zapytań, które różnią się od perspektywy schematem eksplorowanej bazy danych oraz na konstruowaniu modelu kosztów wykonania zapytań eksploracyjnych. Taki model jest konieczny do osiągnięcia ambitnego celu zbudowania optymalizatora szerokiej klasy zapytań eksploracyjnych.

## 6. Literatura

1. Agrawal, R., Imielinski, T., Swami, A., Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD International Conference on Management of Data, Washington, USA, May 1993

2. Agrawal, R., Srikant, R., Fast Algorithms for Mining Association Rules. In Proc. of the 20th International Conference on Very Large Data Bases (VLDB'94), Santiago, Chile, 1994.
3. Agrawal, R., Srikant, R., Mining Sequential Patterns. In Proc. of the 11th International Conference on Data Engineering (IDCDE'95), Taipei, Taiwan, March 1995
4. Agrawal, R., Srikant, R., Mining Sequential Patterns: Generalizations and Performance Improvements. In Proc. of the 5th International Conference on Extending Database Technology (EDBT'96), Avignon, France, September 1996
5. Baralis, E., Psaila, G., Incremental refinement of mining queries. In Proc. of the 1st International Conference on Data Warehousing and Knowledge Discovery (DaWaK'99), Florence, Italy, September 1999
6. Cheung, D.W., Han, J., Ng, V., Wong, C.Y., Maintenance of discovered association rules in large databases: An incremental updating technique. In Proc. of the 12th International Conference on Data Engineering (ICDE'96), New Orleans, USA, February 1996
7. Cheung, D. W., Lee, S. D. and Kao, B., A General Incremental Technique for Maintaining Discovered Association Rules In Proc. of the 5th International Conference on Database Systems for Advanced Applications (DASFAA'97), Melbourne, Australia, April 1997
8. Gupta, A., Mumick, I.S., Maintenance of Materialized Views: Problems, Techniques, and Applications. IEEE Data Engineering Bulletin, Special Issue on Materialized Views and Data Warehousing, 18(2), June 1995
9. Gupta, A., Mumick, I.S., Materialized Views: Techniques, Implementations, and Applications, The MIT Press, 1999
10. Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., Hsu, M.-C., FreeSpan: frequent pattern-projected sequential pattern mining. In Proc. of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'2000), Boston, USA, August 2000
11. Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.-C., Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In Proc. of the 17th International Conference on Data Engineering (ICDE'01), Heidelberg, Germany, April 2001
12. Imielinski, T., Mannila, H., A Database Perspective on Knowledge Discovery. Communications of the ACM, Vol.39, No.11, 1996
13. Morzy, T., Wojciechowski, M., Zakrzewicz, M., Data Mining Query Optimization Using Materialized Views
14. Morzy, T., Zakrzewicz, M., SQL-like Language for Database Mining. In Proc. of the 1st East European Symposium on Advances in Databases and Information Systems (ADBIS'97), St-Petersburg, Russia, September 1997
15. Nag, B., Deshpande, P., DeWitt, D.J., Using a Knowledge Cache for Interactive Discovery of Association Rules. In Proc. of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99), San Diego, USA, August 1999

16. Parthasarathy, S., Zaki, M.J., Ogihara, M., Dwarkadas, S., Incremental and interactive sequence mining. In Proc. of the ACM International Conference on Information and Knowledge Management (CIKM'99), November 1999
17. Thomas, S., Bodagala, S., Alsabti, K., Ranka, S., An Ecient Algorithm for the Incremental Updation of Association Rules in Large Databases. In Proc. of the 3rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'97), Newport Beach, USA, August 1997
18. Toivonen, H., Sampling large databases for association rules. In Proc. of the 22th International Conference on Very Large Data Bases (VLDB'96), Bombay, India, September 1996
19. Wojciechowski, M., Zakrzewicz, M., Itemset Materializing for Fast Mining of Association Rules. In Proc. of the 2nd East European Conference on Advances in Databases and Information Systems (ADBIS'98), Poznań, Poland, September 1998
20. Wojciechowski, M., Interactive Constraint-Based Sequential Pattern Mining. In Proc. of the 5th East European Conference on Advances in Databases and Information Systems (ADBIS'01), Vilnius, Lithuania, September 2001

Mgr inż. Mikołaj MORZY

Instytut Informatyki Politechniki Poznańskiej

ul. Piotrowo 3A tel.: (0-61) 665-21-27

email: Mikołaj.Morzy@cs.put.poznan.pl

Dr inż. Marek WOJCIECHOWSKI

Instytut Informatyki Politechniki Poznańskiej

ul. Piotrowo 3A tel.: (0-61) 665-23-78

email: Marek.Wojciechowski@cs.put.poznan.pl



# ZASTOSOWANIE ZINTEGROWANYCH USŁUG KATALOGOWYCH

Ryszard NIKODEM

**Streszczenie:** W artykule przedstawiono zastosowanie usług katalogowych do administrowania sieciami komputerowymi oraz do integracji danych wykorzystywanych w aplikacjach biznesowych. Modyfikacja, usuwanie oraz wymiana danych między aplikacjami muszą być realizowane przy zagwarantowaniu poprawności i spójności wspólnych danych. Dokonano przeglądu narzędzi programowych rozszerzających możliwości katalogów w tym zakresie.

## Wstęp

W nowoczesnych systemach informatycznych usługi katalogowe znajdują różnorodne zastosowania, w niniejszym artykule skupiono jednak uwagę na integrującej roli usług katalogowych w dwóch obszarach: administrowaniu siecią komputerową oraz funkcjonowaniu całej infrastruktury przedsiębiorstwa, zwłaszcza jego systemu informatycznego i aplikacji biznesowych.

Usługi katalogowe są realizowane przez sieciowe systemy operacyjne i bazują na protokole dostępu do katalogów. Standardem w tym zakresie jest protokół LDAP, zaprezentowany w punkcie 2.

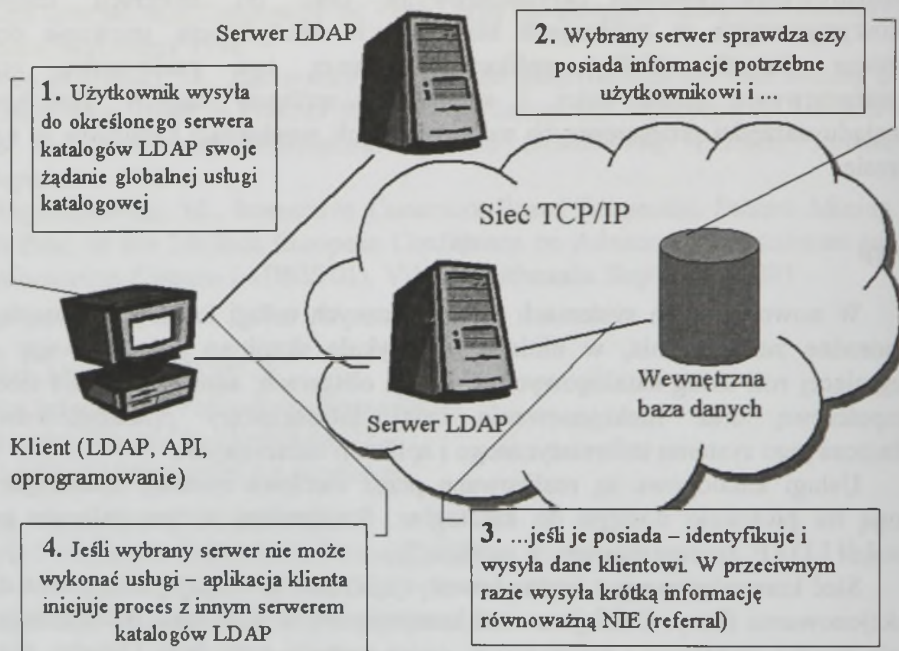
Sieć komputerowa jest ważną inwestycją, często inwestycją strategiczną dla funkcjonowania firmy. Dzisiejsze sieci komputerowe to sieci stale powiększające swoje zasoby sprzętowe i programowe, coraz częściej sieci duże i bardzo duże. Utrzymanie takich sieci w sprawności wymaga monitorowania szeregu parametrów eksploatacyjnych. Administrowanie sieciami komputerowymi przy zastosowaniu usług katalogowych do integracji wielu narzędzi programowych i sprzętowych, wspomagających administratora w jego zadaniu, przedstawiono w punkcie 3.

Usługi katalogowe znajdują również zastosowanie przy integracji danych zbieranych i przetwarzanych przez różne aplikacje biznesowe. W punkcie 4 omówiono zasady działania narzędzi stanowiących interfejs między aplikacjami biznesowymi i usługami katalogowymi.

## 1. Protokół LDAP i usługi katalogowe

Usługi katalogowe są udostępniane przez protokół LDAP (*Lightweight Directory Access Protocol*), który wywodzi się z standardu X.500 w modelu OSI (*Open System Interconnect*), a dokładniej – z protokołu DAP (*Directory Access Protocol*) dostępu do katalogów, i umożliwia dostęp do katalogów adresowych w

sieciach TCP/IP. W praktyce jest stosowany w sieciach Internet i intranet w celu zapewnienia niezawodności, skalowalności i bezpieczeństwa danych. W odróżnieniu od swego poprzednika – protokołu DAP, który wykorzystuje cały stos OSI, co stawia wysokie wymagania zasobom systemu – nie potrzebuje ani szerokiego pasma (co podkreśla słowo “lekki” w nazwie protokołu), ani dużej mocy obliczeniowej. Obecnie protokół ten jest nieformalnym standardem w zakresie dostępu do usług katalogowych.



Rys. 1. Zasada działania protokołu udostępniania usług katalogowych LDAP  
 Źródło: opracowanie własne na podstawie (Vademecum, 1988, s.168).

Dla administratorów sieci komputerowych protokół LDAP – jako znormalizowana metoda dostępu do katalogów – stanowi swego rodzaju esperanto (Janikowski, 2000), cechuje się prostotą i uniwersalnością, zamiast scentralizowanego katalogu preferuje stosowanie w sieci przedsiębiorstwa zatomizowanych katalogów rozproszonych. Zasada funkcjonowania (patrz rys. 1) jest następująca: w sieci współpracuje wiele serwerów katalogu LDAP, z których każdy przechowuje w pamięci cały lub fragment katalogu przedsiębiorstwa. Jeżeli żądanie skierowane przez klienta LDAP do serwera katalogu nie może być zrealizowane lokalnie, jest przesyłane do realizacji do innego serwera, przy czym lokalny serwer ponosi odpowiedzialność za wykonanie zapytania. Ewentualne odwołania między serwerami są dla klienta “przezroczyste” – dla użytkownika jest to zawsze zapytanie do wspólnej struktury katalogu, tyle że przechowywanego w różnych miejscach.

Zaletą rozwiązania zaproponowanego przez protokół LDAP jest możliwość tworzenia i użytkowania małych, lokalnych katalogów, zawierających dane o użytkownikach, i zasobach sieciowych przypisanych do sieci lokalnych, będących segmentami złożonych sieci komputerowych. W ten sposób fragmenty lokalnych katalogów replikowane do baz centralnych dotyczą tylko niewielkiej części danych, łatwe też staje się rozdzielanie funkcji i uprawnień administratora centralnego. Administrator lokalny posiada niezbędne uprawnienia w swojej domenie, może dodawać i usuwać użytkowników, przypisywać im usługi i zasoby sieciowe, tworzyć grupy robocze zgodnie z potrzebami specyficznymi dla konkretnej filii przedsiębiorstwa. Zapewnia to dużą elastyczność i odwzorowanie struktury organizacyjnej wielkich przedsiębiorstw z wieloma autonomicznymi oddziałami.

Katalogi przechowywane w serwerach LDAP przypominają relacyjne bazy danych i implementują takie operacje – podstawowe dla systemów zarządzania bazami danych – jak przeszukiwanie, modyfikowanie, dodawanie i usuwanie zapisów (zakłada się, że wyszukiwanie w katalogach jest realizowane znacznie częściej niż operacje aktywne, czyli zmieniające zawartość katalogu – w przeciwnym przypadku korzyści z tytułu replikowania fragmentów katalogu byłyby problematyczne, wobec kosztów częstej aktualizacji replik). Inne operacje, typowe dla relacyjnych baz danych, takie jak realizacja transakcji lub jej cofnięcie (rollback), nie są wykonywane, co wynika ze specyfiki protokołu udostępniania usług katalogowych. Serwery LDAP cechuje stosowanie uproszczonego modelu danych, duża szybkość, małe koszty i ułatwione zarządzanie. W sposób naturalny katalogi pozwalają odwzorować hierarchiczną strukturę organizacji, obejmującą np. takie obiekty jak kontynent, kraj, województwo i miasto, przez jednostki organizacyjne, filie, wydziały, aż po sale konferencyjne oraz komputery, drukarki i skanery w pokojach pracowników.

Technologia LDAP pozwala budować katalogi o praktycznie dowolnej skali. W dużych firmach, prowadzących np. handel elektroniczny, typowe sytuacje to katalogi o wielkości kilkudziesięciu milionów wpisów i milionach zapytań dziennie. Skalowalność rozwiązania uzyskuje się w wyniku dodania kolejnego serwera do serwerów już współpracujących w realizacji usług katalogowych.

Na rynku produktów realizujących usługi katalogowe liczą się praktycznie dwie pozycje: NDS firmy Novell i Active Directory firmy Microsoft, przy czym niekwestionowanym liderem jest oprogramowanie NDS (*Novell Directory Services*), włączone do systemu operacyjnego NetWare (począwszy od wersji 4), które umożliwia uporządkowanie informacji o sieci, pozwala na odwzorowanie struktury firmy w drzewie katalogowym, organizuje dostęp użytkowników do poszczególnych zasobów systemu informatycznego, w konsekwencji – poprawia jakość korzystania z sieci i zwiększa bezpieczeństwo systemu. Rozwiązanie to można określić jako rozproszone usługi katalogowe, inwentaryzujące wszystkie zasoby sieci komputerowej, niezależnie od ich fizycznej lokalizacji. Hierarchiczna struktura drzewa NDS jest zgodna ze standardem X.500.



Z oprogramowania NDS korzystają wszystkie najnowsze produkty firmy Novell, może być też wykorzystywane w systemach sieciowych innych firm komputerowych, m.in. w Windows NT, Windows 2000 i większości systemów Unix (m.in. Linux, Solaris, SCO, AIX).

Dane katalogowe gromadzone i przetwarzane przez NDS można przechowywać na jednym serwerze, ale można też replikować na inne serwery – dzięki temu będą odporne na awarie sprzętu. W przypadku utraty zasobów jednego z serwerów, informacje o użytkownikach sieci zostaną odtworzone z replik na innych serwerach.

## 2. Administrowanie sieciami komputerowymi

W sieciach lokalnych, obsługiwanych przez kilka serwerów, serwer usług katalogowych (np. serwer NDS) można traktować jako podstawową usługę sieciową, która zapewnia (Kaliński, 1999):

- bazę danych, umożliwiającą zarządzanie zasobami i usługami sieci,
- standardową metodę dostępu do usług, zasobów, informacji sieciowej,
- dynamiczne mapowanie pomiędzy obiektami fizycznymi a zasobami, do których się odnosi,
- pojedynczy punkt logowania.

Usługi NDS ułatwiają odwzorowania struktury firmy. O ile w małych lokalnych sieciach komputerowych ma to niewielkie znaczenie, o tyle przy średnich i dużych sieciach zaczyna odgrywać coraz większą rolę. Podzielenie drzewa np. na pionowy (produkcji, finansowy, marketingu, itd.) pozwala uzyskać pełny obraz zależności między działami, ponadto uporządkować informacje o pracownikach, klientach, serwerach czy grupach roboczych. Administrator ma pełny obraz pracy sieci i możliwość przewidywania przyszłej jej rozbudowy, np. dodania serwera do obsługi wybranych aplikacji.

Można stwierdzić, że im większa jest sieć, tym większa potrzeba stosowania usług katalogowych. W sieciach rozległych globalne usługi katalogowe mają podstawowe znaczenie dla sprawnego działania sieci. Nie tylko zapewniają dostęp do zasobów komputerowych w skali całej firmy, lecz również znacznie ułatwiają zarządzanie siecią i całym przedsiębiorstwem, wspomagają administratora w realizacji rozproszonego zarządzania. Istnieje możliwość przypisania poszczególnym gałęziom drzewa NDS różnych administratorów i nadania każdemu administratorowi prawa do wybranego kontenera (np. oddziału firmy) i związanych z nim obiektów. Rozwiązanie takie ma następujące zalety (Kaliński, 1999):

- jeśli administrowanie kontenerem ma zostać przekazane innemu użytkownikowi sieci, to wystarczy przypisać go temu obiektowi;
- jeśli kilku użytkowników ma pełnić rolę administratora kontenera, to można ich do tego obiektu przypisać;

- jeśli część kontenerów jest zbyt mała, aby każdy z nich miał swojego administratora, to do większej ich liczby można przypisać jednego administratora.

Można zatem w prosty sposób zorganizować rozproszone drzewo NDS, w którym czynności administracyjne nie będą skupiać się na jednej tylko osobie lub wąskiej grupie osób. Z kolei funkcja kontrolowania pozwala nadzorować działania innych użytkowników. Kontroler może na przykład sprawdzać, czy administrator przestrzega zasad przyjętych w firmie.

Technologia NDS pozwala na nieprzerwaną pracę sieci także w czasie, gdy administrator wprowadza zmiany w strukturze katalogów. Zwiększa to nie tylko komfort pracy użytkowników, ale również redukuje koszty administrowania siecią.

Firmy Novell i Microsoft, czyli obaj renomowani dostawcy systemów usług katalogowych, w kolejnych wersjach swoich produktów poszerzają zakres funkcji zarządzania. W odniesieniu do administrowania sieciami komputerowymi usługi katalogowe wspierane są narzędziami do zarządzania użytkownikami, komputerami, ośrodkami, usługami, domenami i uprawnieniami.

Ułatwienia w obsłudze katalogów obejmują także zastosowanie zgodnie z modelem WBEM techniki webowej, tj. zarządzanie z wykorzystaniem do monitorowania stanu sieci komputerowej przeglądarki internetowej (Nikodem, 2001). Na przykład w wersji 6 sieciowego systemu operacyjnego NetWare firmy Novell do wspomagania zarządzania siecią komputerową służy przeglądarka NetWare Remote Manager, która umożliwia – poza zmianami w konfiguracji systemu – wyświetlanie z dowolnego komputera w sieci ekranu konsoli serwera. Administrator ma w ten sposób kontrolę nad wszystkimi aplikacjami działającymi na serwerze. W przeglądarce może też wyświetlić strukturę katalogów oraz wyłączać i restartować serwery.

Jednocześnie oprogramowanie wymienionych firm nie gwarantuje takich możliwości wymuszenia integralności danych jak narzędzia dostawców niezależnych, którzy ponadto rozszerzają usługi katalogowe o oparte na rolach delegowanie zadań administratora. Oferowane intuicyjne narzędzia do delegowania zadań administracyjnych umożliwiają samoobsługę, polegającą na przekazywaniu wybranych zadań użytkownikom i odciążeniu w ten sposób administratora sieci.

Przykładem pakietu do delegowania zadań i wymuszenia reguł funkcjonowania jest FastLane Active Roles firmy Quest Software, współpracujący tylko z usługami katalogowymi firmowanymi przez Microsoft. Bardziej uniwersalnym rozwiązaniem jest pakiet WebConsole firmy PatchLink, który integruje zainstalowane w sieci serwery NDS i *Active Directory*, ponadto zapewnia interfejs webowy do katalogów i pozwala na delegowanie wskazanych zadań administracyjnych do użytkowników.

Do innej grupy narzędzi wspomagających należą produkty bvControl firmyBindView i DRA firmy NetIQ (pierwszy integruje katalogi NetWare i Windows, drugi ma zakres działania ograniczony do środowiska Windows). Ich zadaniem jest scentralizowane monitorowanie i raportowanie funkcjonowania sieci oraz



wymuszanie integralności danych. Centralna konsola umożliwia przeprowadzenie kwerend (sformułowanych w postaci list selekcyjnych, ewentualnie uzupełnionych kryteriami filtrowania) i na ich podstawie realizację funkcji zarządzania siecią komputerową (Zarządzanie, 2001).

### 3. Integracja danych biznesowych

Dla aplikacji i usług z zakresu e-biznesu firma Novell wprowadziła na rynek oprogramowanie rozszerzającą usługi katalogowe NDS – między nimi szczególnie godne uwagi wydają się pakiety NDS eDirectory i Novell Account Management. Istotnymi zaletami tych propozycji są możliwość współpracy z serwerami NetWare, Windows i uniksowymi, czyli praktycznie z wszystkimi stosowanymi powszechnie w sieciach komputerowych heterogenicznych (niejednorodnych) oraz skalowalność, która pozwala łatwo rozbudowywać konfiguracje sieciowe i obsługiwać katalogi zawierające miliardy obiektów.

Pakiet NDS Directory został oparty na usługach katalogowych NDS, ułatwia gromadzenie danych indywidualnych i ma zastosowanie wszędzie tam, gdzie jest wymagana obsługa transakcji realizowanych przez Internet i obsługa indywidualnych profili użytkowników; pozwala przedsięwzięcia e-biznesowe dostosować do oczekiwań i wymagań poszczególnych użytkowników.

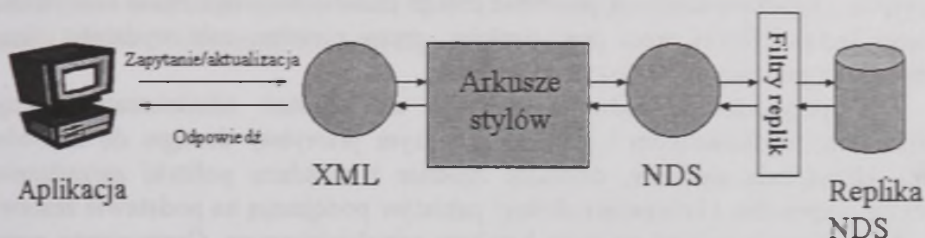
Drugi pakiet – Novell Account Management 2.1 (dawniej znany pod nazwą NDS Corporate Edition) – został z kolei zbudowany na bazie wspomnianego powyżej NDS eDirectory i służy do zarządzania zasobami sieciowymi. Jest to rozwiązanie wieloplatformowe, zatem umożliwia integrację w firmie wszystkich zasobów połączonych siecią komputerową, co w efekcie pozwala administratorowi z jednego punktu (komputera) efektywnie zarządzać zasobami rozproszonymi, zarówno sprzętowymi, jak i programowymi, m.in. serwerami, stacjami roboczymi, zaporami *firewall*, pocztą elektroniczną, plikami, wydrukami, współużytkowanymi zasobami, aplikacjami, kontami użytkowników (Głowiński, 2000a).

Z analizy powyższej listy zarządzanych zasobów można zauważyć, że nie sposób do końca rozdzielić zastosowań usług katalogowych do administrowania sieciami komputerowymi od związanych z danymi biznesowymi – obszary te spotykają się m.in. jako przedsięwzięcia e-biznesowe.

Szczególnie ważne jest zastosowanie usług katalogowych do integracji danych biznesowych. Integracja danych wspólnych dla wielu aplikacji stanowi zazwyczaj poważny problem. Pomimo istnienia wielu standardów, a może właśnie z powodu wielości standardów wzajemnie niezgodnych, formaty danych zapisywanych w bazach danych różnych aplikacji utrudniają wymianę współdzielonych informacji. Nawet w sytuacji, gdy ma miejsce zgodność formatów, pamiętanie powtarzających się informacji w dokumentach i bazach różnych aplikacji stanowi zagrożenie dla integralności danych, w przypadku, aktualizowania jednego wystąpienia i pozostawienia niezmiennych wartości w pozostałych kopiach.



Opisanym powyżej zagrożeniom można zaradzić stosując usługi katalogowe. Problem jednak w tym, że korzystanie przez aplikacje z zawartości katalogu wymaga dopisania dodatkowego fragmentu kodu programu do każdej aplikacji. Sytuacja uległa poprawie od chwili wprowadzenia standardu XML (*Extensible Markup Language*). Z połączenia usług katalogowych i napisanych w języku XML procedur rozszerzających dostępność usług powstaje uniwersalny interfejs do danych katalogowych. Przykładem narzędzia tego typu jest pakiet DirXML firmy Novell, który pozwala dostarczyć do aplikacji informacje gromadzone w drzewie katalogu NDS i przedstawić je w formacie własnym aplikacji.



Rys. 2. Schemat realizacji dostępu do danych katalogowych w DirXML

Źródło: opracowanie własne na podstawie (Głowiński, 2000b).

Pakiet DirXML udostępnia aplikacjom dane z wirtualnych replik katalogowych NDS (por. rys. 2). Mechanizmy usług katalogowych NDS pozwalają tworzyć tego typu repliki lokalnie w węzłach sieci (są zatem łatwo dostępne dla aplikacji, przy minimalizacji wielkości danych przesyłanych w sieci) i synchronizować je między sobą. Ponieważ proces replikacji jest dwukierunkowy, zapytania do replik mogą być pasywne (tj. dotyczyć tylko ekstrakcji danych z katalogu) lub aktywne, czyli aktualizujące zawartość katalogu (tego rodzaju operacje są zarezerwowane dla administratora lub osób przez niego upoważnionych).

Dostęp do wirtualnych replik odbywa się przez filtry replik, przechowywane jako obiekty w drzewie NDS. Filtry opisują sposób wybierania danych z katalogu do wirtualnej repliki, dzięki czemu aplikacji udostępnia się tylko te dane, które są w niej wymagane. Jednocześnie przy aktualizacji danych są przesyłane jedynie dane, których wartości zmodyfikowano, co zmniejsza obciążenie sieci.

Zastosowanie pakietu DirXML jako narzędzia integracji współdzielonych danych nie wymaga modyfikacji istniejących aplikacji, eliminuje konieczność posługiwania się formatami specyficznymi dla usług katalogowych. Interfejs między aplikacją i drzewem katalogów NDS stanowią arkusze stylów, zapisane w języku XSL (*Extensible Stylesheet Language*), jako proste pliki tekstowe opisujące reguły przetwarzania danych eksportowanych lub importowanych do katalogu NDS.

#### 4. Uwagi końcowe

W artykule nie wyczerpano tematu, a jedynie zasygnalizowano wybrane zagadnienia związane z wykorzystaniem usług katalogowych do integracji danych w przedsiębiorstwie. Poza obszarem zainteresowań pozostała np. ciekawa propozycja dotycząca nowych możliwości zastosowań usług katalogowych do zarządzania sieciami komputerowymi. Polega ona na wprowadzeniu do katalogów dodatkowych kategorii danych, które reprezentują urządzenia kierujące ruchem pakietów w sieci (routery, przełączniki itp.), usługi sieciowe oraz zasady ochrony i zarządzania siecią. Umożliwia to szersze wykorzystanie usług katalogowych do zarządzania siecią opartego na regułach. Koncepcja takiego zarządzania polega na przyjęciu i egzekwowaniu na poziomie całego przedsiębiorstwa zasad korzystania z sieci komputerowej przez pracowników, grupy robocze, całe wydziały i inne jednostki organizacyjne (Muszyński, 1999a).

W przypadku zarządzania opartego na regułach administrator nadaje aplikacjom, użytkownikom i grupom roboczym priorytety dostępu do zasobów sieci. Urządzenia sieciowe, działając zgodnie z zasadami polityki zarządzania, decyzje o sposobie i kolejności obsługi pakietów podejmują na podstawie zestawu reguł, zapisanych w poszerzonym katalogu przedsiębiorstwa. Opracowany przez grupę DMTF (*Desktop Management Task Force*) standard DEN (*Directory Enabled Network*) opisuje schemat tak rozbudowanych usług katalogowych i model definiowania urządzeń sieciowych, usług sieciowych i ochrony oraz jakości usług. Przewiduje się, że usługi katalogowe działające w nowym standardzie DEN będą dla aplikacji wspólnym miejscem wymiany informacji i zdarzeń (Muszyński, 1999b).

Warto zauważyć, że jakkolwiek należy spodziewać się nowych zastosowań usług katalogowych do administrowania sieciami komputerowymi i integracji danych, to już obecnie dostępne narzędzia w tym zakresie, szeroko i właściwie stosowane, mogą przynosić wymierne korzyści w praktyce gospodarczej.

#### Literatura

1. Głowiński C., (2000a): Usługi katalogowe w e-biznesie, "NetForum" nr 6, s. 56-57.
2. Głowiński C., (2000b): Integracja danych za pomocą DirXML, "NetForum" nr 6, s. 60-61.
3. Janikowski A., (2000): LDAP – co nowego? "NetWorld" nr 1, s.55-59.
4. Kaliński J., (1999): Novellowe katalogi. "PC Kurier" nr 18.
5. Muszyński J., (1999a): Oprogramowanie 2000. [w:] "Teleinformatyka 2000, cz.2", "Net World", wydanie specjalne, s.35-50.
6. Muszyński J., (1999b): Usługi katalogowe w narzędziach zarządzania siecią. "NetWorld" nr 2, s.41-43.

7. Nikodem R., (2001): Technologie zarządzania sieciami komputerowymi. [w:] Systemy wspomagania organizacji. Red. J. Gołuchowski i H. Sroka. Akademia Ekonomiczna Katowice, s. 409-414.
8. Vademecum teleinformatyka, cz.2, (1988), „NetWorld”, wydanie specjalne.
9. Zarządzanie usługami katalogowymi, (2001), „NetWorld” nr 10, s.44-50.

dr inż. Ryszard Nikodem

Instytut Informatyki Ekonomicznej Akademia Ekonomiczna we Wrocławiu

53-345 Wrocław, ul. Komandorska 118/120, tel/fax 0(prefiks)71 3680-376

e-mail: [Ryszard.Nikodem@ae.wroc.pl](mailto:Ryszard.Nikodem@ae.wroc.pl)





# GENETYCZNA KONCEPCJA ZAGOSPODAROWANIA ROZPROSZONYCH ZASOBÓW

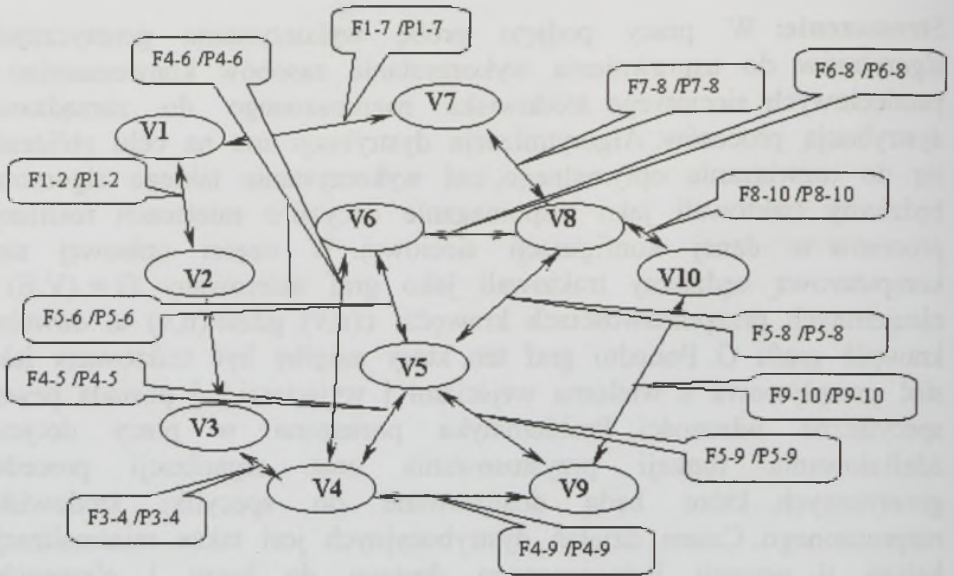
Henryk PIECH, Łukasz KUCZYŃSKI, Dariusz LEKS

**Streszczenie:** W pracy podjęto próbę wykorzystania genetycznych algorytmów do usprawnienia wykorzystania zasobów konwersacyjno – pamięciowych sieciowego środowiska rozproszonego do zarządzania dystrybucją procesów. Algorytmizacja dystrybucji ma na celu zbliżenie się do rozwiązania optymalnego, zaś wykorzystanie takiego algorytmu będziemy traktowali jako wspomaganie decyzji o miejscach realizacji procesów w danej konfiguracji sieciowej. W części opisowej sieć komputerową będziemy traktowali jako graf skierowany  $G = (V, E)$  o nieujemnych przepustowościach krawędzi  $c(u, v)$ , gdzie  $(u, v)$  to dowolna krawędź grafu  $G$ . Ponadto graf ten, który mógłby być traktowany jako sieć przepływowa z wieloma wejściami i wyjściami [ ] posiada pewne specyficzne własności. Problematyka poruszona w pracy dotyczy zdefiniowania funkcji przystosowania oraz organizacji procedur genetycznych, które będą dostosowane do specyfiki środowiska rozproszonego. Celem działań dystrybucyjnych jest także minimalizacja kolizji tj. sytuacji jednoczesnego dostępu do łączy i elementów przetwarzających.

## 1. Opis zasobów procesowych i konwersyjnych środowiska rozproszonego

Środowisko rozproszone można opisać jako sieć przepływową [ ] o wielu wejściach i wyjściach  $G(V, E)$ , w której każda krawędź ma nieujemną przepustowość  $c(u, v) > 0$ , którą możemy uznać za wielkość stałą i dwukierunkową (jednak wykluczającą jednoczesny dwukierunkowy, a nawet jednokierunkowy tranzyt). Na rys. 1 jest przedstawiona przykładowa struktura rozproszona. Każdy węzeł sieci charakteryzuje się następującymi parametrami: moc obliczeniowa ( $P_w$ ), zasoby dostępnej, lokalnej pamięci ( $M$ ) oraz stopień zajętości jednostki obliczeniowej ( $S_z$ ). Stopień zajętości można, z pewnym przybliżeniem, przedstawić zarówno jako zmniejszenie mocy obliczeniowej oraz zmniejszenie zasobów pamięciowych  $S_z = \Delta P_w \& \Delta M$ , gdzie  $\&$  oznacza łącznik składowych złożonego parametru. Elementami wejściowymi w strukturze sieciowej są procesy o określonej złożoności ( $Z$ ), która składa się z komponentów czasowych i pamięciowych. Decyzje dystrybucyjne polegają na skierowaniu procesów do jednostek przetwarzających. Decyzje, w uproszczonej interpretacji, mogą mieć charakter strategiczny tj. uwzględniający optymalną

dystrybucję w pewnym okresie czasowym lub bieżący uwzględniający aktualne potrzeby w zakresie przetwarzania.

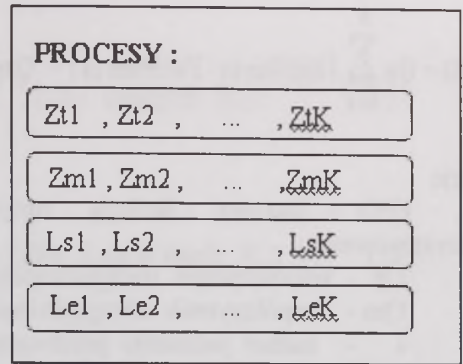
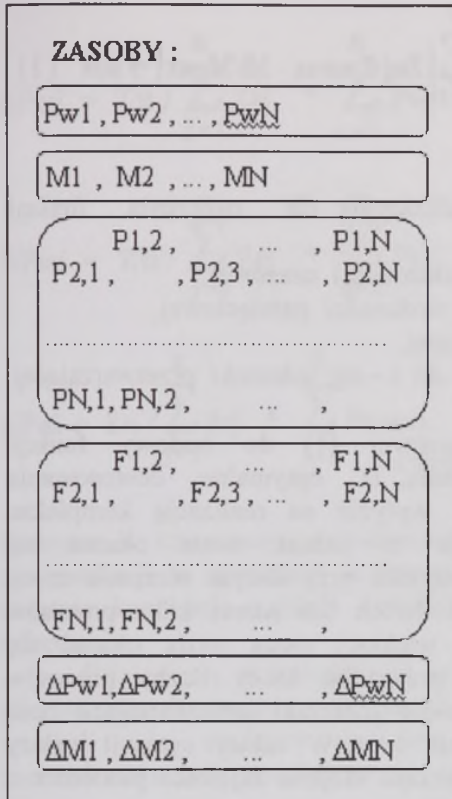


Rys.1 Sieniowa struktura rozproszona, gdzie  $V_i$  – węzeł sieci,  $F_i-j/P_i-j$  - charakterystyka łączy (krawędzi grafu) - przepływ / przepustowość (oddzielone separatorem „/”)

Również pojęcie optymalności dystrybucji może być zdefiniowane na różne sposoby, można na przykład dążyć do minimalizacji czasu realizacji wszystkich zadań lub minimalizacji zajętości jednostek przetwarzających, minimalizacji zajętości zasobów pamięciowych albo też minimalizacji przesyłów komunikacyjnych. W strukturach rozproszonych o charakterze rozległym należy uwzględnić przepustowości (P) oraz wielkości przesyłanej informacji (przepływy - F). Dane środowiskowe można więc podzielić na dwie grupy: dane dotyczące posiadanych zasobów przetwarzających i wspomagających i dane dotyczące procesów, które mają być zrealizowane (rys. 2).

Celem dalszych rozważań będzie tworzenie modeli optymalizujących procedury dystrybucji procesów w środowisku rozproszonym.





N – ilość procesorów,  
 K – ilość procesów,  
 $Zt$  – złożoność czasowa procesu,  
 $Zm$  – złożoność pamięciowa procesu,  
 $Ls$  – lokalizacja wejściowa procesu,  
 $Le$  – lokalizacja wyjściowa procesu tj.  
 miejsce jego rozdystrybuowania.

Rys 2. Koncepcja struktury parametrów środowiska rozproszonego

## 2. Koncepcje funkcji przystosowania

Funkcję przystosowania należy odnieść do funkcji celu oraz dostosować ją do postaci parametrów dystrybucji. Wybór funkcji celu zależy od wielu towarzyszących okoliczności związanych między innymi z

- rozrzutem złożoności czasowej i pamięciowej zadań,
- ilością przesyłanych danych,
- rozproszeniem mocy obliczeniowej poszczególnych węzłów sieci,
- zróżnicowaniem przepustowości poszczególnych łączy.

Jedną z koncepcji wykorzystywanych przy kreowaniu funkcji przystosowania jest dystrybucja zadań stosownie do ich złożoności i z uwzględnieniem mocy obliczeniowej oraz zasobów pamięciowych jednostek (węzłów) przetwarzających.

$$FPO = Q_p \sum_{i=1}^N |Z_{tj}/Z_{tm \max} - P_{wi}/P_{wm \max}| + Q_m \sum_{i=1}^N |Z_{mj}/Z_{mm \max} - M_i/M_{m \max}| \rightarrow \min \quad (1)$$

gdzie  
FPO – bazowa funkcja optymalizacyjna dla tworzenia funkcji przystosowania,

$Q_p$  - współczynnik uwzględnienia złożoności czasowej,

$Q_m$  - współczynnik uwzględnienia złożoności pamięciowej,

$i$  - numer jednostki przetwarzającej,

$j$  - numer procesu przypisanego do  $i$ -tej jednostki przetwarzającej  
 $j = 1, 2, \dots, K$  ( $K \leq N$ ).

W zamyśle, przy wyborze funkcji bazowej (1) do budowy funkcji przystosowania, oparto się na założeniu, iż optymalne dostosowanie złożoności procesu do mocy konwertera wpłynie na realizację kompleksu zadań w najkrótszym czasie. Założenie to jednak może okazać się nieprecyzyjne lub nawet niesłuszne na przykład przy dużym rozrzucie mocy obliczeniowej. W takiej sytuacji realizacja dwóch lub nawet kilku procesów przez jeden z procesorów o znacznie większej mocy może okazać się efektywniejsza i szybsza. Również w przypadku kiedy liczba procesów przewyższa liczbę procesorów ( $K < N$ ) pewne jednostki przetwarzające będą wykonywały większą ilość procesów niż inne. W takiej sytuacji należy wprowadzić do formuły (1) składowe dotyczące stopnia zajętości procesora.

$$FPO = Q_p \sum_{i=1}^N \left\{ |Z_{tj}/Z_{tm \max} - [P_{wi} - \sum_{\substack{l=1 \\ l \neq j}}^{q_i} \Delta P_{wl}] / P_{w \max}| \right\} +$$

$$+ Q_m \sum_{i=1}^N \left\{ |Z_{mj}/Z_{mm \max} - [M_i - \sum_{\substack{l=1 \\ l \neq j}}^{q_i} \Delta M_l] / M_{m \max}| \right\} \rightarrow \min \quad (2)$$

gdzie  $q_i$  – długość kolejki do  $i$ -tego węzła przetwarzającego,

$\Delta P_w, \Delta M$  – składowe stopni zajętości przez poszczególne procesy.

Oszacowanie parametrów  $\Delta P_w, \Delta M$  można przeprowadzić na różne sposoby. Na przykład wykorzystanie technik normalizacyjnych można odnieść do średniej mocy obliczeniowej, sumarycznej mocy obliczeniowej albo też maksymalnej mocy jednego z węzłów sieci.

$$\Delta P_{wi} = Z_{ti} / \sum_{j=1}^K Z_{tj} * \sum_{l=1}^N P_{wl} / N \quad (\text{dla małych } K) \quad (3)$$

$$\Delta P_{wi} = Z_{ti} / \sum_{j=1}^K Z_{tj} * \sum_{l=1}^N P_{wl} \quad (\text{dla dowolnych } K) \quad (4)$$

$$\Delta P_{wi} = Z_{ti} / \sum_{j=1}^K Z_{tj} * \sum_{l=1}^N P_{wmax} \quad (\text{dla średnich } K \text{ i małego rozrzutu mocy obliczeniowej}) \quad (5)$$

Parametr  $K$  szacowany jest odnośnie liczby węzłów sieci  $N$  (uwaga dotyczy rekomendacji w nawiasach w formułach 3, 4, 5).

Ostatnia koncepcja zakłada uwzględnienie przepustowości oraz przepływów informacji.

Odniesienie przepływającej informacji do przepustowości danego łącza stanowi parametr rozproszenia danych. Aby opisać stan rozproszenia informacji w skali całej struktury sieciowej można wykorzystać wartość wariancji znormalizowanych przepływów co przedstawić można następująco:

$$V = 1 / N^2 \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N (F_{i-j} / P_{i-j} - A_v(F/P))^2 \quad (6)$$

gdzie  $V$  – współczynnik rozproszenia przesyłanej informacji w sieci,

$A_v(*)$  – średnia wartości „\*” obliczona tylko dla łączy aktywnych,

$F_{i-j} / P_{i-j}$  - znormalizowany przepływ od węzła „i” do węzła „j”; wartość ta jest równa zero jeżeli między punktami nie istnieje łącze lub jeśli jest ono nie wykorzystane (łącze nieaktywne).

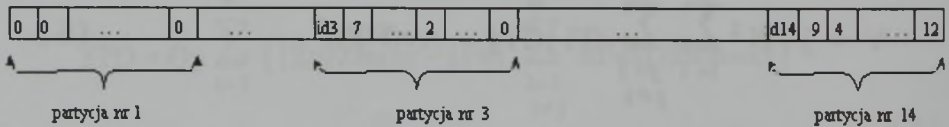
We wzorze (6) miast średniej możemy używać wartości absolutnych. Tak czy inaczej składowa  $V$  w sytuacji optymalnej dystrybucji winna osiągnąć wartość minimalną. Można więc bez skrupułów dołączyć ją do struktury funkcji przystosowania, a ostateczna jej postać przybierze następującą formę: W wyrażeniu (7) uwzględniono zarówno złożoność obliczeniową wejściowych procesów, zajętość procesorów jak i przepływ informacji po łączach o różnych przepustowościach.



$$\begin{aligned}
FPO = & Q_p \sum_{i=1}^N \left\{ \left| Z_{tj} / Z_{tmax} - \left[ P_{wi} - \sum_{\substack{l=1 \\ l \neq j}}^{q_i} \Delta P_{wl} \right] / P_{wmax} \right| \right\} + \\
& + Q_m \sum_{i=1}^N \left\{ \left| Z_{mj} / Z_{mmax} - \left[ M_i - \sum_{\substack{l=1 \\ l \neq j}}^{q_i} \Delta M_l \right] / M_{max} \right| \right\} + \rightarrow \min \quad (7) \\
& + 1 / N^2 \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N (F_{i-j} / P_{i-j} - A_v(F/P))^2 \rightarrow \min
\end{aligned}$$

### 3. Koncepcja kodowania struktury chromosomu .

W proponowanym podejściu zakładamy, iż w chromosomie będzie odzwierciedlone wejście procesu do sieci i droga do procesora realizującego proces. Pozostałe pola mogą być wyłączone z przetwarzania genetycznego bądź mogą stanowić uzupełniające weryfikowanie wymiany genetycznych informacji.



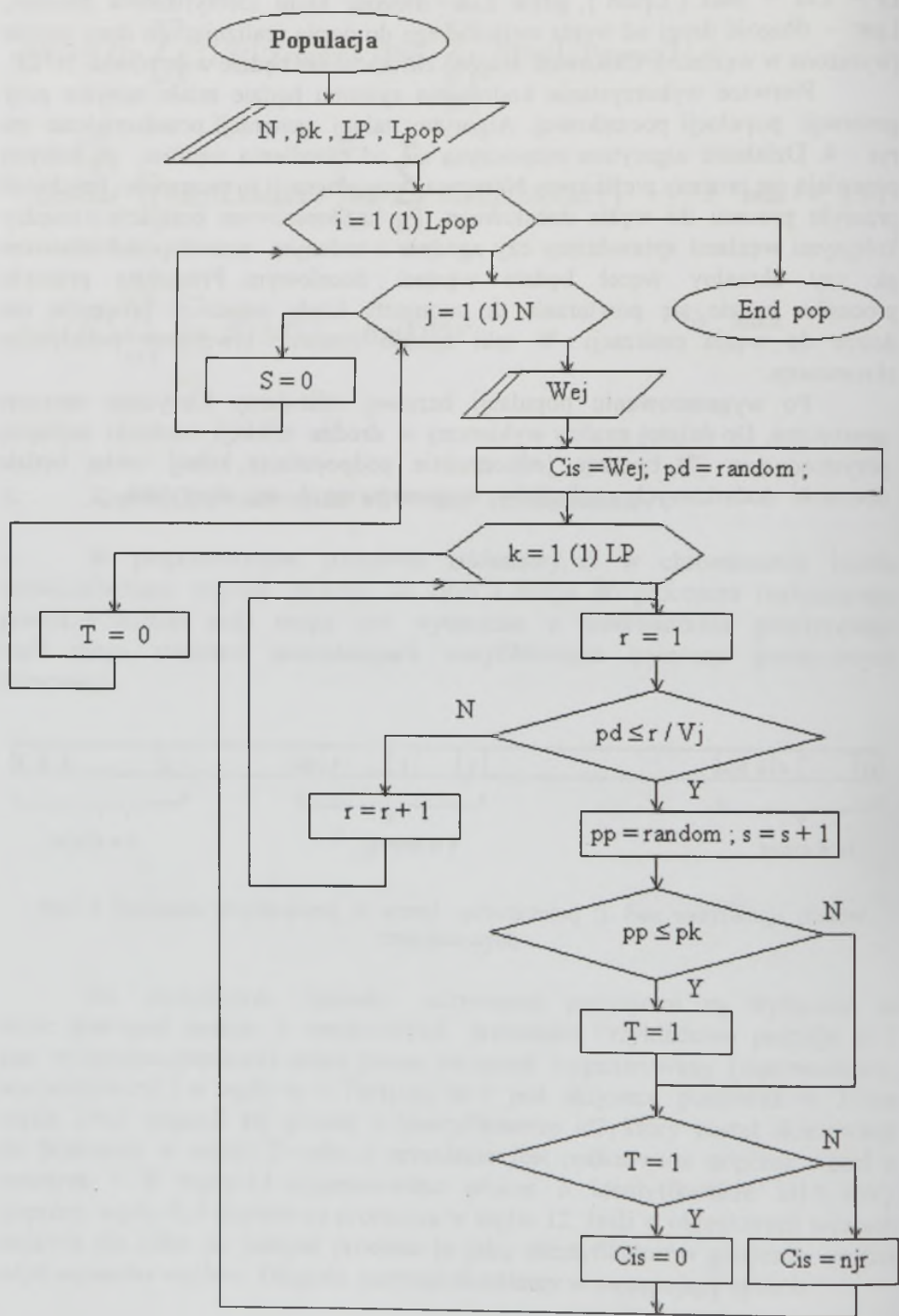
Rys. 3 Struktura chromosomu w wersji uproszczonej tj. bez weryfikacji stanów przejściowych

Na powyższym rysunku aktywnymi partycjami są wyłącznie te które powiązać można z wejściowymi procesami. Przykładowo partycja nr 1 jest wyłączona, ponieważ żaden proces nie został wygenerowany (wprowadzony, wyindukowany) w węźle nr 1. Partycja nr 3 jest aktywna, ponieważ w 3-cim węźle sieci pojawił się proces o identyfikatorze id3, który został skierowany do procesora w węźle 2 – gim, a przesłanie jest realizowane poprzez węzeł o numerze 7. W węźle 14 wygenerowano proces o identyfikatorze id14, który poprzez węzły 9, 4 dociera do procesora w węźle 12. Jeśli w określonych węzłach pojawia się tylko po jednym procesie to jako identyfikatory procesów można użyć numerów węzłów. Długość partycji określamy w następujący sposób

$LP = Lid + \max \{ Lpath \}$ , gdzie  $Lid$  – długość kodu identyfikatora procesu,  $Lpat$  - długość drogi od węzła wejściowego do węzła realizującego dany proces (wyrażona w węzłach). Całkowita długość chromosomu będzie więc równa  $N*LP$ .

Pierwsze wykorzystanie kodowania systemu będzie miało miejsce przy generacji populacji początkowej. Algorytm takiej generacji przedstawiono na rys . 4. Działanie algorytmu rozpoczyna się od określenia węzłów, w których pojawiają się procesy wejściowe. Następną fazą generacji to tworzenie trajektorii przesyłu procesu do węzła docelowego. Po każdorazowym przejściu między kolejnymi węzłami sprawdzamy czy zgodnie z zadaniem prawdopodobieństwem  $p_k$  czy aktualny węzeł będzie węzłem docelowym. Procedura przesyłu procesów będzie się powtarzała do momentu kiedy ostatni z procesów nie dotrze do węzła realizacji . W taki sposób zostanie utworzony pojedynczy chromosom.

Po wygenerowaniu populacji bazowej realizujemy klasyczne operacje genetyczne. Do dalszej analizy wybieramy w drodze selekcji osobniki najlepiej przystosowane (7), tworząc jednocześnie podpopulację, której cechą będzie obecność dodatkowych osobników wygenerowanych wg algorytmu.



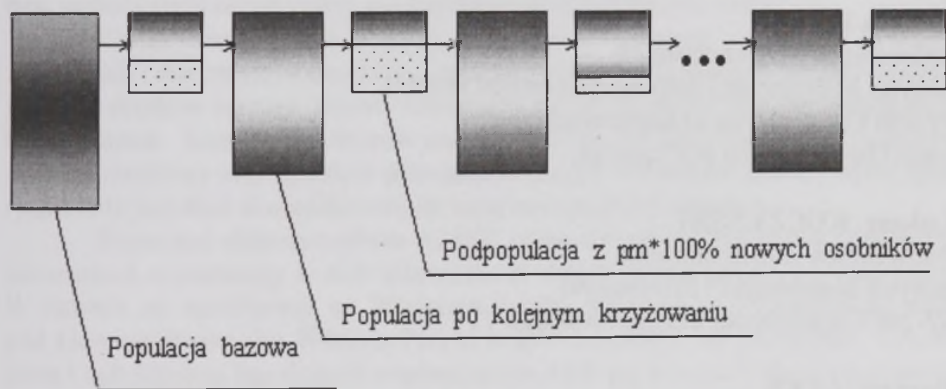


- Lpop - wielkość populacji
- i - numer osobnika w populacji
- Wej - numer węzła z wejściowym procesem
- Cis - s-ty gen i-tego chromosomu
- Vj - ilość dołączonych węzłów do węzła o numerze: = j
- T - wskaźnik zakończenia przesyłu procesu
- pd, pp - pomocnicze zmienne losowe
- njr - numer „mijanego” węzła (przez j – ty proces)

Rys. 4. Algorytm generowania populacji

#### 4. Procedury genetyczne i ocena przystosowania

W zestawie operacji genetycznych wykorzystano jedynie krzyżowanie, jako że inwersja genu tutaj reprezentowanego jako partycję opisującą przemieszczenie procesu z pewnością mogłaby naruszyć topologiczne powiązania danej sieci. Mutację zaś przeprowadzimy tak by nowe geny nie zmieniały topologii sieci co zrealizujemy poprzez wprowadzenie do nowej populacji nowych osobników w skali  $pm \cdot 100\%$ . Nie będzie to mutacja w klasycznym rozumieniu lecz jedynie „przyływ świeżej krwi”. Wracając do krzyżowania, to z prawdopodobieństwem pcr dla losowo wybranego punktu (lub punktów) krzyżowania obejmującego całe partycje przeprowadzamy zamianę części chromosomów [ ]. Przepływ informacji genetycznej ilustruje schemat na rys. 4.



Rys. 4. Schemat przesyłania informacji genetycznych

## **Wnioski :**

1. Zysk rozumiany jako skrócenie czasu przy zastosowaniu optymalizacji genetycznej jest tym większy im większa jest złożoność struktury sieciowej (zależność quasi-liniowa).
2. Włączenie do podpopulacji nowych osobników ma zapewnić szansę lokalizacji globalnego optimum nawet po znacznym stopniu zagłębienia w optimum lokalnym.
3. Funkcja przystosowania w dużym stopniu korzysta z bazy opisującej stan, charakterystyki i topologię sieci. Również klasyczne metody optymalizacji w podobnym stopniu korzystają z takiej bazy.

## **Literatura**

1. Goldberg Dawid: Algorytmy genetyczne w zastosowaniach. Warszawa, WNT, 1995.
2. Arabas Jarosław: Wykłady z algorytmów ewolucyjnych. Warszawa, WNT, 2001.
3. Michalewicz Zbigniew: Algorytmy genetyczne + struktury danych = programy danych. Warszawa, WNT, 1996.
4. Brdyś Mieczysław, Rusczyński Andrzej: Metody optymalizacji w zadaniach. Warszawa, WNT 1985.

### **Henryk PIECH,**

Politechnika Częstochowska  
Instytut Matematyki i Informatyki, tel. 604559300,  
42-200 Częstochowa ul. Dąbrowskiego 73  
e-mail: hpiech@adm.pcz.czest.pl,

### **Łukasz KUCZYŃSKI**

Politechnika Częstochowska  
Instytut Matematyki i Informatyki  
42-200 Częstochowa ul. Dąbrowskiego 73

### **Dariusz LEKS**

Politechnika Częstochowska  
Instytut Matematyki i Informatyki  
42-200 Częstochowa ul. Dąbrowskiego 73

# ZASTOSOWANIE XML W HETEROGENICZNEJ ROZPROSZONEJ BAZIE DANYCH WSPIERAJĄCEJ WIELKI EKSPERYMENT FIZYKI JĄDROWEJ

Tomasz TRACZYK

**Streszczenie:** Artykuł przedstawia przyczyny i okoliczności tworzenia rozproszonej bazy danych wspierającej konstrukcję detektorów dla wielkiego eksperymentu fizyki jądrowej ALICE. Zaprezentowano napotkane problemy i wybrane rozwiązania, ze szczególnym uwzględnieniem zastosowania XML.

## Wstęp

W Europejskim Centrum Fizyki Jądrowej CERN w Genewie jest obecnie budowany nowy akcelerator ciężkich jonów LHC (*Large Hadron Collider* [1]). Jednym z czterech głównych eksperymentów, które prowadzone będą na LHC jest ALICE (*A Large Ion Collider Experiment* [2]), mający na celu badanie właściwości materii w warunkach zderzeń jonów o wielkich energiach, w tym badanie przejścia fazowego do stanu plazmy kwarkowo-gluonowej [3].

LHC stanowi wielkie wyzwanie nie tylko z punktu widzenia fizyki, ale także inżynierii: sam tylko eksperyment ALICE jest jednym z największych obecnie realizowanych projektów badawczych, a skala całego przedsięwzięcia może być porównywana jedynie z programami badań kosmicznych. Dość wspomnieć, że LHC będzie największym urządzeniem zbudowanym w dziejach człowieka.

Eksperyment ALICE wymaga zbudowania ogromnej – zarówno co do złożoności jak i fizycznych rozmiarów – aparatury badawczej mającej na celu śledzenie i analizę skutków zderzeń jonów. Główną część tej aparatury stanowi zespół detektorów cząstek. Każdy z detektorów jest sam w sobie bardzo złożonym urządzeniem; niektóre detektory składają się z dziesiątków tysięcy elementów detekcyjnych, z których każdy jest dość skomplikowanym urządzeniem elektronicznym!

Prace nad eksperymentem ALICE prowadzi międzynarodowy zespół (*collaboration*), uczestniczy w nich kilkadziesiąt instytucji naukowych z całego świata. W ramach tej współpracy na Wydziale Fizyki Politechniki Warszawskiej działa pod kierownictwem dra Wiktora Peryta zespół zajmujący się projektem, wykonaniem i wdrożeniem baz danych wspierających budowę aparatury dla eksperymentu. Zespół składa się z pracowników, doktorantów i studentów kilku wydziałów Politechniki, a autor niniejszego opracowania pełni w nim rolę konsultanta w dziedzinie baz danych i opiekuna naukowego studentów.

## 1. Baza danych konstrukcji detektorów

Podstawowym zadaniem na etapie przygotowania eksperymentu jest zaprojektowanie i wykonanie detektorów. Jest to zadanie trudne ze względu na:



- unikalność sprzętu – detektory są projektowane specjalnie dla eksperymentu, a doświadczenia w dziedzinie projektowania i wykonania podobnej aparatury są bardzo ograniczone; zawsze zaś dotyczą znacznie mniejszej skali;
- niemożność przetestowania sprzętu w warunkach rzeczywistej eksploatacji;
- wielkie rozmiary sprzętu;
- dużą jego złożoność (detektor składać się może z dziesiątek tysięcy komponentów);
- konieczność osiągnięcia dużej precyzji działania sprzętu, by wyniki pomiarów były wiarygodne.

Do tego dochodzą problemy natury organizacyjnej, wiążące się z faktem, iż sprzęt jest projektowany i produkowany w wielu laboratoriach rozproszonych po całym świecie; trzeba więc zapewnić koordynację pracy i przekazywanie jej wyników.

Zadanie produkcyjne o takim rozmiarze jest właściwie niemożliwe do przeprowadzenia bez wsparcia informatycznego. Dlatego konstrukcję detektorów wspierać ma specjalna baza danych; to właśnie ona stanowi temat niniejszego opracowania.

### 1.1. Zadania bazy danych

Rolą bazy danych konstrukcji detektorów (*Detector Construction Database*) jest wspomaganie procesu produkcji detektorów oraz gromadzenie danych pozyskiwanych na etapie produkcji, a niezbędnych w czasie eksploatacji, np. do kalibracji pomiarów lub do dokonywania napraw i wymiany zużytych elementów.

Do najważniejszych zadań bazy danych należy zatem:

- śledzenie obiegu komponentów w czasie produkcji,
- śledzenie montażu komponentów składowych w komponentach złożonych oraz podziału komponentów na mniejsze elementy (np. podziału „wafli” krzemowych na pojedyncze *chipy*);
- gromadzenie danych opisujących produkowane komponenty;
- gromadzenie danych pomiarowych uzyskiwanych w czasie produkcji, w szczególności wyników różnorodnych testów.

### 1.2. Rozproszenie

Podstawowe problemy, warunkujące przyjęte rozwiązania techniczne, wynikają z faktu, iż produkcja detektorów odbywa się w wielu laboratoriach rozproszonych po całym świecie; dysponują one bardzo zróżnicowanymi możliwościami w dziedzinie informatyki.

W szczególności niektóre z nich:

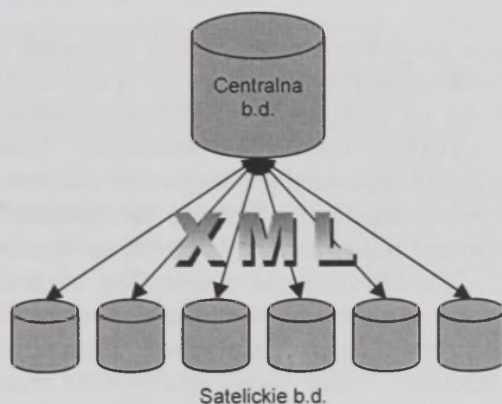
- nie mają dobrego dostępu do wydajnych łącz;
- nie zatrudniają profesjonalnych administratorów baz danych;
- nie dysponują profesjonalnymi systemami zarządzania bazami danych ani nie mają możliwości ich nabycia.

Ze względu na pierwsze z tych ograniczeń nie jest możliwe przyjęcie koncepcji scentralizowanej bazy danych, choć centralizacja niewątpliwie pozwoliłaby uniknąć wielu istotnych problemów.

Pozostałe ograniczenia skutkują niemożnością zastosowania komercyjnego systemu zarządzania bazami danych. Konieczne jest zastosowanie w laboratoriach systemu typu *open source*, który jest dostępny za darmo i łatwy do zainstalowania oraz administrowania.

### 1.3. Architektura rozproszonej bazy danych

Konieczność zainstalowania baz danych w laboratoriach warunkuje przyjęcie architektury rozproszonej. Ponieważ jednak dane tworzone w rozproszonym procesie produkcji będą w przyszłości potrzebne w czasie eksploatacji, a ta oczywiście będzie miała miejsce w CERN, postanowiono oprócz zbioru baz w laboratoriach (nazwanych bazami satelickimi) zbudować także bazę centralną, umieszczoną w CERN. Ogólną architekturę systemu przedstawia rysunek 1.



Rys. 1. Ogólna architektura rozproszonej bazy danych konstrukcji detektorów

Architektura taka ma szereg ważnych zalet:

- umożliwia sukcesywne gromadzenie danych w jednym miejscu i ich udostępnianie w miarę gromadzenia przez jeden centralnie zarządzany serwis WWW;
- pozwala uniknąć konieczności zarządzania „siecią” wymiany danych między bazami satelickimi – wszelkie przekazy informacji odbywają się za pośrednictwem bazy centralnej;
- ułatwia zadanie śledzenia obiegu komponentów – „spis inwentarza” przechowywany w bazie centralnej zawiera informacje o cyrkulacji każdego komponentu między laboratoriami;
- w sposób naturalny zapewnia kanały dystrybucji metadanych – słowników konfigurujących generyczną strukturę danych;

- zwiększa bezpieczeństwo – dane gromadzone w bazie centralnej stanowią swoistą kopię rezerwową danych z baz satelickich; jest to ważne wobec faktu, iż bazy satelickie nie będą zapewne administrowane w sposób profesjonalny; baza centralna będzie zaś odpowiednio administrowana i umieszczona na w pełni zabezpieczonych serwerach;
- po zakończeniu produkcji umożliwi likwidację baz satelickich i dalsze wykorzystywanie jedynie bazy centralnej.

Niestety, struktura ta ma także poważne wady; wynikają ona jednak z samego – niemożliwego z podanych wyżej przyczyn do uniknięcia – faktu rozproszenia bazy danych:

- potrzebę zorganizowania wymiany danych między bazami;
- konieczność zapewnienia zgodności danych między bazami satelickimi a bazą centralną;
- potrzebę synchronizowania możliwości modyfikowania kopii tych samych danych w różnych bazach.

#### 1.4. Heterogeniczność

Centralna baza danych gromadzić będzie wszystkie dane zbierane w bazach satelickich. Wolumen danych gromadzonych w tej bazie będzie zatem bardzo duży; należy spodziewać się, iż wystąpią tu problemy charakterystyczne dla bardzo wielkich baz danych (VLDB – *Very Large Databases*). Dlatego niezbędne są tu rozwiązania technologiczne zapewniające bezpieczne składowanie i efektywny dostęp do danych o wielkiej objętości. Systemy zarządzania bazami danych typu *open source* nie zapewniają odpowiednich środków, np. możliwości partycjonowania tabel. Co więcej, dane zgromadzone w centralnej bazie danych będą miały w przyszłości krytyczne znaczenie dla powodzenia eksperymentu, a ich utrata wiązałaby się z koniecznością powtórzenia części procesów produkcyjnych (np. niektóre złożone urządzenia trzeba by rozłożyć na części w celu przeprowadzenia pomiarów). Dlatego w przypadku centralnej bazy danych zastosowany być musi komercyjny system zarządzania bazami danych „z górnej półki”. Ze względu na zgodność z rozwiązaniami stosowanymi w innych systemach w CERN, zdecydowano użyć bazy danych Oracle 8i (z planowanym przejściem na wersję 9i).

W przypadku baz satelickich, ze względu na koszty nabycia i eksploatacji, zastosowanie rozwiązań komercyjnych nie było możliwe. Użyta technologia musi jednak zapewniać dość zaawansowane mechanizmy, w tym przetwarzanie transakcyjne oraz wyzwalacze – konieczne do ustanowienia złożonych reguł integralności (*business rules*) w projektowanej generycznej strukturze bazy danych. Po analizie dostępnych darmowych systemów zarządzania bazami danych wybrano system PostgreSQL.



## 1.5. Struktury generyczne

Jedną z podstawowych trudności w konstrukcji omawianego systemu informacyjnego polega na dużym zróżnicowaniu struktur danych opisujących poszczególne komponenty i testy.

Pierwsze, dość naiwne, próby stworzenia baz danych wspierających proces produkcji detektorów, podejmowano w samych laboratoriach. Powstało w ten sposób kilka zupełnie różnych i dość dziwacznych tworów. W kilku laboratoriach próbowano do zapisu danych stosować arkusze kalkulacyjne, co dość szybko okazało się pomysłem niefortunnym, gdy ilość danych stała się duża, a potrzebna struktura – bardziej złożona od pojedynczej tabelki. Były też próby stworzenia specjalizowanych struktur relacyjnych, także prowadzące do niepokojących rezultatów, np. dla jednego z detektorów utworzono kilkaset tabel, a niektóre z nich miały ponad tysiąc (!) kolumn.

Takie specjalizowane dla poszczególnych detektorów rozwiązania, nawet gdyby wykonane były poprawnie, i tak miałyby poważne wady:

- utrzymanie kilkunastu baz o zupełnie różnej strukturze wymagałoby bardzo wiele pracy i zdecydowanie przerastałoby możliwości niewielkiego zespołu;
- do każdej z wyspecjalizowanych struktur danych trzeba byłoby stworzyć osobną aplikację, a następnie je niezależnie utrzymywać; to także wymaga bardzo potężnych sił, zdecydowanie przewyższających te będące do dyspozycji;
- scalenie danych w jedną bazę centralną byłoby bardzo trudne.

Koncepcję tworzenia niezależnych struktur dla każdej z baz satelickich należało zatem odrzucić. Zdecydowano się na stworzenie jednej, uniwersalnej generycznej struktury danych, stosowanej we wszystkich bazach satelickich i dla wszystkich typów detektorów oraz komponentów.

W strukturze tej, przedstawionej na rysunku 2, znaczną część stanowią słowniki metadanych (oznaczone na diagramie kolorem szarym), które definiują typy detektorów i komponentów oraz opisujące je dane i struktury wyników testów.

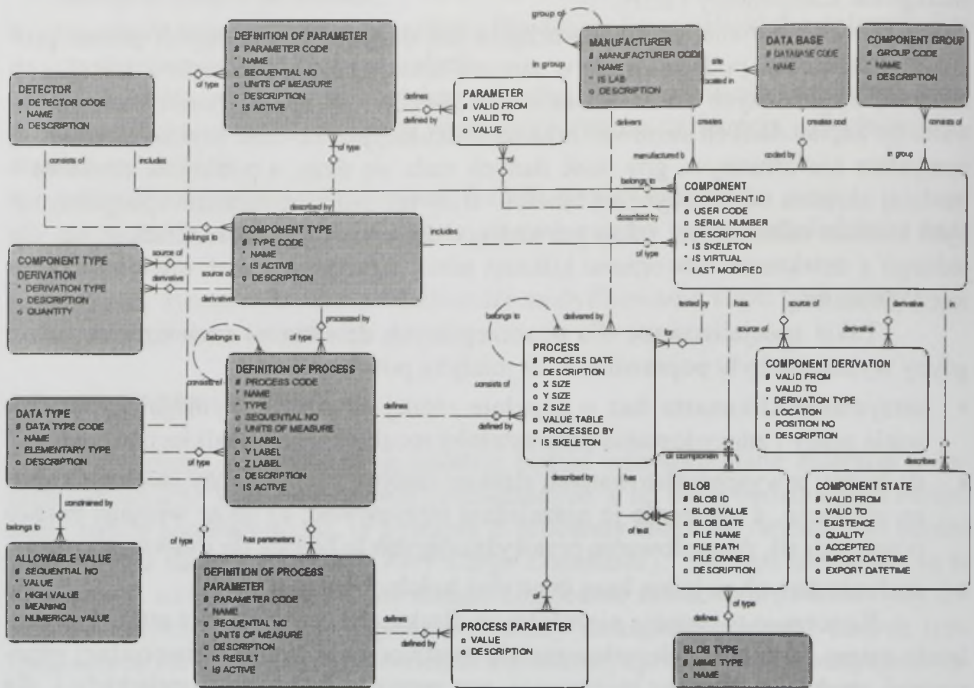
Stosując ujednoliconą strukturę generyczną uzyskano ważne korzyści:

- radykalnie ułatwiono zarządzanie rozwojem systemu;
- umożliwiono stworzenie jednej uniwersalnej aplikacji, sterowanej metadany;
- znacznie ułatwiono gromadzenie danych w bazie centralnej, struktura bazy centralnej może bowiem być bardzo podobna do struktury baz satelickich.

Ze stosowaniem struktur generycznych wiążą się oczywiście także pewne problemy:

- struktury te zwykle prowadzą do trudniejszych zapytań, a przez to do gorszej wydajności systemu; w tym przypadku jednak przed podjęciem ostatecznej decyzji o zastosowaniu struktur generycznych przeprowadzono intensywne testy na realistycznie dużych danych próbnych;

- założenie jednakowej struktury wszystkich baz satelickich prowadzi do konieczności równoczesnego wprowadzania poprawek do nich wszystkich; jednak jest to i tak zadanie łatwiejsze niż zapanowanie nad poprawkami wprowadzonymi niezależnie do kilkunastu różnych struktur.



Rys. 2. Diagram ER generycznej struktury danych baz satelickich (fragment)

Ze względów wydajnościowych zdecydowano, by do przechowywania wyników pomiarów zastosować struktury relacyjno-objektowe: atrybut VALUE TABLE w encji PROCESS jest zagnieżdżoną tablicą.

## 1.6. Aplikacje i API

Zarówno dla baz satelickich jak dla bazy centralnej stworzyć trzeba aplikacje umożliwiające wprowadzanie, a przede wszystkim wygodne wyszukiwanie, przeglądanie i zestawianie danych.

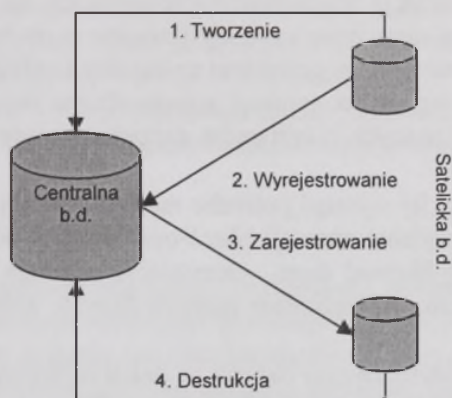
Wszystkie aplikacje w systemie są udostępniane wyłącznie przez WWW. Do stworzenia obecnie funkcjonujących wersji użyto języka PHP, w dalszym rozwoju systemu planowane jest jednak przejście na inne technologie, przede wszystkim JSP. Część aplikacji przeznaczona jedynie dla bazy centralnej (np. edycja słowników metadanych) stworzona została za pomocą *Oracle Web Toolkit* w języku PL/SQL.

Oprócz aplikacji interaktywnych niezbędne jest także stworzenie odpowiedniego interfejsu programistycznego (API) do bazy centralnej, który umożliwi

korzystanie z danych bezpośrednio w programach obliczeniowych. API musi być przystosowane do współpracy ze specjalizowanym, opracowanym w CERN systemem obliczeniowo-prezentacyjnym AliRoot [6], a także do użycia w strukturze obliczeń rozproszonych opartych o infrastrukturę DataGrid [7].

## 2. Śledzenie obiegu komponentów i danych

Jednym z najważniejszych zadań konstrukcyjnej bazy danych jest śledzenie obiegu komponentów. Baza powinna przechowywać informacje o tym w którym laboratorium znajduje się każdy komponent, a także historię jego obiegu. Zadanie komplikuje fakt, że w trakcie produkcji komponenty mogą być montowane do komponentów złożonych i przez to tracić samodzielność; mogą także być dzielone na mniejsze części.



Rys. 3. Uproszczony schemat śledzenia obiegu komponentów

Przeptyw informacji o położeniu komponentu ilustruje rysunek 3. Informacje o obiegu komponentów przechowywane będą w repozytorium w bazie centralnej. Ponieważ jednak awaria łącza do bazy centralnej nie może powodować wstrzymania produkcji, zarejestrowanie nowego komponentu odbywa się w bazie satelickiej; następnie, gdy komponent opuszcza dane laboratorium, zostaje wyrejestrowany, a gdy przybędzie do następnego laboratorium, zostaje zarejestrowany. Gdy komponent przestaje istnieć na skutek fizycznego zniszczenia lub zamontowania w innym komponencie, fakt ten jest rejestrowany w bazie satelickiej laboratorium, w którym nastąpiło zniszczenie komponentu, a następnie informacja o tym jest przekazywana do bazy centralnej.

## 3. Obieg informacji w rozproszonej bazie danych

Niewątpliwie najtrudniejszym zadaniem, które stanęło przed twórcami omawianego systemu informacyjnego, jest zorganizowanie obiegu informacji mię-



dzy bazami danych. Poczyniono tu następujące ważne założenia, nieco upraszczające problem:

- nie jest możliwa wymiana danych bezpośrednio między bazami satelickimi, konieczne jest pośrednictwo bazy centralnej;
- słowniki metadanych są konfigurowane centralnie i dystrybuowane do wszystkich baz satelickich; w bazach tych nie można metadanych zmieniać;
- wymiana danych dotyczących komponentów odbywać się będzie zawsze „całymi komponentami”, tj. nie jest możliwe przesłanie cząstkowych danych dotyczących komponentu.

### 3.1. *Check-in i check-out*

Ważnym problemem, który trzeba rozwiązać w tego typu strukturze rozproszonej, jest synchronizacja wprowadzania modyfikacji do danych. Gdyby bowiem zdarzyło się, że te same dane są modyfikowane w dwóch bazach równocześnie, to podczas transferu do bazy centralnej nastąpiłby konflikt.

Aby zapobiec tego typu sytuacji wprowadzono mechanizmy *check-in* i *check-out*, podobne do znanych z systemów zarządzania wersjami oprogramowania.

- Gdy w bazie satelickiej wystąpi potrzeba modyfikowania danych, zgłasza ona do bazy centralnej żądanie operacji *check-out* i dopiero po pomyślnym jej wykonaniu może modyfikować dane. Oczywiście operacja ta musi być poprzedzona transferem lub odświeżeniem samych danych, które mają być modyfikowane.
- Po zakończeniu modyfikowania danych są one transferowane do bazy centralnej. Jeśli baza satelicka nie jest zainteresowana dalszym utrzymywaniem „wyłączności” prawa modyfikacji danych, to wykonuje operację *check-in*, tym samym umożliwiając innym bazom wykonanie *check-out* i modyfikację danych. Operacje *check-in* i *check-out*, podobnie jak cała wymiana danych, dotyczą zawsze kompletnych danych komponentu.

### 3.2. **Dodatkowe trudności**

Konstruując system obiegu informacji trzeba było wziąć pod uwagę ostre wymagania użytkowników, które dodatkowo komplikują problem.

Potrzebna jest mianowicie możliwość pobrania danych z bazy satelickiej do bazy w komputerze przenośnym. Oznacza to, że obieg fizycznych komponentów i dotyczących ich danych może się różnić. Komponent pozostaje w laboratorium, gdy dane obrabiane są w komputerze przenośnym, poza bazą satelicką związaną z tym laboratorium. Problem rozwiązano w sposób następujący.

- Baza na komputerze przenośnym jest traktowana równorzędnie z bazami satelickimi. Aby więc przenieść do niej dane, trzeba je najpierw skopiować do bazy centralnej, a następnie dopiero z niej pobrać dane do bazy na komputerze przenośnym.

- Aby nie dopuścić do całkowitego zerwania związku między danymi komponentu a samym komponentem założono, iż w normalnym trybie operację *check-out* może wykonać tylko baza danych związana z laboratorium posiadającym fizycznie komponent. Jeśli potrzebne jest pobranie danych na komputer przenośny, baza „posiadająca” komponent musi nie tylko wykonać operację *check-in*, ale także wyrazić jawnie zgodę na wykonanie *check-out* przez inną bazę.

Niezbędna jest także możliwość pracy z danymi komponentów w sytuacji przejściowego braku łączności z bazą centralną. Oznacza to, że nawet jeśli nie istnieje możliwość pobrania znacznika *check-out* na dane komponentu, może być konieczne udostępnienie tych danych do edycji. W takiej sytuacji – nazwanej trybem wymuszonym – dane mogą być jednocześnie modyfikowane w więcej niż jednej bazie; przy zapisywaniu ich do bazy centralnej konieczne jest więc dokonanie rozstrzygnięcia ewentualnych konfliktów.

Kolejna trudność nie wynika z wymagań użytkowników, ale z samego charakteru rozproszenia bazy. Otóż poszczególne serwery rozmieszczone będą niemal dookoła kuli ziemskiej (od Indii po USA), znajdować się zatem będą w różnych strefach czasowych. Dlatego zdecydowano, że dane zawierające datę i czas reprezentowane będą w formie uwzględniającej strefy czasowe.

#### 4. Zastosowanie XML

Heterogeniczność przyjętego rozwiązania powoduje, że nie jest możliwe skorzystanie z gotowych rozwiązań dla rozproszonych baz danych, np. z wbudowanych w niektóre systemy możliwości replikacji danych. Konieczne jest zatem stworzenie specjalizowanych rozwiązań do wymiany danych. XML stanowi bardzo dobry środek do elektronicznej wymiany danych, przede wszystkim ze względu na łatwość tworzenia specjalizowanych dialektów oraz szeroką dostępność narzędzi programistycznych (por. także [9]). W naszym projekcie postanowiono zastosować język XML do wymiany informacji między bazami w heterogenicznym rozproszonym systemie informacyjnym. XML okazał się też bardzo przydatny w tworzeniu pomocniczych systemów obiegu dokumentacji.

##### 4.1. XML w wymianie danych

Specjalnie zaprojektowane dokumenty („komunikaty”) w języku XML użyte będą do wymiany informacji między bazą centralną a bazami satelickimi:

- do dystrybucji metadanych z bazy centralnej do baz satelickich;
- do przekazywania informacji związanych z operacjami *check-in* i *check-out* oraz rejestrowania i wyrejestrowywania komponentu;
- do transferu danych komponentów między bazami.

Konieczne okazało się zastosowanie dwóch trybów transferu:

- *on-line* – potrzebny do przeprowadzania operacji *check-in*, *check-out* oraz rejestrowania i wyrejestrowywania komponentu; ponieważ operacje te wiążą się z

zapisem danych kontrolnych w obu współdziałających bazach, niezbędne jest wprowadzenie mechanizmu zatwierdzania transakcji rozproszonych typu *two phase commit*;

- *off-line* – używany do transferu danych.

## 4.2. XML – technologia

Zdecydowano, by w projektowanych komunikatach XML używać przestrzeni nazw oraz schematów XML. Dzięki temu podstawowe sprawdzenie poprawności przekazywanych danych wykonywane jest przez standardowe narzędzia (parsery XML) i nie wymaga dodatkowej pracy.

Do tworzenia oraz analizy przesyłanych dokumentów XML użyto pakietu *Oracle XML Developers' Toolkit* (XDK), napisanego w języku Java i dobrze współdziałającego ze stosowanymi w projekcie bazami danych i serwerami HTTP. W szczególności używa się technologii XSU, służącej do generowania dokumentów XML na podstawie zapytań SQL do relacyjnej bazy danych (patrz [10]). Do formatowania i transformacji dokumentów XML stosuje się język XSLT. Analizę leksykalną przesyłanych dokumentów XML wykonuje się z użyciem parserów wchodzących w skład XDK.

## 5. Systemy pomocnicze

Ponieważ powstający system jest dość złożony, konieczne stało się opracowanie pomocniczych systemów wspomagających prace projektowe i administracyjne. Jeden z tych systemów, wspomagający obieg dokumentacji, opiera się na technologiach związanych z XML.

Powstająca rozproszona baza danych wymaga dokumentowania, a zadania zarządzania nią (np. dystrybucji kolejnych wersji) powinny być jakoś nadzorowane. Do tego celu tworzony jest system obiegu dokumentacji, o następujących zadaniach:

- sformalizowany zapis i prezentacja w WWW wymagań użytkownika (*user requirements*) i wymagań co do oprogramowania (*software requirements*);
- rejestracja zadań projektowych i administracyjnych; przypisanie ich do wymagań oraz przydzielenie konkretnym osobom do wykonania;
- śledzenie wykonania zadań;
- tworzenie listy często zadawanych pytań wraz z odpowiedziami itp.

System ten, jak widać, nie obejmuje typowej dokumentacji technicznej samej bazy danych (ta dokumentacja jest tworzona za pomocą narzędzia CASE) ani dokumentacji użytkowej aplikacji.

Przyjęto następujące dodatkowe założenia:

- informacje do systemu powinny być wprowadzane przez uprawnionych użytkowników za pomocą interfejsu bazującego na WWW;
- informacje te następnie powinny być dystrybuowane w postaci stron WWW, w formie zdatnej do dalszego przetwarzania.



Wyjściowe informacje powinny mieć dwójaki charakter:

- roboczych stron WWW informujących o zadaniach i postępie w ich realizacji;
- oficjalnych dokumentów zgodnych z wytycznymi ESA [11], o sformalizowanej i sztywnej strukturze, w których znaczna część zawartości jest stała albo bardzo rzadko zmieniana.

Takie wymagania co do postaci wyjściowej skłaniają do przyjęcia następującego rozwiązania:

- informacje stosunkowo często zmieniane (np. zadania, przydziały itp.) przechowywane będą w bazie danych, w specjalnie zaprojektowanej strukturze;
- informacje stałe lub rzadko zmieniane będą reprezentowane z postaci plików XML.

Uprawnieni użytkownicy będą mieli dostęp do zawartości bazy danych przez interfejs WWW (pierwsza wersja powstanie w technologii *Oracle Web Toolkit* w języku PL/SQL, wersja docelowa zapewne w technologii JSP).

W przypadku stron roboczych całość informacji może być tworzona na podstawie zawartości bazy danych przy pomocy standardowych narzędzi. Tworzone dynamicznie dokumenty XML będą zaopatrzone w arkusze stylistyczne w XSL, będzie także dostępne opcjonalne przekształcanie ich do formatu HTML po stronie serwera.

W przypadku oficjalnych dokumentów części wolnozmiennie składowane w postaci plików XML będą łączone z częściami dynamicznie generowanymi na podstawie zawartości bazy danych. Proces ten będzie sterowany odpowiednim skryptem w języku XSLT.

To stworzenia opisywanych tu stron WWW zastosowana zapewne zostanie technologia *XSQL Server Pages* (patrz [10]).

## 6. Podsumowanie

Tworzenie bazy danych konstrukcji detektorów dla eksperymentu ALICE jest zadaniem ekscytującym, gdyż pozwala na udział w jednym z ambitniejszych współczesnych zamierzeń badawczych. Jednocześnie jednak, ze względu na nietypowe uwarunkowania, jest to zadanie dość trudne.

Uwarunkowania te wymuszają budowę systemu rozproszonego i heterogenicznego i zorganizowanie w nim złożonej wymiany informacji. Dzięki zastosowaniu do owej wymiany języka XML można było wykorzystać gotowe narzędzia programistyczne, dobrze współpracujące z użytymi bazami danych.

Omawiana praca jest obecnie zaawansowana, ale jeszcze dość daleka od ukończenia. Projekt baz satelickich jest ukończony i obecnie trwają pierwsze wdrożenia w laboratoriach. Projekt bazy centralnej jest na ukończeniu. Trwają intensywne prace nad systemem wymiany danych i systemem obiegu dokumentacji.

Mimo iż praca nie jest jeszcze ukończona, zgromadzone dotychczas doświadczenia pozwalają stwierdzić, że język XML bardzo dobrze nadaje się do wymiany danych w heterogenicznym systemie rozproszonym. Zastosowanie schema-

tów, języka XSLT oraz gotowych standardowych narzędzi programistycznych pozwala znacznie zredukować nakłady pracy niezbędnej do wykonania zadania.

## Literatura

1. <http://www.cern.ch/lhc>
2. <http://www.cern.ch/alice>
3. Peryt W.: Informacja/oferta współpracy. Materiał wewnętrzny, Wydział Fizyki Politechniki Warszawskiej, 2000.
4. Eksperyment ALICE. Miesięcznik Politechniki Warszawskiej, nr 11, 2000.
5. <http://det-dbalice.if.pw.edu.pl>
6. <http://alisoft.cern.ch/offline/aliroot-pro/manual.html>
7. <http://eu-datagrid.web.cern.ch/eu-datagrid>
8. Traczyk T.: XML i XSL. Materiały XII Górskiej Szkoły PTI, Szczyrk 2000.
9. Traczyk T.: XML – uniwersalny język wymiany informacji w e-świecie. Proceedings of the 3rd conference IBIS'01 (Implementation of Business Information Systems) Global Information Society. Malmoe–Kopenhaga, wrzesień 2001.
10. Traczyk T.: Czy już warto używać XML? Tutorial, Materiały I Seminarium PLOUG, Warszawa 2001.
11. Software Engineering Standards. ESA Board for Software Standardisation and Control, 1994.

dr inż. Tomasz Traczyk  
Politechnika Warszawska  
Instytut Automatyki i Informatyki Stosowanej  
ul. Nowowiejska 15/19, 00-665 Warszawa  
e-mail: [ttraczyk@ia.pw.edu.pl](mailto:ttraczyk@ia.pw.edu.pl)  
URL: <http://www.ia.pw.edu.pl/~ttraczyk>

# AUTOMATYCZNA PERSONALIZACJA SERWERÓW WWW Z WYKORZYSTANIEM METOD EKSPLOKACJI DANYCH

Marek WOJCIECHOWSKI, Maciej ZAKRZEWICZ

**Streszczenie:** Niniejszy artykuł poświęcony jest zagadnieniom automatycznej personalizacji serwisów WWW w oparciu o tzw. adaptatywne serwery WWW. Personalizacja serwisów WWW polega na wykorzystywaniu znanych profili preferencji do dynamicznego dostosowywania zawartości serwisu do potrzeb poszczególnych użytkowników. Adaptatywne serwery WWW automatycznie odkrywają typowe schematy zachowań użytkowników analizując informacje o użytkowaniu serwisu zawarte w logu serwera technikami eksploracji danych. W artykule przedstawiono ogólną ideę adaptatywnych serwerów WWW oraz szczegółowe propozycje ich implementacji. Artykuł poświęca również dużo miejsca technikom gromadzenia wiarygodnych informacji o użytkowaniu serwisów WWW oraz ich wstępnego przetwarzania do formatu odpowiedniego dla technik eksploracji danych.

## Wstęp

Personalizacja serwisów WWW (ang. *Web personalization*) polega na wykorzystywaniu odkrytych profili preferencji do dynamicznego dostosowywania zawartości serwisu do potrzeb poszczególnych użytkowników [10]. W chwili obecnej, personalizacja nabiera coraz większego znaczenia, ze względu na stale rosnącą liczbę konkurujących ze sobą serwisów WWW. Podstawową bronią w walce o klienta, zarówno w kontekście serwisów informacyjnych jak i sklepów internetowych, jest dostosowanie przedstawianej zawartości do jego potrzeb, oczekiwań i zainteresowań. Ze względu na oczywisty fakt, że różni użytkownicy mogą mieć różne oczekiwania i upodobania, serwisy przedstawiające tę samą zawartość wszystkim odwiedzającym, mogą tracić popularność na rzecz serwisów posiadających zdolność adaptacji do różnych i zmiennych w czasie preferencji użytkowników.

Narzucającym się rozwiązaniem, mogącym zapewnić użytkownikom dostosowany do ich potrzeb obraz danej witryny, jest umożliwienie im podjęcia decyzji jakie informacje i w jakiej formie mają im być przedstawiane. Rozwiązanie to jest stosowane w praktyce przez wiele serwisów i portali, np. *Yahoo!*. Jest ono określane raczej jako „możliwość dostosowywania” (ang. *customization*), a nie personalizacja, gdyż nie ma charakteru automatycznego i prawie całkowicie polega na użytkownikach. W przypadku wspomnianego serwisu *Yahoo!*, użytkownicy mają możliwość złożenia z setek dostępnych modułów swojej strony *My Yahoo!* [11]. Dostępne moduły prezentują aktualne informacje o pogodzie, sporcie, kursach akcji, itp. Niestety praktyka pokazuje, że przeważająca



większość użytkowników nie korzysta w ogóle z możliwości dostosowywania zawartości serwisu własnymi siłami. Z pewnością dzieje się tak nie tylko dlatego, że domyślne strony są tak skonstruowane aby odpowiadały preferencjom jak największej grupy użytkowników. Bardzo prawdopodobne jest, że dla większości użytkowników dostosowywanie zawartości odwiedzanych witryn jest zbyt czasochłonne, a nawet po prostu zbyt skomplikowane.

Celem automatycznej personalizacji serwisów WWW jest przerzucenie odpowiedzialności za dostosowywanie prezentowanej zawartości z użytkownika na serwer. W tym wypadku serwer stara się dopasować charakterystykę użytkownika do jednego ze znanych profili i następnie prezentuje użytkownikowi zawartość dostosowaną do dopasowanego profilu preferencji. Kluczowym elementem jest pozyskanie charakterystyk użytkownika takich jak wiek, płeć, zawód, adres zamieszkania, zainteresowania, itd. Najprostszym sposobem uzyskania takich informacji jest wymaganie od użytkownika jawnego ich podania poprzez wypełnienie formularza rejestracyjnego czy ankiety. Serwer po dopasowaniu użytkownika do jednego z profili preferencji dostosowuje zawartość przedstawianą użytkownikowi zgodnie z regułami związanymi z danym profilem. Reguły te mogą być ustalone a priori przez ekspertów (być może posługujących się narzędziami do analizy danych) lub modyfikowane dynamicznie poprzez obserwację zachowań użytkowników przypisanych do danego profilu. Drugie z wymienionych rozwiązań jest bardziej elastyczne – może uwzględnić np. rozbudowę witryny. Ponadto, jest rozwiązaniem bardziej praktycznym gdy zbiór produktów, dokumentów oferowanych użytkownikom w ramach danego serwisu jest bardzo liczny [6]. Rozwiązanie to polega na proponowaniu użytkownikom tego, co zainteresowało innych użytkowników o podobnych charakterystykach (ang. *collaborative filtering*).

Pozyskiwanie informacji o preferencjach użytkowników w oparciu o formularze rejestracyjne i ankiety ma jednak kilka istotnych wad. Po pierwsze, ta forma pozyskiwania wiedzy cechuje się dużą subiektywnością i spotyka się z niechęcią użytkowników, „zmuszanych” do wypełniania dodatkowych formularzy. Ponadto, tak budowane profile użytkowników posiadają charakter statyczny i z upływem czasu ulegają degradacji. Rozwiązania bazujące na jednym statycznym profilu preferencji użytkownika nie biorą również pod uwagę faktu, że ten sam użytkownik może w różnych momentach czasowych poszukiwać różnych informacji. Przykładowo, użytkownik internetowego biura podróży będzie w okresie zimowym zainteresowany dokumentami WWW zawierającymi informacje o kurortach narciarskich w Alpach, natomiast w okresie letnim, ten sam użytkownik życzyłby sobie prezentacji dokumentów WWW opisujących wczasy w basenie Morza Śródziemnego.

W związku z powyższym, w ostatnich latach coraz większą uwagę przyciągają metody personalizacji zawartości serwerów WWW poprzez niejawne obserwowanie trendów w zachowaniach użytkowników WWW. W pracy [14],

zaproponowano termin *adaptatywne serwery WWW* (ang. *adaptive web sites*), opisujący serwery WWW, które automatycznie ulepszają swoją zawartość i organizację na podstawie obserwacji ścieżek dostępu użytkowników. Idea wykorzystania koncepcji adaptatywnych serwerów do personalizacji serwisu polega na analizie plików logu serwera, wyławianiu z nich statystycznych korelacji pomiędzy pobieranymi dokumentami lub pracującymi użytkownikami, a następnie wykorzystywaniu znalezionych korelacji do modyfikacji struktury dokumentów WWW, wysyłanych użytkownikom. Do analizy logu, adaptatywne serwery WWW wykorzystują techniki eksploracji danych (ang. *data mining*). Eksploracja danych polega na automatycznym odkrywaniu nietrywialnych, interesujących, wcześniej nieznanych wzorców i zależności w dużych wolumenach danych. Eksploracja logów serwerów WWW (ang. *Web usage mining* [4]) jest jednym z podstawowych obszarów zastosowań technik eksploracji danych, posiadającym swoją specyfikę. W przypadku eksploracji danych o zachowaniach użytkowników WWW, szczególne znaczenie mają techniki gromadzenia danych źródłowych gwarantujące ich wiarygodność oraz algorytmy wstępnej transformacji i „czyszczenia” zgromadzonych danych źródłowych.

Niniejszy artykuł opisuje stan nauki i technologii w zakresie personalizacji serwisów WWW w oparciu o adaptatywne serwery WWW, wykorzystujące techniki eksploracji danych. Artykuł przedstawia ogólną ideę adaptatywnych serwerów WWW, prezentując różne propozycje ich implementacji, które pojawiły się dotychczas w literaturze.

## **1. Gromadzenie i wstępne przetwarzanie informacji źródłowych dla eksploracji danych**

Podstawowym źródłem danych dla wszelkich analiz zachowania użytkowników serwisu WWW jest plik logu (dziennik) serwera WWW. Najczęściej jest on jedynym źródłem informacji, choć serwisy działające w oparciu o serwery aplikacji (ang. *application servers*) mogą gromadzić również informacje o działaniach użytkowników specyficzne dla danego serwisu na poziomie aplikacji. Podstawowe problemy występujące w przypadku zaawansowanej analizy logu serwera WWW (np. na potrzeby automatycznej adaptacji serwera), to zapewnienie jak największej wiarygodności gromadzonych danych oraz transformacja danych zawartych w logu do formatu odpowiedniego dla konkretnej techniki eksploracji danych.

### **2.1 Struktura pliku logu**

Jak już wspomniano wcześniej, informacje o dostęпах do serwera WWW zapisywane są w pliku logu. Dla każdego dostępu do pojedynczego pliku znajdującego się na serwerze, w logu pojawia się nowy zapis. Jednakże ilość informacji pamiętana w związku z danym dostępem może być różna w przypadku

różnych serwerów WWW. Aby umożliwić tworzenie uniwersalnych narzędzi służących do analizy logu, pojawiły się próby standaryzacji jego formatu. Dzisiaj można założyć, że przeważająca większość serwerów WWW generuje pliki logu zgodne z formatem znanym pod nazwą *Common Logfile Format* [9]. Niektóre serwery pamiętają również pewne dodatkowe informacje (standard *XLF*). *Common Logfile Format* przewiduje, że zapis w logu powinien mieć następującą postać:

```
remotehost rfc931 authuser [date] "request" status bytes
```

W powyższym formacie pole *remotehost* oznacza nazwę lub adres IP komputera, z którego nastąpiło odwołanie. Pole *rfc931* zawiera nazwę użytkownika na danym komputerze (*ang. logname*). Pole *authuser* jest wypełnione gdy serwer przeprowadza autoryzację użytkowników przy dostępie do danego zasobu i zawiera w takim przypadku nazwę użytkownika podaną przy autoryzacji. Pole *[date]* informuje o tym kiedy nastąpiło odwołanie (data i czas). Pole *"request"* zawiera żądanie przesłane do serwera w takiej formie, w jakiej wygenerował je klient. Obejmuje ono na ogół typ operacji i nazwę pliku, do którego nastąpiło odwołanie, wraz ze ścieżką dostępu. Pole *status* zawiera zwracany klientowi kod statusu, zgodnie z protokołem HTTP wykorzystywanym w usłudze WWW. Długość zawartości przesyłanego dokumentu pamiętana jest w polu *bytes*. Przykład zawartości pliku logu serwera WWW przedstawiono na rysunku 1.

```
154.11.231.17 - - [13/Jul/2000:20:42:25 +0200] "GET / HTTP/1.1" 200
1673
154.11.231.17 - - [13/Jul/2000:20:42:25 +0200] "GET /apache_pb.gif
HTTP/1.1" 200 2326
192.168.1.25 - - [13/Jul/2000:20:42:25 +0200] "GET /demo.html
HTTP/1.1" 200 520
192.168.1.25 - - [13/Jul/2000:20:42:25 +0200] "GET /books.html
HTTP/1.1" 200 3402
160.81.77.20 - - [13/Jul/2000:20:42:25 +0200] "GET / HTTP/1.1" 200 1673
154.11.231.17 - - [13/Jul/2000:20:42:25 +0200] "GET /car.html HTTP/1.1"
200 2580
192.168.1.25 - - [13/Jul/2000:20:42:25 +0200] "GET /cdisk.html HTTP/1.1"
200 3856
10.111.62.101 - - [13/Jul/2000:20:42:25 +0200] "GET /new/demo.html
HTTP/1.1" 200 971
```

Rys. 1. Przykładowy plik logu serwera WWW



## 2.2 Czyszczenie plików logu

Z punktu widzenia eksploracji danych, mającej na celu odkrycie typowych schematów zachowań użytkowników WWW, dane źródłowe powinny mieć postać zbiorów lub sekwencji stron WWW (ang. *Web page*) pobieranych z serwera przez poszczególnych użytkowników. Zapisy w logu dotyczą jednak pojedynczych plików, a nie stron traktowanych jako obiekty złożone. W przypadku dostępu do strony zawierającej np. obrazy, dźwięki lub filmy, w logu znajdzie się zapis dotyczący głównego dokumentu HTML (najczęściej z rozszerzeniem `html` lub `htm`), ale także zapisy związane ze wszystkimi obiektami zagnieżdżonymi w stronie (obrazami, filmami, itp.). Na szczęście charakter pliku można w dużym stopniu wywnioskować z jego rozszerzenia. Przykładowe rozszerzenia nazw plików odpowiadające obiektom zagnieżdżanym w dokumentach to `jpg`, `jpeg`, `gif` dla obrazów, `au`, `wav` dla dźwięków, `avi`, `mov` dla filmów. Aby dane źródłowe do analiz zawierały tylko informacje o dostęпах do istotnych dokumentów, należy poddać plik logu serwera WWW procesowi filtracji, w wyniku którego ignorowane są zapisy dotyczące plików nie będących głównymi dokumentami odpowiadającymi stronom WWW.

Należy tu zwrócić uwagę, że filtracja logu może nie być zadaniem trywialnym, gdy poszczególne elementy stron WWW są generowane na serwerze dynamicznie przez programy CGI lub serwlety. Na szczęście najczęściej dynamicznie generowane są dokumenty HTML, a zagnieżdżone w nich obiekty multimedialne pobierane są z systemu plików. Wtedy można założyć, że odwołania do programów czy serwletów są odwołaniami do stron WWW, tak jak odwołania do statycznych dokumentów HTML. Tak samo traktowane są dokumenty tworzone w oparciu o różnego rodzaju technologie dynamicznej generacji stron, takie jak np. ASP, PHP, JSP, posiadające swoje specyficzne rozszerzenia plików, oraz dokumenty XML.

## 2.3 Identyfikacja dostępow poszczególnych użytkowników

Z punktu widzenia analizy zachowań użytkowników istotnymi informacjami w logu serwera WWW są: nazwa lub adres IP komputera, z którego nastąpiło odwołanie, nazwa użytkownika dokonującego odwołania, dokładna data i czas oraz pełna nazwa pliku, którego dotyczyło żądanie. W oparciu o te informacje plik logu jest transformowany do postaci zbiorów lub sekwencji dostępow poszczególnych użytkowników do stron WWW. Operacja ta polega na wyodrębnieniu z logu informacji o dostęпах poszczególnych użytkowników. Jest to realizowane na podstawie adresu IP lub nazwy komputera oraz nazwy użytkownika. Niestety nie zawsze nazwa użytkownika jest znana. Sytuacja taka ma miejsce często w przypadku gdy użytkownik korzysta z systemu operacyjnego, który nie zakłada wielodostępu. Na szczęście fakt, że z komputera pracującego pod kontrolą systemu operacyjnego, który nie jest wielodostępny, może w danej chwili

korzystać tylko jeden użytkownik, pozwala traktować odwołania pochodzące z tego samego komputera jako odwołania jednego użytkownika, gdy nazwa użytkownika nie jest znana. Oczywiście powyższe założenie jest poprawne tylko w przypadku odwołań, których czasy zawierają się w okresie odpowiadającym możliwemu czasowi trwania pojedynczej sesji użytkownika. Mechanizm ten nie pozwala więc na identyfikację sekwencji dostępu w ramach wielu sesji użytkownika na przestrzeni np. miesiąca, gdyż z danego komputera może w różnych godzinach korzystać wiele osób.

Ze względu na fakt, że użytkownik może wielokrotnie korzystać z usług danego serwera WWW za każdym razem szukając innych informacji, niekiedy wskazane jest rozbicie sekwencji dostępu danego użytkownika na fragmenty odpowiadające poszczególnym sesjom. Najprostsze rozwiązanie tego problemu polega na wyodrębnianiu sesji użytkowników w oparciu o założenie, że jeśli czas między kolejnymi odwołaniami do serwera jest znacznie dłuższy niż typowy czas przeglądania jednej strony, to odwołania te nastąpiły w ramach dwóch różnych sesji. Alternatywnym rozwiązaniem może być rozszerzenie funkcjonalności serwera o obsługę identyfikatorów sesji na czas zbierania informacji o zachowaniach użytkowników [16].

Typowy profil zachowania użytkowników może być rozumiany jako grupa stron lub grupa ścieżek nawigacyjnych, często powtarzająca się w poszczególnych sesjach. W przypadku śledzenia ścieżek nawigacji istotne są informacje o wszystkich stronach, do których odwoływał się dany użytkownik z uwzględnieniem kolejności odwołań. Alternatywą jest ograniczenie się tylko do tych stron, których treść zainteresowała użytkownika (strony służące jedynie jako ścieżka dostępu do szukanego dokumentu nie są uwzględniane). W [3] zaproponowano podział odwołań do stron na zorientowane na zawartość i zorientowane na nawigację. Niektóre strony zawierają głównie odnośniki do innych stron, w związku z czym odwołania do nich na pewno będą miały charakter nawigacyjny. Jednakże wiele stron zawiera zarówno treść jak i odnośniki do innych stron. Takie strony mogą różnym użytkownikom służyć do różnych celów. Dlatego rozsądnym kryterium podziału dostępu na zorientowane na nawigację i zawartość wydaje się czas, na jaki użytkownik zatrzymuje się na danej stronie (być może znormalizowany w stosunku do rozmiaru strony). Czas przeglądania danej strony jest obliczany jako różnica etykiet czasowych dwóch kolejnych zapisów w logu (odpowiadających następnej i bieżącej stronie). W przypadku stron kończących sesję użytkownika przyjmuje się, że dostęp do nich miał miejsce ze względu na ich zawartość, choć oczywiście w konkretnym przypadku wcale nie musi to być prawdą.

## **2.4 Wiarygodność informacji zawartych w logu serwera WWW**

Dokonując jakichkolwiek analiz plików logu serwera WWW należy zdawać sobie sprawę z niedoskonałości mechanizmu odnotowywania przez serwer



odwołań użytkowników do dokumentów. Informacje zawarte w logu mogą być nie tylko niepełne, ale również zafałszowane ze względu na wykorzystywanie serwerów proxy i podręcznej pamięci przeglądarek [13]. Serwer proxy służy jako „okno na świat” dla wielu komputerów, pozwalając uzyskać dostęp do Internetu użytkownikom na nich pracującym. Zapisy w logu serwera WWW odpowiadające odwołaniom użytkowników komputerów „ukrytych” za serwerem proxy są opisane adresem serwera proxy. W związku z tym fakt, że kilka zapisów w logu dotyczy jednego adresu IP, nie musi wcale oznaczać, iż zapisy te odpowiadają odwołaniom z tego samego komputera. W [15] zaproponowano metodę wykrywania takich sytuacji w oparciu o założenie, że jeśli dane odwołanie dotyczy dokumentu, do którego nie ma łącza w poprzednio żądanym dokumencie, to prawdopodobnie żądania są kierowane przez dwóch różnych użytkowników. Mimo że doświadczenia pokazują [5], iż dostęp do kolejnego dokumentu jest najczęściej wynikiem wybrania dostępnego w dokumencie łącza (*ang. hyperlink*) lub powrotem do poprzedniego dokumentu (operacja „Back”), wspomniana metoda nie gwarantuje żadnej pewności. Dlatego dla celów identyfikacji użytkowników stosuje się tzw. *cookies* lub dodatkową autoryzację. Cookie jest identyfikatorem generowanym przez serwer i przesyłanym do klienta (przeglądarki) w celu późniejszej identyfikacji użytkownika. Niedoskonałość tego mechanizmu wynika z faktu, że użytkownicy mogą w dowolnej chwili usunąć cookie lub w ogóle zabronić akceptacji cookies. Dodatkowa identyfikacja użytkowników poprzez żądanie wypełnienia formatki rejestracyjnej również wymaga dobrej woli użytkowników, gdyż dane przez nich podawane mogą być przecież fałszywe.

Równie istotnym problemem jak identyfikacja użytkowników jest identyfikacja faktycznych odwołań do dokumentów. Ze względu na stosowanie przez przeglądarki pamięci podręcznej, kolejne odwołania danego użytkownika do tego samego dokumentu mogą nie być odnotowane na serwerze, gdyż mogą być zrealizowane przez sprowadzenie dokumentu z pamięci podręcznej przeglądarki a nie z serwera. W sposób znaczący może to zakłócić odkryte ścieżki nawigacji użytkowników. Jeszcze poważniejszy problem wynika ze stosowania pamięci podręcznej przez serwery proxy. Jeśli użytkownik, korzystający z Internetu poprzez serwer proxy, odwołuje się do dokumentu znajdującego się w pamięci podręcznej proxy, serwer WWW może być w ogóle nieświadomy, że dany użytkownik odwoływał się do danego dokumentu. Aby obronić się przed wspomnianymi sytuacjami serwery WWW mogą stosować techniki zapobiegające wykorzystywaniu pamięci podręcznej określane jako *cache-busting*, polegające np. na podawaniu dat z przeszłości jako terminów upłynięcia ważności poszczególnych dokumentów. Tego typu techniki mogą być uciążliwe dla użytkowników, gdyż wydłużają czas odpowiedzi. Z tego względu pojawiły się propozycje, aby zamiast monitorowania wszystkich dostępu do serwera, ograniczyć się tylko do pewnej próbki statystycznej i na jej bazie dokonywać analiz.



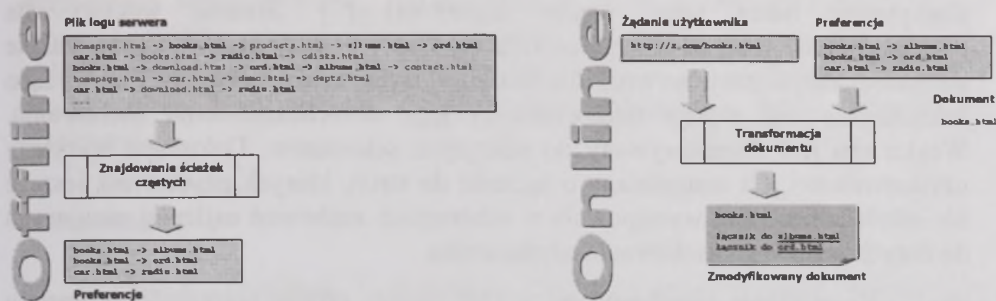
## 2. Automatyczna adaptacja serwera WWW w oparciu o wyniki eksploracji danych

Proces adaptacji serwera WWW przebiega w dwóch fazach:

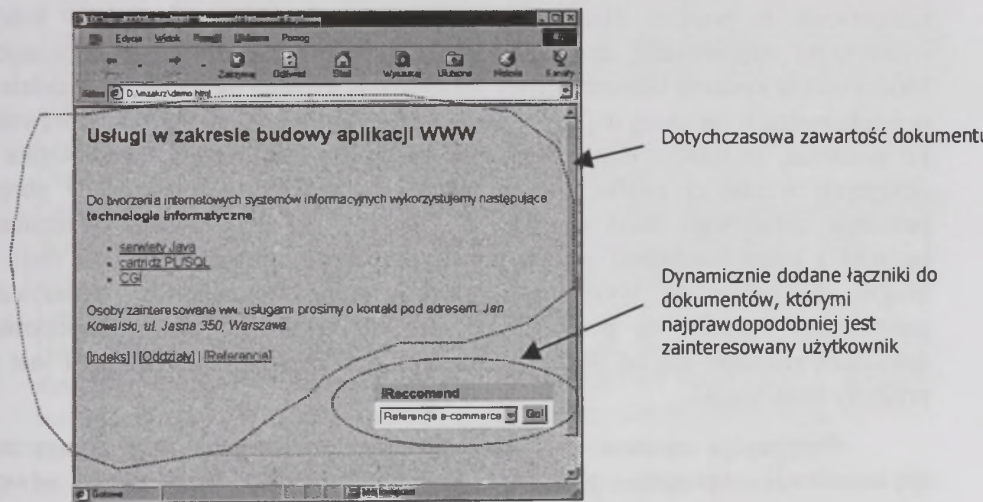
1. Offline: wykorzystanie pliku logu serwera do odkrycia typowych profili zachowań użytkowników reprezentowanych przez zbiory lub sekwencje stron WWW. Faza ta realizowana jest asynchronicznie względem połączeń użytkowników, np. w odstępach tygodniowych lub miesięcznych. W fazie tej stosowane są różne techniki eksploracji danych, po uprzedniej transformacji i oczyszczeniu pliku logu. Podstawowe znaczenie ma grupowanie (ang. *clustering*) [7], polegające na podziale zbioru obiektów na grupy w taki sposób, aby obiekty wewnątrz każdej z grup były maksymalnie podobne do siebie, a jednocześnie możliwie jak najbardziej różniące się od obiektów przydzielonych do innych grup. W kontekście adaptatywnych serwerów WWW, grupowanymi obiektami są sekwencje lub zbiory stron reprezentujące poszczególne sesje użytkowników. Do grupowania wybierane są takie algorytmy, które oprócz podziału na grupy dostarczają opis poszczególnych grup, w tym wypadku w postaci zbioru stron WWW lub ścieżek nawigacyjnych typowych dla danej grupy. Proces grupowania może być poprzedzony fazą odkrywania asocjacji [1] lub wzorców sekwencyjnych [2], jeśli stosowany algorytm grupowania tego wymaga.
2. Online: wykorzystywanie znalezionych grup stron lub ścieżek nawigacyjnych do tworzenia *dynamicznych rekomendacji* dla użytkowników, czyli zbioru łączników do dokumentów, którymi ci użytkownicy będą najprawdopodobniej (statystycznie) zainteresowani. Faza ta jest realizowana podczas obsługi każdego żądania użytkownika. Od chwili pierwszego podłączenia się użytkownika do serwera WWW, wszystkie operacje tego użytkownika są rejestrowane w formie tzw. *historii sesji*. Za każdym razem, kiedy użytkownik żąda przesłania dokumentu, historia jego sesji jest dopasowywana do odkrytych profili zachowań i wybierane są te profile, które wykazują się największym dopasowaniem. Zbiór łączników do dokumentów opisujących dopasowane profile staje się dodatkowym elementem wizualnym, który dynamicznie jest dołączany do żądanego dokumentu [16].

Przedstawmy przykład prostej adaptacji serwera WWW, zilustrowany na rysunku 2. Serwer WWW został odwiedzony przez pięciu użytkowników, których pełne ścieżki nawigacyjne zostały zapisane w pliku logu. W pierwszej fazie adaptacji (offline) wykonywana jest analiza pliku logu i znalezione zostają następujące ścieżki częste: books.html -> albums.html, books.html -> ord.html, car.html -> radio.html. Każda z tych ścieżek pojawiła się w 40% odwiedziny opisanych w pliku logu i w związku z tym będą one traktowane przez nas jako preferencje dla innych użytkowników. W drugiej fazie (online), nowy użytkownik wysyła do serwera żądanie przesłania dokumentu WWW (books.html). Serwer

pobiera dokument z dysku i przegląda znaleziony wcześniej zbiór ścieżek częstych – wynika z niego, że użytkownicy, którzy pobierali dokument books.html, byli później zainteresowani dokumentami albums.html i ord.html. W związku z tym, w celu ułatwienia nawigacji, do dokumentu books.html dynamicznie dodawane są łączniki do powyższych dokumentów. Tak zmodyfikowany dokument trafia do użytkownika (rysunek 3).



Rys. 2. Przykładowy proces adaptacji serwera WWW



Rys. 3. Przykład dokumentu WWW wzbogaconego o dynamicznie wygenerowane rekomendacje

W ramach ogólnej koncepcji adaptatywnych serwerów WWW zaproponowano w literaturze kilka sposobów ich implementacji, różniących się stosowanymi algorytmami grupowania, formą reprezentacji odkrytych profili

zachowań użytkowników oraz sposobem wykorzystania odkrytych schematów do dynamicznej modyfikacji dokumentów.

W [16] zaproponowano podejście bazujące na wektorowej reprezentacji sesji użytkowników. Wymiary przestrzeni odpowiadają w tym przypadku stronom WWW serwisu, wartość konkretnej współrzędnej wektora zależy od czasu spędzonego przez użytkownika na stronie odpowiadającej tej współrzędnej (lub liczby odwiedzeń strony w ramach sesji). Sesje użytkowników są grupowane algorytmem lidera (ang. *leader algorithm*) [7]. Średnie wektory dla poszczególnych grup sesji reprezentują typowe schematy zachowań. W trakcie normalnej eksploatacji serwisu, dla każdego użytkownika tworzony i na bieżąco uaktualniany jest wektor odpowiadający jego dotychczasowemu zachowaniu. Wektor ten jest dopasowywany do odkrytych schematów. Dokument wysyłany użytkownikowi jest uzupełniany o łączniki do stron, których użytkownik jeszcze nie odwiedził, a które występowały w schematach zachowań najlepiej pasujących do dotychczasowego zachowania użytkownika.

Rozwiązanie przedstawione w [10] tworzy profile preferencji w oparciu o współwystępowanie dokumentów w historycznych sesjach użytkowników odnotowanych w logu. Najpierw odkrywane są często występujące asocjacje stron, następnie na bazie znalezionych asocjacji generowane są potencjalnie nakładające się na siebie grupy stron stanowiące profile zachowań. Grupowanie jest realizowane za pomocą algorytmu partycjonowania hipergrafu [8], w którym wierzchołki odpowiadają stronom serwisu, a krawędzie odkrytym asocjacjom. Moduł online systemu automatycznej personalizacji działa na tej samej zasadzie co w implementacji opisanej w [16], z uwzględnieniem dwóch drobnych usprawnień. Po pierwsze, w trakcie dopasowywania bieżącego zachowania użytkownika do odkrytych wcześniej profili, uwzględniane są dostępy realizowane w obrębie pewnego ustalonego okna czasowego, tak aby na generowane dynamicznie łączniki z podpowiedziami wpływ miał tylko ostatni fragment historii sesji. Po drugie, dołączane do wysyłanych użytkownikom dokumentów dynamicznie generowane łączniki są posortowane wg ich wagi (im dalej proponowany dokument znajduje się od bieżącego w topologii serwisu, tym większa jest mu przypisywana waga).

Propozycja opisana w [12] może być postrzegana jako rozszerzenie implementacji opisanej powyżej o uwzględnienie kolejności odwołań realizowanych w ramach poszczególnych sesji. Preferencje użytkowników są w tym wypadku reprezentowane przez zbiory podobnych najczęściej stosowanych ścieżek nawigacyjnych. W celu znalezienia preferencji, realizowany jest dwufazowy algorytm. W pierwszej fazie przeszukiwany jest log serwera WWW w celu znalezienia wszystkich najczęściej występujących ścieżek nawigacyjnych, mających postać wzorców sekwencyjnych. W fazie drugiej, znalezione ścieżki nawigacyjne są grupowane algorytmem o nazwie POPC, kierującym się współwystępowaniem ścieżek w historiach dostępow użytkowników (tzn.



podobieństwo dwóch ścieżek wynika z tego, iż wielu użytkowników, którzy podążają jedną z nich, podąża również drugą).

### 3. Podsumowanie

W artykule przedstawiono aktualne podejścia do personalizacji serwisów WWW, ze szczególnym naciskiem na automatyczną personalizację w oparciu o adaptatywne serwery WWW. Opisana została ogólna koncepcja adaptatywnych serwerów WWW, ze wskazaniem obszarów, w których zastosowanie znajdują techniki eksploracji danych. Szczególny nacisk położony został na problematykę obróbki pliku logu serwera, stanowiącego podstawowe źródło danych do eksploracji.

### Literatura

1. Agrawal R., Imielinski T., Swami A., "Mining association rules between sets of items in large databases". Proc. of the ACM SIGMOD Conference on Management of Data, 1993.
2. Agrawal R., Srikant R.: "Mining Sequential Patterns" Proc. of the 11th International Conference on Data Engineering, 1995.
3. Cooley R., Mobasher B., Srivastava J., "Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns", Proc. of the 1997 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX), 1997.
4. Cooley R., Mobasher B., Srivastava J., "Web Mining: Information and Pattern Discovery on the World Wide Web", Proc. of ICTAI'97, 1997.
5. Catledge L.D., Pitkow J.E., "Characterizing Browsing Strategies in the World Wide Web", Proc. of the 3rd Int'l World Wide Web Conference, 1995.
6. Greening D.R., "Data Mining on the Web", Web Techniques, 2000.
7. Hartigan J., Clustering Algorithms, John Wiley, 1975.
8. Han, E-H, Karypis, G., Kumar, V., Mobasher, B., "Clustering based on association rule hypergraphs", Proc. of SIGMOD'97 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'97), May, 1997.
9. Luotonen A., "The common log file format", <http://www.w3.org>, 1995.
10. Mobasher, B., Cooley, R., Srivastava, J., "Creating Adaptive Web Sites Through Usage-Based Clustering of URLs", Proc. of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX), 1999.
11. Manber U., Patel A., Robison J., "Experience with Personalization on Yahoo!", Communications of the ACM, Vol. 43, No. 8, 2000.
12. Morzy T., Wojciechowski M., Zakrzewicz M.: "Web Users Clustering", Proc. of the ISICIS 2000 Conference, 2000.
13. Pitkow J., "In search of reliable usage data on the www", Sixth Int'l World Wide Web Conference, Santa Clara, California, 1997.

14. Perkowitz, M., Etzioni, O., "Adaptive Web Sites: an AI challenge", Proc. 15th Int. Joint Conf. AI, 1997.
15. Pirolli P., Pitkow J., Rao R., "Silk From a Sow's Ear: Extracting Usable Structure from the World Wide Web", Conference on Human Factors in Computing Systems (CHI 96), 1996.
16. Yan T.W., Jacobsen M., Garcia-Molina H., Dayal U., "From User Access Patterns to Dynamic Hypertext Linking", Proc. of the 5th Int'l World Wide Web Conference, 1996.

Marek Wojciechowski, Maciej Zakrzewicz  
Politechnika Poznańska, Instytut Informatyki  
ul. Piotrowo 3a, 60-965 Poznań  
e-mail: {marek,mzakrz}@cs.put.poznan.pl

# KOSZTOWA OPTYMALIZACJA ZAPYTAŃ W PRAKTYCE

Maciej ZAKRZEWICZ

**Streszczenie:** W większości przypadków, zapytanie użytkownika wyrażone w deklaratorywnym języku SQL może być zrealizowane na wiele różnych sposobów. W systemie zarządzania bazą danych za wybór i implementację najbardziej efektywnej metody wykonania zapytania odpowiada moduł kosztowego optymalizatora zapytań. Artykuł prezentuje wewnętrzne mechanizmy działania optymalizatora kosztowego na przykładzie komercyjnego systemu zarządzania bazą danych Oracle9i.

## Wstęp

Jednym z podstawowych zadań funkcjonalnych stawianych systemom baz danych jest realizacja zapytań użytkowników. Zapytania są zwykle formułowane w języku SQL, którego główną cechą jest deklaratorywność, co oznacza, że użytkownik określa „co ma być znalezione”, a nie „w jaki sposób ma być znalezione”. W związku z tym, dla każdego otrzymanego zapytania, system zarządzania bazą danych musi wygenerować odpowiedni mikroprogram, który zrealizuje zleczone zadanie. W nomenklaturze bazodanowej, mikroprogram taki nazywany jest *planem wykonania zapytania* (query execution plan).

W ogólności, przekład deklaratorywnej treści zapytania na stosowny plan wykonania nie jest jednoznaczny – często, w zależności od stopnia skomplikowania samego zapytania, może istnieć kilka, kilkanaście, a nawet kilka tysięcy alternatywnych rozwiązań. Ponieważ każde z tych rozwiązań może się cechować bardzo różną wydajnością (czasem wykonania zapytania), to wybór właściwego (optymalizacja) powinien być dokonany bardzo precyzyjnie. Przyjęta metodyka wybierania planów wykonania zapytań będzie miała znaczący wpływ na wydajność całego systemu bazy danych.

Moduł systemu zarządzania bazą danych, który odpowiada za dobieranie planów wykonania zapytań nazywa się *optymalizatorem zapytań* (query optimizer). Przez wiele lat ewolucji systemów zarządzania bazami danych, stosowano dwa podstawowe typy optymalizatorów zapytań: optymalizatory regułowe (rule-based) i optymalizatory kosztowe (cost-based). *Optymalizatory regułowe* posługują się predefiniowanym zbiorem reguł postępowania, przy pomocy których generują najbardziej pożądaną plan wykonania konkretnego zapytania. Zaletami tego typu optymalizatorów są prostota budowy i duża szybkość działania.

W przeciwieństwie do optymalizatorów regułowych, *optymalizatory kosztowe* najpierw generują dla danego zapytania wszystkie alternatywne plany, a następnie dla każdego z nich estymują przybliżony czas wykonania. Optymalizator kosztowy wybiera ten plan, dla którego wyliczony oczekiwany czas wykonania jest najmniejszy. Aby uniezależnić się od czasu rzeczywistego, na który wpływa wiele



czynników niezależnych, zamiast pojęcia czasu wykonania zapytania, posługujemy się zwykle pojęciem *kosztu wykonania zapytania* (query cost). W systemie zarządzania bazą danych Oracle jednostka kosztu odpowiada w przybliżeniu czasowi wykonania pojedynczej operacji I/O w systemie operacyjnym.

Należy zwrócić uwagę na to, że sama optymalizacja musi być wykonana w bardzo krótkim czasie tak, aby nie stanowiła istotnego narzutu na całość realizacji zapytania. W środowiskach OLTP oznacza to, że optymalizacja nie powinna zabierać więcej czasu niż ułamek sekundy.

Struktura tego artykułu jest następująca. W rozdziale 2 przedstawiono składniki statystycznych modeli obiektów bazy danych. Rozdział 3 omawia metody estymacji kosztów planów wykonania zapytań, posługując się przykładami pełnego odczytu tabeli, dostępu do tabeli poprzez indeks typu B\*-drzewo, pełnego odczytu indeksu typu B\*-drzewo, sortowania rekordów oraz operacji połączenia tabel. Rozdział 4 stanowi podsumowanie.

## 1 Statystyczne modele obiektów bazy danych

Analityczne wyznaczenie kosztu dla wybranego planu wykonania zapytania wymaga znajomości wielu parametrów bazy danych, np. liczby rekordów, jakie zostaną znalezione, rozmiarów łączonych tabel, liczby poziomów indeksu B\*-drzewo, itd. Pozyskiwanie tych parametrów w czasie pracy optymalizatora wymagałoby realizacji wielu dodatkowych zapytań, które negatywnie wpływałyby na czas samej optymalizacji. W związku z tym, optymalizator kosztowy posługuje się statystycznymi modelami tych obiektów bazy danych, których dotyczy zapytanie. Modele te konstruowane są na żądanie użytkownika i są trwale przechowywane w słowniku danych.

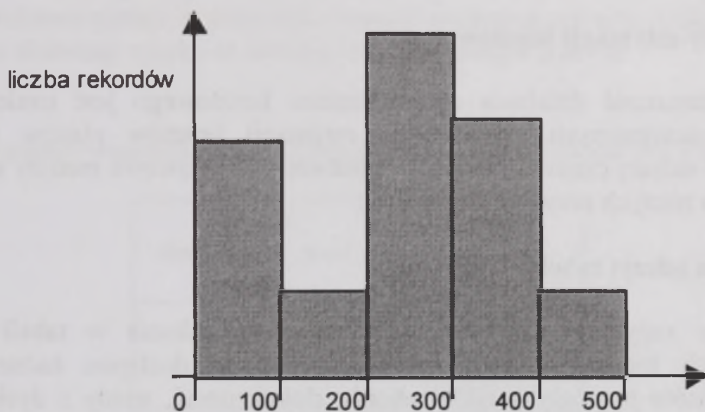
Na model statystyczny tabeli, wykorzystywany przez optymalizator kosztowy systemu zarządzania bazą danych Oracle, składają się m.in. następujące parametry:

- liczba rekordów tabeli
- liczba bloków bazy danych, jakie są lub były wykorzystywane przez tabelę
- średni rozmiar każdej kolumny tabeli
- minimalna i maksymalna wartość znajdująca się w każdej kolumnie
- liczba różnych wartości każdej kolumny
- rozkład wartości kolumny – *histogram* – przedstawiający punkty podziału wartości atrybutu na równoliczne przedziały

Szczegółowego komentarza wymaga pojęcie histogramu kolumny. Podstawowym zadaniem histogramów jest umożliwienie oszacowania liczby rekordów, które spełnią zadany warunek selekcji. W systemach baz danych powszechnie wykorzystuje się dwa typy histogramów: histogramy o stałej szerokości i histogramy o stałej wysokości. System Oracle korzysta z drugiego z wymienionych typów.

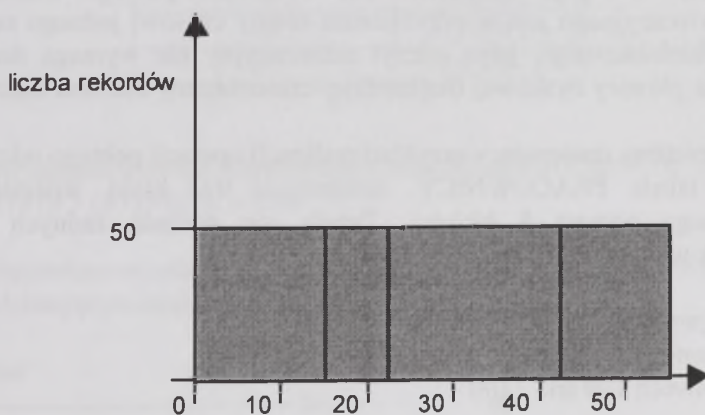
Przykład histogramu o stałej szerokości oraz jego interpretację graficzną przedstawiono na rysunku 1a. Z histogramu tego odczytać możemy, że w tabeli występuje 35 rekordów zawierających wartości od 0 do 100, 11 rekordów zawierających wartości od 100 do 200, itd. Z kolei na rysunku 1b przedstawiono przykład histogramu o stałej wysokości. Możemy z niego odczytać, że 50 rekordów tabeli posiada wartości od 0 do 15, że 50 rekordów posiada wartości od 15 do 22, itd.

histogram : 35, 11, 50, 38, 11



Rys. 1a Histogram o stałej szerokości

histogram : 0, 15, 22, 42, 55



Rys. 1b Histogram o stałej wysokości

Model statystyczny indeksu B\*-drzewo oparty jest m.in. na następujących elementach:

- liczba poziomów drzewa
- liczba bloków liści
- liczba różnych wartości klucza
- współczynnik uporządkowania, który jest liczbą o wartości bliskiej liczbie bloków tabeli, jeżeli jej rekordy są fizycznie uporządkowane zgodnie z indeksem, lub o wartości bliskiej liczbie rekordów, jeżeli rekordy nie wykazują żadnego uporządkowania fizycznego w stosunku do indeksu.

## 2 Metody estymacji kosztów

Skuteczność działania optymalizatora kosztowego jest uzależniona od precyzji dokonywanych przez niego estymacji kosztów planów wykonania zapytań. W dalszej części artykułu przedstawimy podstawowe metody szacowania kosztów dla różnych przykładów planów.

### 2.1 Pełen odczyt tabeli

Gdy zapytanie użytkownika wymaga wyszukania w tabeli rekordów spełniających zadany warunek selekcji, a nie są dostępne żadne struktury indeksów, które pomogłyby takie rekordy zlokalizować, wtedy z dysku pobrane muszą być wszystkie bloki tabeli. Operacja pobierania kolejno wszystkich bloków tabeli nazywa się pełnym odczytem tabeli (full table scan). Ponieważ podczas wykonywania pełnego odczytu tabeli, z dysku pobierane są fizycznie sąsiadujące bloki, dlatego możliwe jest wykorzystanie dyskowego odczytu sekwencyjnego (wieloblokowego), w ramach którego jedno żądanie we/wy powoduje jednoczesne pobranie wielu kolejnych bloków. Czas wykonania jednego wieloblokowego odczytu sekwencyjnego jest w przybliżeniu równy czasowi jednego swobodnego odczytu jednoblokowego, gdyż odczyt sekwencyjny nie wymaga dodatkowego przesuwania głowicy dyskowej (najbardziej czasochłonny element kosztu dostępu do dysku).

Rozważmy następujący przykład realizacji operacji pełnego odczytu tabeli. Dana jest tabela PRACOWNICY, zawierająca 973 bloki. Rozmiar odczytu sekwencyjnego wynosi 8 bloków. Tabela nie posiada żadnych indeksów. Użytkownik wykonuje następujące zapytanie:

```
select max(pensja)
from pracownicy
where id between 100 and 1100
```

Ponieważ nie dysponujemy żadnymi indeksami, jedyną metodą wykonania powyższego zapytania będzie realizacja pełnego odczytu tabeli. Koszt takiej operacji szacowany jest według poniższego wzoru:



$$\text{cost} = \left[ K_m \frac{\text{blocks}}{\text{db\_file\_multiblock\_read\_count}} \right]$$

gdzie:

- *blocks* to liczba bloków tabeli
- *db\_file\_multiblock\_read\_count* to ustawiony w systemie rozmiar wieloblokowych odczytów sekwencyjnych
- $K_m$  to współczynnik korygujący, zależny od wartości *db\_file\_multiblock\_read\_count* (tabelka 1); reprezentuje on ryzyko wydłużenia odczytu sekwencyjnego w przypadku napotkania końca cylindra dyskowego lub końca ekstentu; ryzyko to zależne jest od rozmiaru odczytu

Tabela 1. Wartości współczynnika korygującego  $K_m$  dla odczytów sekwencyjnych

<i>db_file_multiblock_read_count</i>	$K_m$
1	0,596500
2	0,755967
3	0,868342
4	0,958064
5	1,033995
6	1,100481
7	1,160019
8	1,214190
9	1,264068
10	1,310419
11	1,353812

Zgodnie z powyższym wzorem, koszt pełnego odczytu tabeli PRACOWNICY wyniesie:  $1.21419 * (972+1) / 8 = 148$

Poniżej przedstawiono wynik naszych obliczeń dostarczony przez optymalizator kosztowy w działającym systemie Oracle9i:

#### Execution Plan

```
-----
0  SELECT STATEMENT Optimizer=CHOOSE (Cost=148 Card=1 Bytes=7)
1  0  SORT (AGGREGATE)
```

2 1 TABLE ACCESS (FULL) OF 'PRACOWNICY' (Cost=148 Card=1001 Bytes=7007)

## 2.2 Dostęp do tabeli przy użyciu indeksu typu B\*-drzewo

Indeksy B\*-drzewo służą do poprawy efektywności dostępu do pojedynczych rekordów tabeli, spełniających równościowe lub nierównościowe warunki selekcji. Wartość kosztu dostępu do tabeli poprzez indeks B\*-drzewo wynika z odczytu węzłów drzewa na drodze od korzenia do liści, odczytu liści zawierających wskaźniki do rekordów, oraz odczytu samych rekordów z tabeli.

Rozważmy następujący przykład dostępu do tabeli poprzez indeks B\*-drzewo. Dana jest tabela PRACOWNICY, zawierająca 50000 rekordów zapisanych w 973 blokach. Rozmiar odczytu sekwencyjnego wynosi 8 bloków. Tabela posiada indeks I\_ID, zbudowany na kolumnie ID. Model statystyczny tego indeksu to: 2 poziomy, 480 bloków liści, współczynnik zgrupowania = 970. Histogram wartości atrybutu ID został przedstawiony na rysunku 2. Z histogramu tego możemy odczytać np., że w tabeli PRACOWNICY jest tyle samo rekordów posiadających wartości ID pomiędzy 1 a 667, co rekordów posiadających wartości ID pomiędzy 667 a 1334.

1	667	1334	2001	2668	3335	4002	4669	5336	6003	6670	7337	...	50000
---	-----	------	------	------	------	------	------	------	------	------	------	-----	-------

Rys 2. Histogram kolumny ID (razem 75 przedziałów)

Założmy, że użytkownik wykonuje następujące zapytanie:

```
select max(pensja)
from pracownicy
where id between 100 and 1100
```

Do realizacji powyższego zapytania może być użyty jeden z dwóch planów: pełen odczyt tabeli lub dostęp do szukanych rekordów przy pomocy indeksu I\_ID. Koszt pełnego odczytu tabeli został wyliczony w poprzednim podrozdziale. Natomiast dostęp do rekordów tabeli poprzez indeks I\_ID będzie wymagał następujących kroków:

1. Odczyt korzenia indeksu (koszt = 1)
2. Odczyt procentu liści indeksu, równego selektywności warunku selekcji (id between 100 and 1100); według histogramu, w każdym przedziale znajduje się  $50000/75$  rekordów = 667; wartości z przedziału pomiędzy 100 a 1100 mieszczą się w dwóch zakresach histogramu:  $<1;667$ ) i  $<667;1334$ ), w pierwszym stanowią  $(667-100)/667=85\%$ , w drugim  $(1100-667+1)/667=65,07\%$ ; w związku z tym, oczekiwana liczba rekordów spełniających warunek selekcji wynosi  $667*(0.85+0.6507) = 1001$ ; zatem w celu odczytania z indeksu tej liczby wskaźników do rekordów wymagane jest pobranie  $(1001/50000) * \text{liczba\_liści} = 10$  (koszt = 10)

3. Odczyt bloków tabeli, wskazywanych przez wskaźniki z liści indeksu. Liczba odczytów wyliczana jest w oparciu o współczynnik zgrupowania indeksu w następujący sposób:  $(1001/50000) * \text{współczynnik zgrupowania} = 20$  (koszt = 20)

Łączny koszt dostępu do tabeli poprzez indeks B\*-drzewo wyniósł w naszym przykładzie 32. Ponieważ jest to mniej, niż koszt pełnego odczytu tabeli, dlatego optymalizator wybiera ten plan do realizacji zapytania:

#### Execution Plan

```

-----
0  SELECT STATEMENT Optimizer=CHOOSE (Cost=32 Card=1 Bytes=7)
1  0  SORT (AGGREGATE)
2  1  TABLE ACCESS (BY INDEX ROWID) OF 'PRACOWNICY' (Cost=32
Card=1001 Bytes=7007)
3  2  INDEX (RANGE SCAN) OF 'I_ID' (NON-UNIQUE) (Cost=12
Card=1001)

```

### 2.3 Pełen odczyt indeksu bez dostępu do tabeli

Pewne typy zapytań mogą zostać w pełni wykonane bez konieczności dostępu do tabeli bazowej, a jedynie w oparciu o dane zawarte w indeksach. Rozważmy przykład poniższego zapytania do tabeli PRACOWNICY, na której atrybucie ID zbudowany jest indeks B\*-drzewo o nazwie I\_ID:

```

select avg(id)
from pracownicy
where id between 100 and 20000

```

Zauważmy, że zapytanie to może zostać wykonane przy użyciu samego indeksu I\_ID, bez konieczności sięgania do tabeli bazowej PRACOWNICY. Ponadto, indeks I\_ID może być użyty na jeden z dwóch sposobów: (1) z zastosowaniem tradycyjnej nawigacji korzeń-liście w celu wyszukania liści zawierających szukane wartości (tzw. odczyt zakresowy) lub (2) z zastosowaniem wieloblokowego odczytu sekwencyjnego całej struktury indeksu, a następnie wyławiania szukanych wartości z liści. Koszt pełnego odczytu indeksu wyliczony jest analogicznie do kosztu pełnego odczytu tabeli:

$$\text{cost} = \left[ K_m \frac{\text{blocks}}{\text{db\_file\_multiblock\_read\_count}} \right]$$

$$\text{cost} = 1.21419 * (480+1) / 8 = 73$$



Łączny koszt pełnego odczytu indeksu B\*-drzewo wyniósł w naszym przykładzie 73. Ponieważ w tym przypadku jest to mniej, niż koszt zakresowego odczytu indeksu oraz mniej niż koszt pełnego odczytu całej tabeli bazowej, dlatego optymalizator wybiera ten plan do realizacji zapytania:

#### Execution Plan

```
-----  
0  SELECT STATEMENT Optimizer=CHOOSE (Cost=73 Card=1 Bytes=4)  
1  0  SORT (AGGREGATE)  
2  1  INDEX (FAST FULL SCAN) OF 'I_ID' (NON-UNIQUE) (Cost=73  
Card=19891 Bytes=79564)
```

## 2.4 Sortowanie rekordów

Operacja sortowania rekordów jest realizowana m.in. na potrzeby klauzul ORDER BY, GROUP BY, DISTINCT, połączenia, itd. Algorytm sortowania wykorzystywany przez Oracle to wielokanałowy Merge Sort, polegający na sortowaniu (sort) fragmentów tabeli w buforze pamięci operacyjnej i zapisywaniu ich w segmencie tymczasowym, a następnie na stopniowym łączeniu (merge) posortowanych fragmentów w większe, aż do osiągnięcia jednego fragmentu wynikowego. Programista posiada możliwość określania maksymalnego rozmiaru pamięci operacyjnej przeznaczonej na sortowanie oraz rozmiaru pojedynczego okna wykorzystywanego w fazie Merge. System zarządzania bazą danych wykorzystuje technikę bezpośredniego (szybszego) dostępu do segmentu tymczasowego, polegającą na pominięciu bufora danych i standardowego procesu zapisującego bloki z bufora do plików danych.

Rozważmy następujący przykład sortowania danych. Użytkownik wykonuje sortowanie kolumn NAZWISKO i STANOWISKO (według kolumny NAZWISKO) z tabeli PRACOWNICY. W modelu statystycznym, tabela PRACOWNICY posiada 50000 rekordów, kolumna NAZWISKO posiada rozmiar 10 bajtów, a kolumna STANOWISKO rozmiar 9 bajtów. System dysponuje 34428 bajtami pamięci operacyjnej dla sortowania, a rozmiar pojedynczego okna wykorzystywanego w fazie Merge wynosi 4096 bajtów. Poniżej przedstawiono treść zapytania użytkownika.

```
select nazwisko, stanowisko  
from pracownicy  
order by nazwisko
```

Rozmiar sortowanych rekordów określany jest jako suma rozmiarów wszystkich sortowanych kolumn, plus po jednym bajcie na kolumnę, plus 10 bajtów binarnej sygnatury, która faktycznie wykorzystywana jest do operacji porównań. W omawianym przypadku oznacza to, że sortowaniu podlegać będą rekordy  $10+9+1+1+10 = 31$  bajtowe. Rekordów tych będzie 50000, co daje łączny ich

rozmiar  $50000 \cdot 31 = 1550000$  bajtów, podzielonych na 2048-bajtowe bloki, których będzie  $1550000 / (2048 - 24) = 766$  (24 bajty nagłówek bloku).

Znając rozmiar bufora w pamięci operacyjnej, możemy obliczyć, na ile fragmentarycznych sortowań zostaną podzielone dane:  $1550000 / 34428 = 46$ . Następnie wyznaczymy liczbę kanałów łączenia, jakie zmieszczą się w pamięci operacyjnej. Każdy kanał zajmuje 4096 bajtów, a zatem do dyspozycji mamy  $34428 / 4096 = 5$  kanałów. W związku z tym, algorytm Merge Sort będzie wymagał  $\log_5 46 = 3$  faz łączenia. Koszt każdej fazy łączenia to odczyt wszystkich sortowanych bloków, a następnie ich wieloblokowy zapis sekwencyjny (rozmiar zapisu równy liczbie kanałów). To daje koszt pojedynczej fazy:  $766 + 766 / 5 = 920$ . Koszt całego sortowania wyniesie  $\frac{1}{2} \cdot (3 \cdot 920 + 766) = 1763$ , gdzie  $\frac{1}{2}$  reprezentuje poprawę wydajności wynikającą z dostępu bezpośrednich. Należy jeszcze uwzględnić koszt początkowego pełnego odczytu tabeli (148), aby uzyskać ostateczny koszt planu wykonania zapytania: 1911. Poniżej przedstawiono wynik dostarczony przez optymalizator kosztowy:

#### Execution Plan

---

```
0  SELECT STATEMENT Optimizer=CHOOSE (Cost=1911 Card=50000
Bytes=950000)
  1  0  SORT (ORDER BY) (Cost=1911 Card=50000 Bytes=950000)
    2  1  TABLE ACCESS (FULL) OF 'PRACOWNICY' (Cost=148 Card=50000
Bytes=950000)
```

### 2.5 Połączenie tabel

Operacja połączenia tabel w bazie danych Oracle jest realizowana jedną z trzech metod: Nested Loops, Sort Merge lub Hash Join. Wyboru najlepszej z nich dokonuje optymalizator kosztowy, odpowiednio estymując koszt planu wykonania zapytania. Poniżej przedstawiono sposób wyliczania kosztu połączenia metodą Sort Merge dla następującego zapytania użytkownika:

```
select p1.nazwisko, p2.nazwisko
from pracownicy p1, pracownicy p2
where p1.stanowisko=p2.stanowisko
```

Operacja połączenia metodą Sort Merge polega na wstępnym posortowaniu łączonych rekordów w obu tabelach, a następnie na wykonaniu dwukanałowego łączenia odpowiadających sobie rekordów. Koszt takiej operacji połączenia stanowi sumę kosztów odczytu każdej z tabel oraz ich sortowania. Ponieważ rozmiar sortowanego rekordu jest taki sam, jak w poprzednim podrozdziale, dlatego sortowanie będzie kosztować łącznie  $1763 \cdot 2 = 3526$ , a koszt pełnego odczytu obu łączonych tabel to  $2 \cdot 148 = 296$ . Ostateczny koszt realizacji operacji

połączenia metodą Sort Merge to  $3526 + 296 = 3822$  i jest to w tym przypadku najniższy koszt, jaki znalazł optymalizator kosztowy:

#### Execution Plan

---

```
0  SELECT STATEMENT Optimizer=CHOOSE (Cost=3822
Card=795468474 Bytes=30227802012)
  1  0  MERGE JOIN (Cost=3822 Card=795468474 Bytes=30227802012)
    2  1  SORT (JOIN) (Cost=1911 Card=50000 Bytes=950000)
      3  2  TABLE ACCESS (FULL) OF 'PRACOWNICY' (Cost=148
Card=50000 Bytes=950000)
        4  1  SORT (JOIN) (Cost=1911 Card=50000 Bytes=950000)
          5  4  TABLE ACCESS (FULL) OF 'PRACOWNICY' (Cost=148
Card=50000 Bytes=950000)
```

### 3 Wnioski końcowe

Trafność estymacji kosztu planu wykonania zapytania ma wpływ na wybory dokonywane przez optymalizator. Z kolei trafność tych wyborów determinuje całkowitą wydajność systemu. W artykule przedstawiono ogólne zasady wyznaczania kosztów na przykładzie komercyjnego optymalizatora zapytań systemu zarządzania bazą danych Oracle9i. Zwrócono uwagę na znaczenie uwzględniania m.in. rozmiarów danych, rozkładów ich wartości, ilości dostępnej pamięci operacyjnej, charakterystyk indeksów oraz natury dostępow I/O.

Można zauważyć, że zaprezentowane metody estymacji kosztu ignorują istnienie pewnych znaczących składowych efektywności, spośród których wyróżnić należy buforę bazy danych, buforę sterowników dyskowych, czy parametry czasowe dysków twardej. W pewnym zakresie fakt ten wynika z trudności modelowania matematycznego tych elementów oraz z rygorystycznych wymagań związanych z wydajnością samego procesu optymalizacji. Oczekuje się, że w kolejnych wydaniach, kosztowe optymalizatory zapytań będą się cechować większą świadomością obecności takich składników „wspomagających”, co pozwoli im uzyskać lepszą zbieżność kosztów prognozowanych z rzeczywistymi.

Maciej Zakrzewicz  
Politechnika Poznańska, Instytut Informatyki  
maciej.zakrzewicz@cs.put.poznan.pl



# METODY POPRAWY EFEKTYWNOŚCI EKSPLOKACJI DANYCH METODĄ ASOCJACJI

Anna ZYGMUNT, Marek A. VALENTA

**Streszczenie:** W artykule przedstawione zostaną problemy związane z poprawą efektywności eksploracji danych metodą asocjacji. Poprawa efektywności może odbywać się na etapie przygotowania danych, eksploracji i porządkowania odkrytych wyników. Główny nacisk położony został na przedstawienie problemu eksplozji ilości odkrywanych wzorców oraz określenie ich atrakcyjności dla użytkownika. Przedstawiono metody redukcji liczby otrzymywanych reguł oraz zaprezentowano koncepcję metody ich porządkowania.

## Wstęp

Celem eksploracji danych jako dziedziny badań jest opracowanie metod pozwalających analizować ogromne ilości danych znajdując w nich interesujące regularności czy wyjątki. Eksploracja danych stanowi jeden z etapów procesu zwanego odkrywaniem wiedzy w bazach danych (potocznie często pojęcia te używane są zamiennie). Typowe problemy, jakie należy rozwiązać przy opracowaniu efektywnych algorytmów eksploracji danych, wynikają zarówno z dużych rozmiarów nie tylko zbiorów danych przeznaczonych do eksploracji, ale i zbiorów wynikowych wzorców (na przykład w postaci reguł) stanowiących odkrytą wiedzę. Prowadzone badania ukierunkowane są na znajdowanie najkorzystniejszych (a więc również najefektywniejszych) rozwiązań zarówno na etapie przygotowania danych, jak też ich eksploracji, a wreszcie porządkowania uzyskanych wyników (w szczególnym przypadku zastosowania metody reguł asocjacyjnych jest to zbiór reguł).

W odkrywaniu reguł asocjacyjnych szczególną uwagę jak dotąd zwracano na optymalizację etapu zastosowania określonego algorytmu na odpowiednio przygotowane dane, niedostateczną uwagę poświęcając, według autorów, etapowi porządkowania wyników.

Okazuje się, że ograniczanie liczby otrzymywanych w procesie odkrywania wiedzy wzorców, poprzez wprowadzenie statystycznych miar ich „ważności”, jest niewystarczające. Jak pokazuje praktyka, duża część otrzymywanych tak reguł jest trywialna, zbyt szczegółowa lub po prostu nieinteresująca dla końcowego użytkownika. Bardzo trudno jest ocenić poprzez statystykę jakie reguły dla użytkownika będą rzeczywiście interesujące. Z tego powodu konieczna jest dodatkowa „obróbka” otrzymanych wzorców, a więc ich czyszczenie, porządkowanie czy też grupowanie, tak by użytkownik sam mógł znaleźć wśród nich te interesujące.

## 1. Problem odkrywania reguł asocjacyjnych

Zaproponowane przez Agrawal'a w [1] *odkrywanie reguł asocjacyjnych* jest jak dotąd jedną z nielicznych rzeczywiście oryginalnych propozycji nowego podejścia do eksploracji danych w ciągu ostatnich kilku lat.

Zaproponowana w pierwotnej wersji, do analizy koszyka zakupów, eksploracja reguł asocjacyjnych, obecnie odeszła daleko od czysto marketingowego zastosowania, stając się jedną z najpopularniejszych metod eksploracji danych.

**Celem odkrywania reguł asocjacyjnych** jest znajdowanie interesujących związków między atrybutami w bardzo dużych bazach danych.

Baza danych składa się z sekwencji zapisów charakteryzowanych przez wartości atrybutów. Każdy taki zapis traktowany jest jako *transakcja*.

Podzbiór wartości atrybutów zwany jest *zbiorem elementowym* (ang. itemset).

*Poparcie* (ang. support) zbioru elementowego określa liczba transakcji, w których on występuje. Zbiór elementowy jest *częsty*, jeżeli wartość jego poparcia jest większa lub równa od wyznaczonej przez użytkownika wartości progowej, zwanej *minimalnym poparciem*. Reguła asocjacyjna jest wyrażeniem „jeżeli przesłanka, to konkluzja”, gdzie zarówno przesłankę jak i konkluzję tworzą rozłączne zbiory elementowe.

*Poparcie reguły* określa liczba transakcji, w której występuje zarówno przesłanka jak i konkluzja tej reguły. Z kolei *potwierdzenie reguły* (ang. confidence) określa liczba transakcji zawierających konkluzję pod warunkiem, iż zawierają przesłankę. Reguła jest *regułą silną*, gdy zbiór elementowy z przesłanki i konkluzji jest częsty, a jej *potwierdzenie* jest większe lub równe od określonej przez użytkownika wartości progowej.

Proces odkrywania reguł asocjacyjnych składa się z dwóch kroków:

1. znajdowanie częstych podzbiorów elementowych,
2. generowanie z nich silnych reguł.

Ze względu na szerokie możliwości wykorzystania eksploracji reguł asocjacyjnych w wielu obszarach zastosowań (np. analiza: transakcji sprzedaży, danych telekomunikacyjnych, tekstu, wizyt użytkowników na stronach WWW, itp.) zaproponowano wiele wydajnych algorytmów. Pierwszym zaproponowanym algorytmem eksploracji reguł asocjacyjnych był algorytm Apriori. Szybko okazało się, że mimo iż problem odkrywania jednoczesnego występowania elementów w małych zbiorach danych jest stosunkowo prostym zadaniem, jednak ogromne zbiory danych powodują, że zadanie przestaje być trywialne i wymaga wydajnych algorytmów. Algorytm Apriori ze względu na sposób generowania kandydatów i przechowywania o nich informacji oraz konieczność wielokrotnego przeglądania bazy danych okazuje się mało efektywny zwłaszcza w przypadku baz danych, w

których można znaleźć bardzo długie częste zbiory elementowe (na przykład baza danych zawierająca informacje o spisie ludności).

Próby poszukiwania wydajnego algorytmu, który w jak najkrótszym czasie znajdowałby częste zbiory elementowe i z nich generował reguły, cały czas są kontynuowane. Jak dotąd nie udało się znaleźć takiego, który działałby jednakowo efektywnie w przypadku każdego typu bazy danych (bez względu na: liczbę transakcji, liczbę atrybutów i przyjmowanych przez nie wartości).

Opracowywane algorytmy w różny sposób próbują rozwiązać podstawowe problemy mające decydujący wpływ na poprawę efektywności działania. Wśród koniecznych do rozwiązania problemów można wymienić:

- sposób generowania kandydatów (poprzez ograniczanie zbioru oraz zrównoleglanie jego tworzenia),
- sposób obliczania poparcia kandydatów,
- liczbę przeglądnięć całej bazy (dążenie do ich redukcji),
- dodatkowe wprowadzone struktury (na przykład tablice mieszające),
- sposób wykorzystania pamięci.

Analizując opracowania dotyczące metod eksploracji częstych wzorców można dojść do wniosku, iż w zaproponowanych podejściach do rozwiązania tego problemu można wyodrębnić trzy grupy algorytmów.

**Pierwsza** z nich wykorzystuje sieć częstych zbiorów elementowych; na  $k$ -tym poziomie sieci znajduje się częste  $k$ -elementowe zbiory.

**Druga** grupa algorytmów zakłada wydobywanie maksymalnych (w aspekcie zawierania się zbiorów) częstych zbiorów elementowych

**Trzecia** grupa reprezentowana jest przez algorytmy Close [5] i Charm [8] i bazuje na teoretycznych podstawach przedstawionych w [3].

Algorytmy te wykorzystują tak zwane domknięcie połączenia Galois'a. W podejściu tym wprowadza się częste domknięte zbiory elementowe, które wraz z poparciami wydobywane są z bazy poziomami. Z tego też powodu nie ma konieczności rozpatrywania wszystkich częstych zbiorów elementowych podczas obliczania poparcia zbiorów elementowych i w związku z tym przestrzeń przeszukiwań ulega drastycznej redukcji, zwłaszcza w przypadku silnie skorelowanych danych.

## 2. Problem ilości odkrywanych wzorców

Niewątpliwa zaleta eksploracji danych metodą asocjacji polega na jej kompletności, czyli możliwości znalezienia wszystkich reguł istniejących w bazie. Niestety, zaleta ta zaczyna być odczuwana jednocześnie jako jedna z najsłabszych stron metody, gdy przystępujemy do interpretowania uzyskanych wyników.



Charakterystyczne bowiem dla procesu odkrywania reguł asocjacyjnych jest dostarczanie ogromnych ilości reguł wynikowych. Ilość ta jest na tyle duża, iż utrudnia użytkownikowi zidentyfikowanie wśród nich tych rzeczywiście go interesujących.

Niestety, jak dotąd nie uzyskano satysfakcjonujących rezultatów. Główny nurt badań obserwowany od momentu zaproponowania algorytmu Apriori skupia się na opracowaniu coraz efektywniejszych algorytmów znajdowania zbiorów częstych. A wydaje się, że końcowe przetwarzanie otrzymanych reguł odgrywa równie ważną rolę w poprawie jakości otrzymywanej wiedzy.

Zaproponowane wraz ze sformułowaniem problemu odkrywania reguł asocjacyjnych, progi minimalnego poparcia i potwierdzenia, mają decydujący wpływ na ograniczanie liczby wydobytych reguł.

Poparcie odgrywa bardzo ważną rolę w wyrażaniu statystycznej istotności wzorca. Jest ono również ważnym czynnikiem wpływającym na zmniejszanie wielkości przestrzeni przeszukiwań, z powodu posiadania bardzo przydatnej własności – tak zwanego „domknięcia w dół” (antymonotoniczności).

Własność ta polega na tym, że jeżeli zbiór elementów  $I$  występuje w  $t$  transakcjach, to każdy podzbiór  $I$  musi wystąpić w co najmniej  $t$  transakcjach. Tak więc, jeżeli zbiór elementów  $I$  nie osiągnie progu minimalnego poparcia (nie wystąpi w odpowiedniej liczbie transakcji), żaden jego nadzbiór tego progu nie osiągnie. Można więc te nadzbiory przy dalszych poszukiwaniach zignorować i nie brać ich pod uwagę.

Z kolei potwierdzenie mierzy warunkowe prawdopodobieństwo zdarzeń wyrażone przez określoną regułę. Niestety, pomiar pewności może być w pewnych rzeczywistych sytuacjach mylący (przypadek zdarzeń niezależnych). Często wykorzystywaną metodą odrzucania takich zbiorów elementów jest statystyczny test chi-kwadrat. Ocenia on istotność wzorców asocjacyjnych.

Ograniczanie ilości reguł wynikowych poprzez ustawienie minimalnych progów poparcia i potwierdzenia, choć w zdecydowany sposób ogranicza liczbę otrzymywanych reguł, wymaga od użytkownika intuicyjnego przewidywania wyników i charakteru danych. Otrzymywane wynikowe reguły, przy zbyt wysokich ustawieniach progów, najczęściej rozczarowują, potwierdzają bowiem ogólnie znane prawa, gubiąc rzadkie, aczkolwiek być może ciekawsze przypadki. Z kolei zbyt niskie ustawienie progów, spowoduje eksplozję otrzymanych zbiorów częstych, a co za tym idzie – eksplozję ilości reguł. Tak więc, optymalny wybór obydwu parametrów wymaga najczęściej obecności eksperta dziedzinowego oraz dokładnego poznania charakteru danych.

### 3. Określenie atrakcyjności wzorców

W wyniku procesu wydobywania wiedzy z danych otrzymuje się określonego rodzaju wzorce. Można postawić tezę, że wybór rodzaju wzorców definiuje w pewnym sensie sposób określania, które wzorce można traktować jako potencjalnie użyteczne.

„W pewnym sensie”, gdyż niestety okazuje się, że nie wszystkie otrzymane wzorce są interesujące dla danego użytkownika lub w danej chwili. Zazwyczaj bowiem to, co jest interesujące zależy od sytuacji i jest mocno uzależnione od cech samego użytkownika, jego osobistych aspiracji i perspektyw.

Wiedza trywialna dla jednego użytkownika może nie być prosta i oczywista dla innego. Trudno jest jednak przewidzieć, co dla danego użytkownika będzie faktycznie interesujące i użyteczne. Dostarczanie mechanizmów do filtrowania reguł w oparciu o subiektywne oceny jest sprawą ważną, ale niestety bardzo trudną.

Powodów, dla których pewne reguły mogą okazać się nieinteresujące może być wiele, chociażby:

- wydobyta reguła reprezentuje wiedzę znaną użytkownikowi (dzieje się tak często z regułami bardzo silnymi – najczęściej reprezentują ogólnie znane zależności),
- reguła odwołuje się do nieistotnych atrybutów,
- reguła jest nadmiarowa (czyli można ją w jakiś sposób wywnioskować z innych reguł).

Kloesgen w [4] zdefiniował pewne kryteria atrakcyjności wzorców (stawiane wymagania):

- pewność – rozumiana jako istotność odkrycia według statystycznych kryteriów,
- nadmiarowość – spowodowana podobieństwami między odkryciami,
- użyteczność – czyli odkrycie spełnia oczekiwania i cele użytkownika,
- oryginalność – czy odkrywanie wzorców nie pokrywających się ze znaną już wiedzą,
- prostota – mierzona złożonością syntaktyczną,
- uogólnienie – procent ogółu, do którego odnosi się odkrycie.

Z kolei Silberschatz i Tuzhilin [6] kryteria atrakcyjności podzielili na dwie kategorie, które zostały ogólnie zaakceptowane:

1. miary obiektywne,
2. miary subiektywne

Miary obiektywne istotności odkrywanych reguł mogą być realizowane przy pomocy technik nie wymagających jakiegokolwiek wiedzy odnośnie dziedziny czy zastosowania. W literaturze można znaleźć wiele stosownych opracowań. Druga kategoria miar służy do pomiaru subiektywnej atrakcyjności wzorca dla użytkownika.

Wzorec jest atrakcyjny dla danego użytkownika, jeśli jest dla niego nieoczekiwany lub poprzednio mu nieznanym oraz jeśli poznanie tego wzorca pomoże mu podjąć działania prowadzące do uzyskania własnych korzyści.

Nieoczekiwane wzorce są interesujące, gdyż dostarczają użytkownikowi nowej wiedzy. Ale czasem użytkownikowi nie zależy na znalezieniu takich wzorców, a jedynie na sprawdzeniu, czy jego obecny stan wiedzy odnośnie zawartości bazy danych jest poprawny.

Zauważono że miary obiektywne, aczkolwiek pod wieloma względami użyteczne, okazują się niewystarczające przy ustalaniu atrakcyjności odkrytych wzorców. Potrzebne są więc miary subiektywne. Niestety, opracowanie ogólnych zasad miar subiektywnych jest bardzo trudne, gdyż:

- zainteresowania użytkownika uzależnione są od dziedziny,
- zbiór odkrytych w bazie wzorców nie jest w całości jednakowo atrakcyjny dla każdego użytkownika; różni użytkownicy są zainteresowani różnymi grupami znalezionych wzorców,
- nawet dla samego użytkownika, w różnych momentach, jego zainteresowania mogą się różnić w zależności od określonej sytuacji.

Tak więc przy definiowaniu miar subiektywnych konieczna jest duża wiedza nie tylko o bazie danych, czy dziedzinie, ale o zainteresowaniach użytkownika.

Prościej jest więc zrealizować miary subiektywne w konkretnym zastosowaniu w danej dziedzinie (zależne dziedzinowo).

#### **4. Redukcja liczby otrzymanych reguł**

W wyniku działania algorytmu eksploracji danych metodą asocjacji, ilość otrzymanych reguł może, w zależności od charakteru danych, osiągać trudne do ogarnięcia rozmiary. Ta wada metody nie jest ostro widoczna w przypadku analizy koszyka zakupów będącego sztandarowym przykładem zastosowania algorytmów reguł asocjacyjnych. Wynika to z faktu, iż zazwyczaj w pojedynczym rekordzie transakcji liczba elementów jest niewielka w porównaniu z ogólną liczbą różnych elementów (asortyment sklepu). Ta rzadkość elementów wykorzystywana jest przez algorytm Apriori (i jemu podobne) dla efektywności eksploracji.

Problem zaczyna się pojawiać w przypadku danych gęstych, na przykład w analizie danych telekomunikacyjnych, analizie spisów ludności, w zadaniach klasyfikacyjnych i predykcyjnych.

Dane te charakteryzują się następującymi własnościami:

- wiele często występujących elementów (np. płeć=kobieta),
- elementy są silnie skorelowane,
- wiele elementów występuje w każdym rekordzie.

Ogromna liczba wygenerowanych reguł stanowi poważną przeszkodę lub wręcz uniemożliwia użytkownikowi ich analizę w celu znalezienia tych rzeczywiście interesujących czy użytecznych. Niezadowolające byłoby jednak podejście polegające na arbitralnym wybraniu małego podzbioru reguł i



zaprezentowanie go użytkownikowi, podczas gdy w rzeczywistości w danych ukryta jest ogromna ilość reguł. Podzbiór taki byłby tylko częściowym obrazem rzeczywistości. Czym należałoby się poza tym kierować przy wyborze takiego, a nie innego podzbioru?

Czy możliwe byłoby więc zachowanie w pełni podstawowej siły reguł polegającej na ich zupełności „nie zalewając” jednocześnie użytkownika ogromną ich ilością?

W literaturze widoczne są dwa podejścia do zredukowania ilości otrzymanych reguł:

- porządkowanie *a posteriori* – przetwarzanie końcowe (postprocessing),
- porządkowanie *apriori* – przetwarzanie wstępne (preprocessing)

Pierwszy ze sposobów; porządkowanie *a posteriori*, polega na wygenerowaniu wszystkich silnych reguł, a następnie zastosowaniu do nich pewnych mechanizmów poprawy jakości oraz filtrujących w celu wyeliminowania z nich różnego rodzaju anomalii, takich jak na przykład nadmiarowości.

Drugi sposób, określany jako porządkowanie *apriori*, polega na wydobyciu tylko pewnego podzbioru silnych reguł asocjacyjnych, nie zaś całego zbioru, jak w poprzednim podejściu. Uzyskuje się to poprzez rozszerzenie podstawowego algorytmu – opartego o pewność i poparcie – o dodatkowe, wbudowane w algorytm kryteria (często dobierane indywidualnie do rozpatrywanych zagadnień).

## 5. Znaczenie metod przetwarzania końcowego

Można stwierdzić, że proces odkrywania wiedzy jest procesem wymagającym uczenia; pierwsze podejście do eksploracji danych najczęściej bowiem polega na wstępnej analizie odkrytych reguł w celu uzyskania informacji, jakiego rodzaju wiedzę można odkryć w wybranych i przygotowanych wcześniej danych. Dopiero wtedy można dokładnie sprecyzować, jakiego rodzaju związki są interesujące, które atrybuty powinny się znaleźć w przesłance czy konkluzji, na jakim poziomie ustawić minimalne wartości poparcia, czy potwierdzenia. I powtórzyć proces odkrywania. W takim procesie człowiek odgrywa centralną rolę. Brachman [2] podejście takie określa mianem „archeologii danych”: precyzyjne określenie strategii odkrywania, decydujące problemy i podstawowe cele czy zadania zostają wypracowane w czasie takiej nieprzewidywalnej interaktywnej eksploracji danych z centralną rolą człowieka. Archeologia danych może więc być traktowana jako proces ze szczególną rolą człowieka zadawania zapytań, eksploracji, analizy, interpretacji i uczenia się przy wzajemnym oddziaływaniu z bazą danych.

Zanim jednak będzie to możliwe, konieczne jest przedstawienie zbioru odkrytych reguł w takiej postaci, by człowiek mógł nabrać ogólnego spojrzenia na występujące w nich zależności.

Techniki przetwarzania końcowego wspomagają użytkownika w poruszaniu się po zbiorze odkrytych reguł i pomagają mu ten zbiór interpretować.

Wiele komercyjnych systemów oferuje możliwości wykonywania przetwarzania końcowego odkrytych reguł. Systemy odkrywające reguły asocjacyjne, takie jak na przykład najbardziej popularne: SGI MineSet, IBM Intelligent Miner, Clementine, DBMiner prezentują odkryte reguły asocjacyjne w formie tekstowej lub graficznej. Zazwyczaj reguły są uporządkowane według wartości ich pewności i poparcia. Takie podejście powoduje, że zazwyczaj użytkownik skupia się tylko na regułach o bardzo wysokich wartościach potwierdzenia i poparcia. Jednak wysokie wartości tych parametrów niekoniecznie muszą oznaczać ważność, czy atrakcyjność reguły. Można więc stwierdzić, że ta prosta strategia nie pokazuje w pełni potencjału odkrytych reguł.

Przed wszystkim konieczne jest ograniczenie zbioru odkrytych reguł do takiej ilości, która możliwa by była do zanalizowania. Pozostaje pytanie: na jakiej zasadzie wybrać ten zbiór?

Zredukowany zbiór reguł powinien z jednej strony dalej być zupełny, by można było dostać obraz wszystkich zależności, z drugiej zaś nie zawierać wszystkich reguł.

Jak więc pogodzić te wydawałoby się sprzeczne wymagania?

Wyjściem z takiej sytuacji byłoby zbudowanie na podstawie odkrytych reguł pewnej **struktury bazowej** (nieredukowalnego zbioru reguł), z której, po zastosowaniu pewnych reguł inferencyjnych (wnioskowania), można by było otrzymać pozostałe reguły.

Taka struktura bazowa zawierałaby najbardziej ogólne i najprostsze reguły dające obraz ogólnych związków zachodzących w danych.

Jednak te ogólne reguły, choć proste, łatwo zrozumiałe i dające obraz całości – okazują się niewystarczające. Często bowiem szczegóły okazują się bardziej interesujące. W pierwszym podejściu bowiem, dzięki regułom ogólnym użytkownik zaczyna być w stanie bliżej sprecyzować, to co go interesuje. A potem, kolejno chciałby się dowiedzieć czegoś więcej.

Sytuacja może być porównana do szukania artykułu w materiałach konferencyjnych poruszającego interesującą nas problematykę. Tytuły zazwyczaj niewiele mówią, ale po streszczeniach jesteśmy już w stanie wybrać interesujące nas artykuły.

Jeżeli więc dojdziemy do wniosku, po przeczytaniu streszczenia, że jesteśmy zainteresowani pewnymi szczegółami, łatwiej znajdziemy je w pełnym artykule.

Analogicznie powinno się więc umożliwić poruszanie się po zbiorze reguł i analizowanie jego zawartości w bardziej zaawansowany sposób niż tylko uporządkowanie według wartości minimalnego potwierdzenia i poparcia. Jednocześnie konieczne jest ograniczenie ilości zbioru reguł.

W ramach prac badawczych prowadzonych w Katedrze Informatyki Akademii Górniczo-Hutniczej w Krakowie zaproponowano i przetestowano metodę strukturyzacji otrzymywanych reguł. Metoda ta polega na wyodrębnieniu w zbiorze odkrytych reguł asocjacyjnych stosunkowo niewielkiego zbioru –



„nieredukowalnych reguł”, a następnie wykorzystanie go do stworzenia struktury logicznej hierarchii, pomocnej do znalezienia interesującego aspektu w pozostałych regułach.

Szczegółowy opis koncepcji metody oraz wyniki przeprowadzonych eksperymentów przedstawione zostały w [9].

## 6. Porządkowanie reguł wygenerowanych ze zbiorów częstych

W czasie generowania reguł asocjacyjnych, ze znalezionych wcześniej częstych zbiorów elementowych, nie można wykorzystać żadnej przydatnej własności, jak w przypadku znajdowania zbiorów częstych, która ograniczałaby ilość tych reguł. W efekcie, otrzymany zbiór odkrytych silnych reguł asocjacyjnych jest, w zależności oczywiście od założonego progu minimalnego poparcia i potwierdzenia, na tyle duży, iż analizowanie go w celu oceny przydatności odkrytych reguł wydaje się być nieefektywny i zniechęcający.

Okazuje się jednak, że w tym na pozór nieuporządkowanym i chaotycznym zbiorze reguł można znaleźć pewien podzbiór reguł, który nazwijmy *zbiorem generującym*. Wyodrębnienie takiego podzbioru pozwala na traktowanie dużej liczby odkrytych reguł asocjacyjnych jako nadmiarowych względem reguł należących do zbioru generującego (można je z tego zbioru w jakiś sposób wywnioskować). Doświadczenie pokazuje jednak, iż reguł nadmiarowych nie można całkowicie odrzucić, przedstawiając użytkownikowi tylko zbiór generujący, tworzą go bowiem tylko najbardziej ogólne reguły ze wszystkich wygenerowanych reguł asocjacyjnych. Interesujące bowiem mogą być szczegóły. Ogólne reguły mogą stanowić pewnego rodzaju przewodnik dalszych poszukiwań czyniąc ten proces bardziej ukierunkowanym. Tak więc, w myśl zasady „od ogółu do szczegółu” na podstawie znalezionej ogólnej reguły można stwierdzić, czy oglądanie jej uszczegółowienia będzie interesujące.

Koncepcja zaproponowanej metody strukturalizacji oparta została o Analizę Konceptu Formalnego [3]. Podstawowym pojęciem w tej teorii jest formalny koncept, zwany również pojęciem, będący domkniętym zbiorem elementowym. *Pojęcia* stanowią pewien podzbiór częstych zbiorów elementowych. Okazuje się również, że poparcie zbioru elementowego jest równe poparciui najmniejszego domkniętego zbioru elementowego zawierającego dany zbiór elementowy. Tak więc wystarczy rozpatrywać zależności między *pojęciami*. W pracy [9] zaproponowano kilka twierdzeń, na podstawie których można grupować odkryte reguły w pewne klasy reguł równoważnych, których reprezentantem jest reguła wyrażająca zależności między sąsiednimi pojęciami. Pojęcia tworzą sieć, zwaną siecią Galoise’a, która z kolei zawiera się w sieci częstych zbiorów elementowych (szczegóły przedstawione są w [9]).

Tak więc wyodrębnienie zbioru generującego, poza ograniczaniem liczności zbioru odkrytych reguł, ma dodatkową wartość: zbiór reguł nadmiarowych można podzielić na równoważne grupy, a każdą z reguł ze zbioru

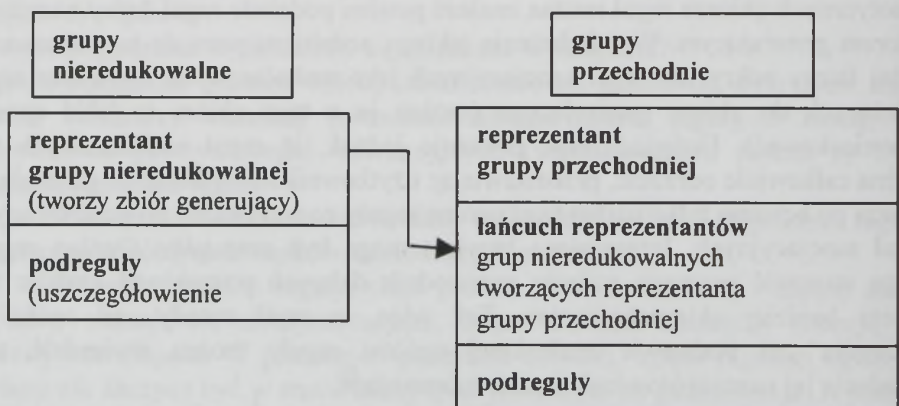


generującego uczynić regułą reprezentantem danej grupy reguł nadmiarowych. Zostało to przedstawione na rysunku 1.

Przedstawione na rysunku nieredukowalne grupy reguł to takie grupy reguł, których reprezentant jest regułą ze zbioru generującego, nie może być więc wynioskowany z innych reguł (np. w wyniku zastosowania prawa o przechodniości).

Pozostałe reguły wchodzące w skład danej nieredukowalnej grupy, mają to samo poparcie i potwierdzenie oraz łączą te same koncepty, co reguła-reprezentant. Mają jednak dodatkowe elementy w przesłance lub konkluzji. Stanowią więc uszczegółowienie reguły-reprezentanta.

Z kolei grupę przechodnią tworzą reguły łączące tę samą parę konceptów pośrednich (jest między nimi pewna liczba konceptów oraz reguły je łączące). Każda grupa przechodnia ma swojego reprezentanta (najbardziej ogólną regułą grupy), to samo poparcie oraz potwierdzenie, jak również tę samą grupę reguł ze zbioru generującego, który je tworzą.



Rys. 1. Struktura zbioru odkrytych reguł asocjacyjnych

## 7. Opis przeprowadzonych eksperymentów

Wykonano szereg eksperymentów, których celem było sprawdzenie, w jaki sposób zaproponowany mechanizm porządkowania odkrytych reguł asocjacyjnych, polegający na ich grupowaniu w zbiory równoważnych reguł oraz podsumowaniu przy wykorzystaniu zbioru generującego, wpływa na ograniczenie ich ilości.

Ponieważ, jak już zostało zauważone, na liczbę wygenerowanych reguł decydujący wpływ ma charakter samych danych, eksperymenty przeprowadzono na bazach danych różniących się między sobą ilością danych, długością rekordów oraz liczbą elementów. Są wśród nich zarówno bazy gęste, wykorzystywane do

algorytmów znajdujących maksymalne zbiory elementowe, jak również bazy rzadkie. W testach wykorzystano zarówno bazy danych z danymi rzeczywistymi, jak i bazy z danymi wygenerowanymi losowo.

W ramach eksperymentu m.in. porównano liczbę nieredukowalnych reguł z ogólną liczbą wygenerowanych reguł ze zbiorów częstych, dla wybranych baz danych.

Dla wszystkich baz danych można zaobserwować prawidłowość, że dla wybranych wartości minimalnego poparcia wraz ze zmniejszaniem wartości minimalnego potwierdzenia, rośnie stosunek ogólnej liczby wygenerowanych reguł do liczby reguł nieredukowalnych. Zastosowany mechanizm grupowania spowodował zdecydowaną redukcję zbioru wygenerowanych reguł (nawet 60-krotną). Redukcja ta jest tym większa im mniejsza jest wartość minimalnego poparcia i jest ściśle uzależniona od charakteru bazy danych. Można również postawić hipotezę, że wraz z obniżaniem progu minimalnego poparcia mechanizmy porządkowania wykazują wzrost efektywności.

Opisane powyżej eksperymenty są eksperymentami ilościowymi; potwierdziły, iż zaproponowany mechanizm przyczynił się do redukcji liczby wygenerowanych reguł.

Celem wprowadzenia wśród odkrytych reguł pewnego porządku było również osiągnięcie jak najbardziej zrozumiałego układu otrzymanej bazy reguł, tak by poprawiły się możliwości analizy i podniosła jakość usług systemu.

Kryterium to niestety nie jest mierzalne, a weryfikacja wymaga wykorzystania rzeczywistej bazy danych i dostępu do eksperta dziedzinowego.

Autorzy uczestniczyli w programie badawczym [7], w ramach którego stworzona została internetowa aplikacja do zbierania i gromadzenia danych o zakażeniach szpitalnych. Dane te gromadzone są systematycznie przez Polskie Towarzystwo Zakażeń Szpitalnych od 1997 roku z ponad stu szpitali w całej Polsce w bazie danych w Katedrze Mikrobiologii CM UJ. Dane te stanowią cenne źródło informacji na temat charakterystyki różnorodnych przypadków zakażeń szpitalnych.

Eksploracja danych została wykorzystana do wspomagania kontroli pojawiania się i rozprzestrzeniania drobnoustrojów opornych na leki przeciwdrobnoustrojowe w szpitalach jak również do analizy wpływu pewnych czynników etiologicznych, w tym przypadku bakterii na określone formy zakażenia szpitalnego.

Wyniki przedstawione zostały do weryfikacji lekarzom z Katedry Mikrobiologii CM UJ i wydają się potwierdzać celowość zastosowania eksploracji do danych medycznych. Zaproponowany mechanizm porządkowania zbioru odkrytych reguł wykorzystujący teorię FCA, w znaczący sposób ułatwia znajdowanie wśród nich reguł interesujących.

## Literatura

1. R. Agrawal, T. Imieliński, A. Swami. Mining association rules between sets of items in large databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data, May 1993.
2. R.J. Brachman, T. Anand. The process of knowledge discovery in databases. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*, pp 37-57. AAAI Mit Press, 1996.
3. B. Ganter, R. Wille. *Formal Concept Analysis: Mathematical Foundation*. Springer, Heidelberg, 1999.
4. W. Kloesgen. Efficient discovery of interesting statements in databases. *Journal of Intelligent Information Systems*, 4(1):53-69, 1995.
5. N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal. Efficient mining of association rules using closed itemset lattice. *Journal of Information Systems*, 24(1):25-46, 1999.
6. A. Silberschatz, A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*. 8(6):970-974, 1996.
7. R. Tadeusiewicz. Badania nad opracowaniem ogólnopolskiego modelu zapobiegania i zwalczania zakażeń szpitalnych. Raport 2 etapu realizacji projektu badawczego zamawianego, KBN nr 2198/PO5/98/13, AGH. 2000.
8. M.J. Zaki, H. Ching-Jui. Charm: An efficient algorithm for closed association rule mining. Technical report 99-10, RPI, 1999.
9. A. Zygmunt. Eksploracja danych metodą asocjacji (wybrane aspekty optymalizacji). Praca doktorska. AGH. Kraków, 2001.

Dr inż. Anna Zygmunt  
Katedra Informatyki AGH Kraków,  
Al. Mickiewicza 30  
e-mail: [azygmunt@agh.edu.pl](mailto:azygmunt@agh.edu.pl)

Dr inż. Marek A. Valenta  
Katedra Informatyki AGH Kraków,  
Al. Mickiewicza 30  
e-mail: [valenta@agh.edu.pl](mailto:valenta@agh.edu.pl)



## ROZDZIAŁ 2

# METODYKI WDROŻEŃ. NORMY I STANDARDY



# DOKUMENTOWANIE I SYMULOWANIE PROCESÓW BIZNESOWYCH — OD ISO 9000 DO SIX SIGMA

Piotr T. BIERNACKI

## Streszczenie:

Inicjatywy doskonalenia  
EPM – Co to ten EPM?  
Współczesny rynek  
Elementy jakości  
Koncepcyjny model Jakości  
Kąg EPM — Obszary wykorzystania  
Rynek EPM  
Cel prawidłowego oprogramowania EPM  
Optymalizacja a ISO 9000 — Stara a nowa norma ISO 9000  
Graficzna Księga Jakości  
Znaczenie optymalizacji tworzenia dokumentacji  
Six Sigma  
Narzędzia i ich użytkownicy

## Cytaty

„Świat zmieni się bardziej w przeciągu najbliższych 5 lat niż w ciągu ostatnich pięćdziesięciu ...”

*Bill Gates 1999*

„Procesy staną się centrum podstawą wszelkiej aktywności już w 2003 roku”

*Gartner*

## Inicjatywy doskonalenia

Inicjatywy doskonalenia produktów występowały już od dawna. Początkowo wiązano je z poprawą jakości produktu w oderwaniu od jakości produkcji. Miało to działać na zasadzie „będziemy robić tak jak robiliśmy tyle tylko, że bardziej się postaramy”. W latach 80. Przejawiało się to takimi inicjatywami jak: Przewodnictwo poprzez jakość, W Jakości, Zero braków, Wypełnianie wymogów klienta. Tak było do przełomu lat 80/90, gdy w pełni uzmysłowiono sobie, że jakość produktów zależy zawsze od jakości procesów wytwórczych.. Okazało się, że podnoszenie jakości procesów nie tylko podnosi jakość wyrobów, ale często również obniża koszty ich wytwarzania. Modnymi terminami tamtych czasów stały się: Zarządzanie procesami biznesowymi, Produktywność, Restrukturyzacja pod kątem zdejmowania kosztów, Benchmarking.

Dokładniejsze przyjrzenie się procesom biznesowym wykazało, że o ile stosunkowo łatwo zapewnić jakość procesów głównych, to z procesami



pomocniczymi różnie bywa. Trzeba było znaleźć lek na tą sytuację. Stał się nim Outsourcing. Dla firm wykonujących usługowo procesy pomocnicze dla innych firm procesy ich wykonywania stawały się procesami głównymi. Ale firmy aby móc ocenić, czy taka kooperacja ma szansę zaczęły wymagać jakiegoś potwierdzenia jakości działania. Powszechnie stała się nim norma ISO 9000. Koniec wieku to spostrzeżenie, że procesy w firmie muszą być dynamiczne, że trzeba je ciągle analizować i usprawniać nie tylko dlatego, że są niedoskonałe, ale również dlatego, że zmienia się otaczająca rzeczywistość. Efektami tych przemyśleń są m.in. ISO 9001:2000, Optymalizacja procesów, Six Sigma, Zarządzanie wiedzą, Zarządzanie Procesami w Przedsiębiorstwach (Organizacjach).

## **EPM – Co to ten EPM?**

EPM (Enterprise Process Management) łączy w sobie szereg metod i działalności mających na celu doskonalenie organizacji (przedsiębiorstwa). Wykorzystywane są w tym celu Schematy blokowe, diagramy przepływu, modele, modele systemowe.

Przykłady zastosowań

Dokumentacja procesów ISO-9001:2000,

BPR, Six Sigma, symulacje wykorzystania zasobów, czas, koszt, wąskie gardła

Rachunek kosztów działań (ABC) i zarządzanie kosztami działań (ABM)

E-process, CASE, EAI

EPM skupia się na:

- Na rozwiązaniach pomagających organizacji w poprawie efektywności
- Działaj tak abyś był lepszy, szybszy, tańszy.
- “Rozpocznij od zmapowania procesów, skontroluj je i modeluj, aż osiągniesz doskonałość Six Sigma”

## **Obserwacje**

Kluczowe warunki prowadzenia obecnego biznesu są znane czasem pod nazwą „Nowa Ekonomia”. Jej najważniejsze przesłania i warunki działania to:

- Klient jest naszym panem
- Internet
- Otwarcie się granic
- Deregulacja
- Witryny wymiany handlowej i aukcje internetowe

W wyniku tego Siła przesuwana się z dostawcy na klienta i konsumenta. To on wybiera i jemu wszystko jest podporządkowane. Jednocześnie występuje erozja:

- Tradycyjnych kanałów dostaw

- Trwałych bezpiecznych modeli biznesowych
- Marginesu zysku.

Sytuację można podsumować jednym zdaniem:

**WSZYSTKIE RODZAJE DZIAŁALNOŚCI SĄ POD NACISKIEM ABY DOSTARCZAĆ LEPIEJ — SZYBCIEJ — TANIEJ**

## Przykłady

Nowa ekonomia wywiera wpływ na wszelkie rodzaje działalności. Nie ma „bezpiecznych” obszarów biznesowych. Poniżej przykłady wpływu zasad na poszczególne sektory.

- Sektor publiczny analiza kosztów — konkurencja wewnętrzna czy outsourcing
- Sektor obrony — zastępowanie kontraktów z cenami indeksowanymi cenami stałymi
- Sektor farmaceutyczny – ceny wolnorynkowe w miejsce cen gwarantowanych i likwidacja ochrony ceny
- Przemysł samochodowy – kontrakty zakładające coroczną obniżkę kosztów
- Konsumenci – efektywne wyszukiwarki pozwalając na porównanie cen u różnych dostawców + obniżanie kosztów dostawy oraz jednolita platforma waluty — EURO

## Jakość a procesy

Znaczenie jakości procesów świetnie widać na przykładzie przemysłu samochodowego. Koncepcja auta o trwałości ponad 20. Letniej powstała w firmie Porsche w latach 70. Spotkała się ona z bardzo negatywną opinią sektora — jeśli auta będą tak trwałe to kot kupi nowy samochód, skoro stary ciągle jeździ. Jest to typowy sposób myślenia Rynku Producenta. Pierwsi znaczenie jakości w tej sferze jakości odkryli Japończycy. Na wiele lat samochody japońskie (a zwłaszcza Toyota) stały się niedościgłym wzorem jakości. Zawdzięczała to na równi mentalności japończyków jak i jakości procesów wytwórczych. Na przeciwnym krańcu stanął przemysł samochodowy w USA. Stało się to powodem jego ogromnych problemów. Lecz i tu z czasem (w tym także przy wydatnym udziale japończyków) powstały enklawy dobrej jakości. Np. Saturn ma jedną nierozwiązaną skargę na 14.000 sprzedanych samochodów a Ford — na 400. W efekcie takich różnic Ford w USA spadł na 28. pozycję pod względem satysfakcji klientów. Aby zapobiec utracie rynku przygotowywano się do pełnego zarządzania jakością zgodnie z metodologią Six Sigma. Ford Focus jest pierwszym autem, do którego projektowania i produkcji wykorzystano met. Six Sigma — efekt? Ford Focus jest na pierwszym autem pod względem niezawodności wg Dekry z przewagą nad drugą w tej klasyfikacji Toyotą taką samą jaką ona ma nad jedenastą marką na tym rynku.

## Z perspektywy posiadaczy akcji

Dokumentacja i analiza procesów ma mieć wpływ przede wszystkim na efektywność przedsiębiorstwa. Wynika ona przede wszystkim z:

- Nacisku inwestorów na zwrot nakładów
- Zmiany poglądów — od ok. 5 lat „Wiedza” przedsiębiorstwa traktowana jest jak składnik majątkowy

Pełne udokumentowanie procesów jest parametrem mierzenia Wiedzy w przedsiębiorstwie.

Tylko przedsiębiorstwa z najefektywniejszymi procesami przetrwają. Przy czym procesy w przedsiębiorstwach są definiowane od początku działalności aż po uzyskanie zainwestowanych pieniędzy. Działa to zgodnie z modelem:

Planowanie i Rozwój — Wdrożenie — Wytwarzanie — Dostarczanie — Sprzedaż — Serwis — Windykacja

Posiadanie wiedzy na temat procesów w przedsiębiorstwie powoduje, że przedsiębiorstwo potrafi wyodrębnić obszary w których posiada przewagę nad konkurencją i w ten sposób, dzięki większej marży podwyższać swój zysk. Jest też w stanie skupić się nad wyborem optymalnego klienta, bądź określić zasady postępowania z klientami generującymi straty. Wiedza ta pozwala również na zaobserwowanie dynamiki rynku i dopasowanie własnych działań do zmian z niej wynikających (wzrost, spadek popytu, spadek cen itp.)

Przykładowo w sektorze farmaceutycznym w ostatnich latach skrócono czas wprowadzenia leku na rynek z 15 do 5 lat w tym czasie zamyka się cykl: Wynalezienie — Testy kliniczne — Produkcja — Łańcuch dostaw — licencjonowanie itd.

## Sektor produkcyjny

Dokumentacja i analiza procesów biznesowych pozwala na zrozumienie „Kosztów Jakości” w tym znaczenia produktów niezgodnych ze specyfikacją. Taka niezgodność nie oznacza defektu "od razu", ale pozwala przewidzieć defekty później. Analiza wpływu na koszt wytwarzania defektów spowodowanych przez niezgodność komponentów ze specyfikacją składa się z 3 faz:

- Analiza Przyczyna-Skutek
- Rachunek Kosztów Działań (ABC — Activity-Based Cost Management)
- Doskonalenie procesów

Przykład. Naprawa gwarancyjna wadliwego komponentu kosztującego \$5 kosztuje \$250. Komponent kosztujący \$5,5 ulega uszkodzeniu w okresie gwarancyjnym ponad 1000 x rzadziej. Bez analizy procesu obsługi posprzedażnej (serwis) nie można ocenić wpływu podwyższenia kosztu produkcji na całkowite obniżenie kosztu wyrobu.



## Procesy

Pełna dokumentacja procesów w przedsiębiorstwie pozwala na odpowiedź na pytanie: Inteligencja procesu kontra automatyzacja?

Wiele podprocesów w przedsiębiorstwie można wystandaryzować i doprowadzić dzięki temu do wykorzystania najlepszych praktyk procesowych. Możliwe, że takie standardowe procedury nie są doskonałe, ale dzięki temu, że są powszechnie znane w przedsiębiorstwie:

- można to robić 10 x szybciej
- ludzkie umiejętności i procesy, w których występują pozostają niezmienione
- wydziały funkcjonalne zmieniają się tylko w małej skali

Podsumowując: Prawidłowa analiza pozwala na uzyskanie poprawy funkcjonowania najczęściej przy minimalnej ingerencji w funkcjonowanie przedsiębiorstwa.

„Należy najpierw skoncentrować się na zaprojektowaniu procesu, a następnie na dopasowaniu go potrzeb”

Lloyds TSB Group

Znaczenie prawidłowych procesów

„70% projektów informatycznych, które zakończyły się niepowodzeniem zakończyły się tak, ponieważ wadliwy był proces ich implementacji”

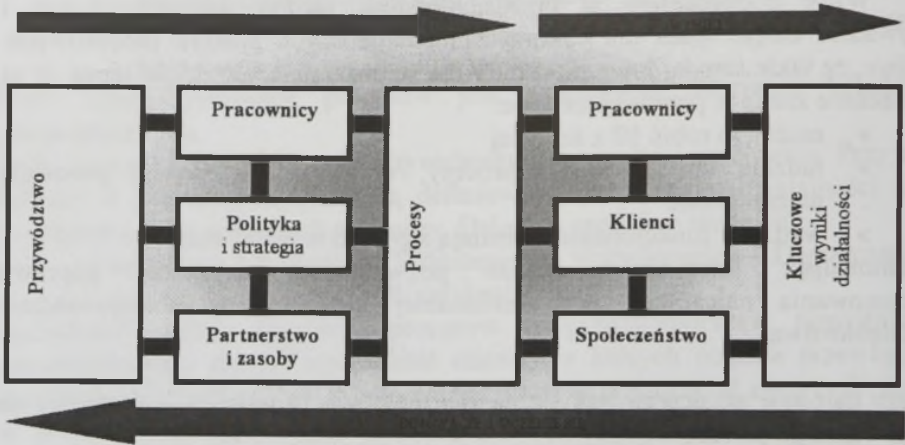
Gartner

## Elementy jakości

Gdy mówimy o jakości w produktach to należy brać pod uwagę następujące parametry:

- Serwis
- Czas cyklu
- Dostępność
- Niezawodność (MTBF)
- Odporność / długowieczność
- Obsługa
- Powtarzalność i sensowność procesów
- “Zgodność z oczekiwaniami “

## Koncepcyjny model Jakości



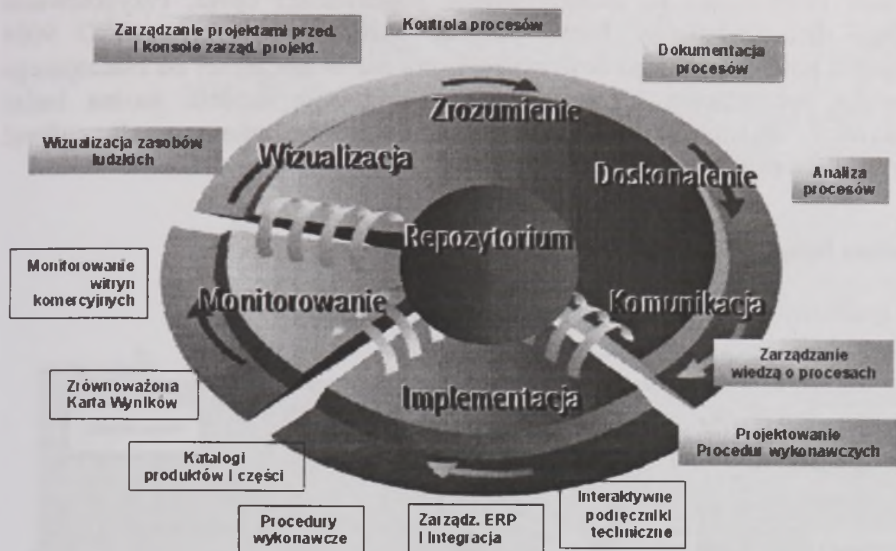
Przy takim podejściu do jakości można zadać sobie pytanie. Jaka restauracja daje najlepszą Jakość ?

- Wierzynek
- Belvedere
- Tsarevna
- Adria
- NIE
- McDonald's

Dlaczego?

Produkty McDonald'a zapewniają "Realizacja potrzeb". Co to znaczy? McDonald's wypełnia oczekiwania klientów, w każdym miejscu, w każdym czasie i na czas. Każdy, kto wchodzi do McDonald'a wie co uzyska. Jeśli mu potrawy nie smakują, to po prostu tam nie wchodzi. Obsłużony będzie na takich samych warunkach w dowolnym miejscu na świecie. Wie, jaki produkt otrzyma jego dziecko (Happy Meal z zabawką) a czego sam się może spodziewać dla siebie. Jak realizowana jest ta jakość? – poprzez realizację procesów McDonald'a

## Krąg EPM — Obszary wykorzystania



## Rynek EPM

Six Sigma

TQM (Total Quality Management)

ISO 9000:2000

Zarządzanie procesami biznesowymi (Business Process Management)

Zarządzanie wydajnością procesów

Rekonstrukcja procesów (BPR Business Process Reengineering)

Zarządzanie środkami

Zarządzanie ryzykiem

## Cel prawidłowego oprogramowania EPM

Dostarczenie jednolitego zestawu narzędzi, które pozwolą na wielokrotne wykorzystanie raz opracowanych dokumentów procesowych przy ewentualnym uzupełnieniu ich o specyficzne informacje potrzebne do realizacji danego zadania. Dzięki wykorzystywaniu narzędzi EPM wdrożenie ISO 9000 staje się środkiem, a nie celem opisu procesów. Celem jest doskonalenie się.

## Optymalizacja a ISO 9000 — Stara a nowa norma ISO 9000

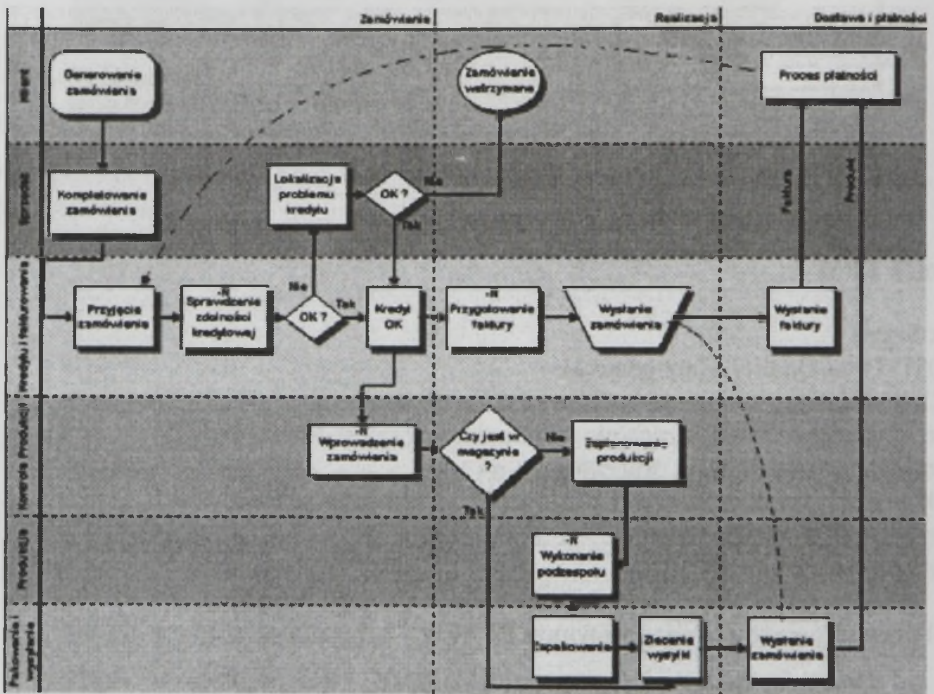
ISO 9001:2000 promuje rozwiązania procesowe. Nie wymaga formy graficznej opisu tych procesów, ale forma graficzna ich reprezentacji wyraźnie ułatwia ich dokumentowanie. (Patrz ISO 9001:2000 sekcja 4.2.3 *Kontrola dokumentów.*) Graficzna forma ułatwia kontrolę — widać powiązania pomiędzy



czynnościami i procesami. Często problemem przy wdrożeniach ISO 9000 jest wątpliwość, czy proces jest opisany poprawnie. Dlatego istotna jest możliwość symulacji, pozwalająca na skontrolowanie poprawności opisu. Przygotowanie symulacji daje dodatkowy bonus poprzez spełnienie dyrektywy ISO 9004 mówiącej o potrzebie stałego doskonalenia procesu w zależności od otaczającego środowiska prowadzenia biznesu. Poprzez symulację modelu można badać proponowane zmiany w procesach przed ich wdrożeniem, czyli uniknąć eksperymentowania na żywym organizmie..

### Graficzna Księga Jakości?

















Opisy graficzny i słowny tego samego procesu (przykład).



### Produkcja Oprogramowania

1. Klient zgłasza się z zamówieniem (Sposób generowania transakcji opisany jest w załączniku nr 1)
2. Dział sprzedaży kompletuje zamówienie, po określeniu jego treści przekazuje do działu kredytów i fakturowania (zasoby, czas trwania czynności określone są w zał. Nr 2)
3. Dział kredytów i fakturowania przyjmuje zamówienie i wprowadza je do systemu komputerowego (zasoby, czas trwania czynności określone są w zał. Nr 2)

4. Następnie sprawdza "zdolność kredytową" odbiorcy. (zasoby, czas trwania czynności określone są w zał. Nr 2)
  - a. Jeśli dział KiF nie może określić, czy klient jest wiarygodny dokumenty przekazywane są do działu Sprzedaży, który sprawdza wiarygodność klienta (zasoby, czas trwania czynności określone są w zał. Nr 2)
    - i. Jeśli weryfikacja jest negatywna zamówienie jest wstrzymywane a klient jest informowany o wstrzymaniu realizacji zamówienia i jego przyczynach.
    - ii. Jeśli weryfikacja jest pozytywna dokumenty z odpowiednią adnotacją wracają do działu kredytu i fakturowania (stosunek pozytywnych do negatywnych weryfikacji określa zał. Nr 3)
  - b. Jeśli klient jest wiarygodny przechodzi się do fazy realizacji. Przebiega ona dwutorowo.
    - i. Produkcja i wysyłka produktu
    - ii. Przygotowanie i wysyłka faktury
5. Produkcja i wysyłka produktu
  - a. Dział kontroli produkcji wprowadza zamówienie do realizacji (zasoby, czas trwania czynności określone są w zał. Nr 2)
  - b. Następnie sprawdza, czy zamówiony produkt jest w magazynie (statystyczny rozkład występowania sytuacji, gdy produkt jest w magazynie określa zał. Nr 3)
    - i. Jeśli produktu nie ma, planowana jest produkcja
      1. Zamówienie przekazywane jest do działu produkcji gdzie wykonuje się produkt. (precyzyjna instrukcja produkcji opisana jest dokumentem "Produkcja wyrobu" stanowiącym zał. Nr 4)
      2. Gotowy produkt przekazuje się do działu pakowania i wysyłki, gdzie jest pakowany (zasoby, czas trwania czynności określone są w zał. Nr 2)
    - ii. Po wyprodukowaniu produktu lub jeśli produkt jest w magazynie przygotowuje się Zlecenie wysyłki (zasoby, czas trwania czynności określone są w zał. Nr 2)
    - iii. Produkt wysyłany jest do klienta.
6. Przygotowanie i wysyłka faktury
  - a. Dział kredytu i fakturowania przygotowuje fakturę (zasoby, czas trwania czynności określone są w zał. Nr 2)
  - b. Po otrzymaniu informacji z działu pakowania i wysyłki, że towar jest wysyłany, drukowana i wysyłana jest faktura. (zasoby, czas trwania czynności określone są w zał. Nr 2)
7. Sposób płacenia przez klienta należy uwzględnić przy ocenie zdolności kredytowej.

Schematy blokowe kontra tekst		
	Tekst	FlowChart
Przejrzystość		
Zachęta do czytania		
Łatwość zrozumienia		
Łatwość aktualizacji		
Przydatność do szkoleń		
Spójność		
Łatwość tworzenia		
Pokazywanie relacji		

## Znaczenie optymalizacji tworzenia dokumentacji

Powszechnie w przedsiębiorstwach tworzenie dokumentacji produkcyjnej



**39 % czasu pracy**  
**Tworzenie dokumentacji**

Źródło: Irwin - Dunn & Bradstreet,  
Survey of 1900 registered firms. '96

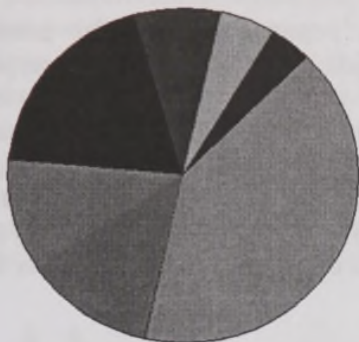
odbierane jest jako bariera działania lub jako zło konieczne. Graficzne metody dokumentacji pozwalają skrócić czas tworzenia dokumentacji od 10 do 30%. (wg opinii konsultantów współpracujących z Corel Corp.) Jeszcze wyraźniejsze jest to przy tworzeniu dokumentacji na potrzeby ISO 9000, gdzie procesy i uwarunkowania pomiędzy nimi muszą być precyzyjnie pokazane. Widać to b. wyraźnie przy wdrożeniach ISO gdzie stworzenie odpowiedniej dokumentacji jest głównym czynnikiem odpowiadającym za czas trwania certyfikacji.

Graficzne metody dokumentacji pozwalają skrócić czas tworzenia dokumentacji ISO 9001:2000 od 20 do 40%

(wg opinii konsultantów współpracujących z Corel Corp.)



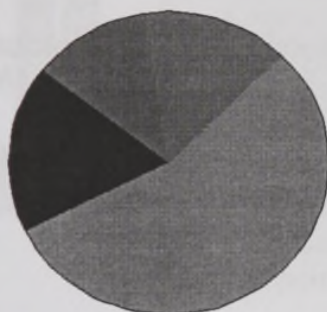
## Proces certyfikacji ISO 9000, jak rozkładał się czas pracy?



**41% czasu pracy**  
**Dokumentacja procedur**

Source: Irwin - Dunn &  
Bradstreet, Survey of 1900  
registered firms, '96

Wdrożenie to jedno a utrzymanie normy to drugie. Wymagana jest odpowiednia opieka nad dokumentami, zwłaszcza ich ciągła aktualizacja wynikająca ze zmian w istniejących procesach. Graficzne metody dokumentacji pozwalają skrócić czas opieki nad dokumentami ISO 9001:2000 od 30 do 60% (wg opinii konsultantów współpracujących z Corel Corp.)



**55 %**  
**Zapewnianie aktualności**  
**dokumentów**

Source: Irwin - Dunn &  
Bradstreet, Survey of 1900  
registered firms, '96

## Six Sigma

- Metodologia poprawy jakości zogniskowana na poprawie zadowolenia klienta poprzez wirtualne eliminowanie defektów procesów lub produktów, która w efekcie prowadzi do poprawy podstaw działalności
- W przypadku tworzenia dokumentacji narzędziami iGrafx można wykorzystywać dokumenty iGrafx przygotowane na potrzeby ISO 9000
- (Six Sigma) "To proces biznesowy, który pozwala przedsiębiorstwom na drastyczną poprawę podstaw działalności przez projektowanie i monitorowanie codziennych aktywności w sposób, który minimalizuje straty i wykorzystanie zasobów i jednocześnie podnosi zadowolenie klientów" *Mikel Harry*
- (Six Sigma to) "Zdyscyplinowane zastosowanie narzędzi statystycznych rozwiązujących problemy związane z niepotrzebnymi kosztami i precyzyjnie kierujące ku poprawie" *Greg Brue*

## Zasady pracy w Six Sigma

Filozofia Six Sigma zakłada poprawę rentowności przedsiębiorstwa poprzez doskonalenie procesów. Niejako "ubocznym" skutkiem jest poprawa jakości wyrobów. Ten "uboczny" skutek jest miernikiem doskonałości przedsiębiorstwa. Nazwa metodologii została zaczerpnięta z nazwy miary odchylenia standardowego. 6 sigma to 3,4 braku na 1.000.000 operacji. Podstawą metodologii Six Sigma jest cykl: Definiuj (cel usprawnienia) — Mierz (istniejący system) — Analizuj (system lub proces) — Usprawniaj — Kontroluj (poprawiony system lub proces). Poniżej typowy cykl działania w metodologii Six Sigma:

### Definicja

- Określenie celu usprawnienia

### Pomiar

- Mapowanie Procesu
  - Odkrywanie ukrytych współczynników
  - Wybór charakterystyk CTQ
- “Już proces mapowania procesów prowadzi do ich doskonalenia.”  
*Mikel Harry, Ph.D.*

### Analiza

- Porównanie procesów
- Definiowanie celów do poprawy
- ANOVA
- Zmiana Produktów / Procesów • Badanie Hipotez
- Identyfikacja/Uwierzytelnienie /Priorytet Zmiennych

### Doskonalenie

- Eksperymenty przez symulacje
- DOE (budowa eksperymentów)
  - Pojedynczy współczynnik
  - Podwójny współczynnik
  - Wiele współczynników
  - Ułamki

### Kontrola

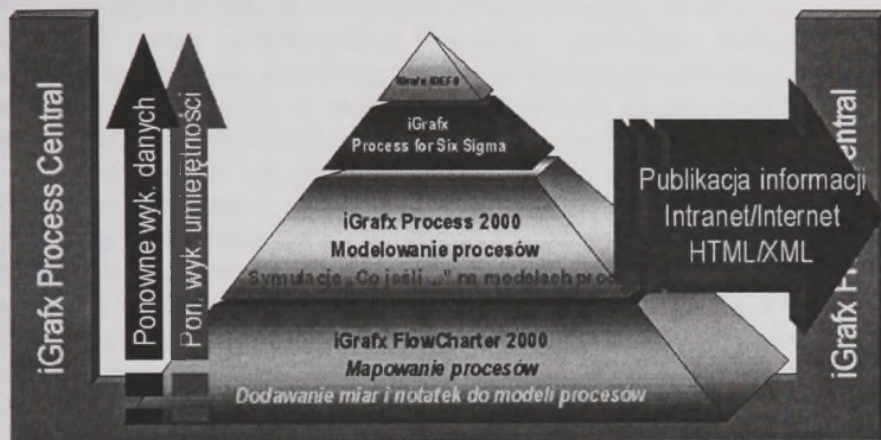
- Mapowanie Poprawionych Procesów
- Diagramy SPC
- Komunikowanie się
- Monitorowanie Efektywności Procesów

### Narzędzia i ich użytkownicy

Narzędzia iGrafx są typowym przykładem narzędzi EPM. Pozwalają na

graficzne wspomaganie tworzenia systemu zintegrowanego zarządzania poprzez jakość. W przedsiębiorstwie wykorzystywane są w różnych obszarach przez różnych użytkowników. Najprostszą formą jest oglądanie obowiązujących procedur. Mogą być one publikowane w Intranecie lub w formach papierowych. Praktycznie każdy uprawniony do zaznajamiania się z procedurami może z nich w ten sposób korzystać. Średni szczebel zarządzający wykorzystuje narzędzia do dokumentacji procedur w obszarach mu podległych. Modelowanie i analiza procesów to domena działów rozwoju i kontroli produkcji. Zarządzaniem dokumentami zajmują się pionierzy IT i/lub pionierzy pełnomocnika ds. jakości

Przykładowe rozwiązanie przy wykorzystaniu narzędzi iGrafx



## Literatura

1. Mirosława Lasek, Marek Pęczkowski, Bartosz Otmianowski, Analiza Procesów Biznesowych z Wykorzystaniem Programów iGrafx Process 2000 for Six Sigma / iGrafx FlowCharter 2000 Professional PL, WSISiZ, Warszawa 2002
2. Mikel Harry, Richard Schroeder, Six Sigma, Oficyna Ekonomiczna, Kraków 2001
3. Brian M. McCarthy — Materiały Szkoleniowe CoreEPM

mgr inż. Piotr T. Biernacki

CoreEPM Representative, czł. Zarz. PRO i KA PliiT, wykł.. PJWSTK

ul. Wilanowska 14a/10, 00-422 Warszawa

tel.:(0-22) 71-111-71

e-mail: piotrek@pol.pl





# STUDIUM PRZEDSIĘWZIĘCIA DLA PROJEKTU INFORMATYCZNEGO DLA POTRZEB ADMINISTRACJI PUBLICZNEJ – PROPOZYCJA STANDARDU

Edward GAWĘŁ

## Wstęp

Każda ludzka działalność, niezależnie od jej charakteru i stopnia złożoności, jest wsparta szeregiem reguł, zasad i metod postępowania, wypracowanych w oparciu o wieloletnie doświadczenia i obserwacje. Trudno sobie wyobrazić pracę współczesnego dentysty, architekta, budowniczego czy nauczyciela, który nie przestrzega reguł obowiązujących w branży. Każdy z nich ma swoje reguły, które w zależności od sytuacji, czasem stosuje wiernie, ale najczęściej przystosowuje do konkretnej, zastanej sytuacji.

Tak też jest i w przypadku prac dotyczących systemów informacyjnych wspieranych metodami informatycznymi. Można bez przesady powiedzieć, że „grzebanie się” w systemie informacyjnym jakiegokolwiek organizacji jest bardziej złożone niż leczenie zęba, budowa mostu czy montaż samochodu. Tam bowiem mamy do czynienia z obiektami konkretnymi, które można z każdej strony obejrzeć, z obiektami, które chociaż czasem są bardzo skomplikowane (komputer), to dadzą się zwymiarować w czasie i przestrzeni. W ostateczności, każdy z takich obiektów, po nieudanych na nich zabiegach, można wyrzucić. Trochę to kosztowne (samochód) albo bolesne (zęb) – ale możliwe.

Czy można zwymiarować system informacyjny organizacji? W czasie – nie, ponieważ procesy na ten system się składające są powtarzalne od początku istnienia organizacji aż do jej „śmierci”. W przestrzeni – też nie, bo system nie jest ograniczony żadnymi wymiarami, płotami, granicami – jest i jakby go nie było. A jeśli się twórcom nie uda, to nie da się wyrzucić – chyba, że razem z organizacją, którą obsługuje.

Implikacje z tego wynikające są oczywiste. Zabiegi nad systemem informacyjnym nie mogą być prowadzone „byle jak”. Muszą być oparte na jak najdokładniejszym rozpoznaniu organizacji, jej celów i uwarunkowań, jej szans i zagrożeń, jej potrzeb i możliwości. Muszą być oparte o wiedzę i doświadczenie. Muszą być w końcu oparte o jakieś konkretne wskazania co do kolejności i charakteru poszczególnych kroków i zadań składających się na efekt końcowy. Efekt końcowy to, z punktu widzenia modelowanej organizacji, model sprawnego i skutecznego systemu informacyjnego, wspartego metodami informatycznymi a z punktu widzenia zespołu analitycznego, to przedsięwzięcie zrealizowane w przewidzianym dla niego czasie z przewidzianym dla niego budżetem. Wymienione wsparcie informatyczne ma być realizowane w oparciu o wyniki analizy, która stanowi *specyfikację wymagań* modelowanej organizacji.

Trzeba niestety zauważyć, że praktyka przygotowywania przedsięwzięć informatycznych jest inna. Ta inność polega na tym, że albo przedsięwzięcia takie

nie są poprzedzone analizą funkcjonującego systemu informacyjnego albo analiza taka robiona jest szybko, pobieżnie i jest traktowana jako mało ważny etap wstępny, „niepotrzebnie” opóźniający moment rozpoczęcia właściwych implementacji informatycznych. Taką opinię można również znaleźć w opracowaniu pt. „*Inżynieria oprogramowania w projekcie informatycznym*” – pod redakcją Janusza Górskiego – wydanie II rozszerzone. Na stronie 54 jest tam napisane:

...*W większości organizacji związanych z budową i eksploatacją systemów informatycznych jakość procesów IW (czyt. Inżynierii Wymagań) jest niska, ponieważ pozyskiwania wymagań, podobnie jak całościowego opracowania SWS (czyt. Specyfikacja Wymagań Systemowych), nie traktuje się dostatecznie poważnie. ... Najczęściej uważa się, że przez IW należy przejść możliwie jak najszybciej, by móc zacząć „prawdziwą pracę”...*

Departament Rozwoju Informatyki i Systemu Rejestrów Państwowych MSWiA podjął próbę uporządkowania czynności związanych z inicjowaniem przedsięwzięć projektowych z obszaru IT w administracji rządowej w postaci *Zaleceń* pt. *Wniosek o podjęcie przedsięwzięcia informatycznego* – traktując tę próbę jako pierwszy etap porządkowania czynności związanych z pełnym cyklem życia systemu informatycznego wspomagającego system informacyjny organizacji. W sformułowaniu *Zaleceń* przyjęte zostały następujące zasady:

1. Potrzeba opracowania i wdrożenia systemu informatycznego powinna wynikać z posiadanego strategicznego planu rozwoju informatyki i powinna być poprzedzona analizą właściwego systemu informacyjnego.
2. Podstawą do podejmowania jakichkolwiek czynności realizacyjnych powinien być zatwierdzony *Wniosek* o rozpoczęcie projektu systemu informatycznego, sformułowany w rezultacie wykonanego studium przedsięwzięcia. W tym etapie następuje sformułowanie problemu użytkowego siłami i środkami własnymi przyszłego użytkownika systemu, umożliwiające:
  - 1) podjęcie wszelkich decyzji związanych z przydziałem lub zaplanowaniem zasobów: organizacyjnych, personalnych, finansowych, materialnych itp.,
  - 2) sformułowanie specyfikacji istotnych warunków zamówienia, utworzenie struktur organizacyjnych i podjęcie innych niezbędnych działań zmierzających do rozpoczęcia projektowania lub zakupu gotowego systemu informatycznego.

Po akceptacji przedmiotowego *Wniosku* mogą rozpocząć się dalsze prace studialne i analizy, takie, jak: szczegółowe zdefiniowanie problemu użytkowego, analiza wymagań, analiza systemowa itd. oraz prace projektowe z ewentualnym zaangażowaniem wyspecjalizowanych podmiotów profesjonalnych.

Niniejsze opracowanie zawiera charakterystykę *Zaleceń* w:

- części proceduralnej – czyli jaka powinna być droga *Wniosku*? – rozdział 2



- części dokumentacyjnej – co powinien zawierać *Wniosek*? – rozdział 3

Treść rozdziałów 2 i 3 jest wyciągiem z opracowanych w Departamencie *Zaleceń*.

## **1. Część proceduralna Zaleceń**

### **1.1. Organizacja sporządzania Wniosku.**

Zaleca się, aby wniosek przygotował zespół powołany do jego opracowania, zwany dalej Zespołem.

Zalecane jest włączenie do składu Zespołu głównych użytkowników opracowywanego systemu informatycznego oraz przedstawiciela służby informatycznej z obszaru systemu lub jednostki organizacyjnej zarządzającej tym obszarem.

Zespół powołuje osoba zarządzająca obszarem systemu lub osoba pełnomocna, przedstawiając zespołowi wymagania ważne z punktu widzenia kierownictwa określając jednocześnie sponsora przedsięwzięcia oraz kierownika projektu kierującego zespołem. Zaleca się, aby sponsorem przedsięwzięcia była osoba ze ścisłego kierownictwa obszaru systemu a kierownikiem projektu osoba reprezentująca głównego użytkownika.

Zespół kończy pracę z chwila zatwierdzenia Wniosku lub jego odrzucenia.

### **1.2. Sporządzanie Wniosku.**

Podczas sporządzania Wniosku należy przestrzegać następujących zasad wykonywania analiz i badań:

- zasady kompletności - uwzględnienia wszystkich istotnych elementów i cech przedsięwzięcia,
- zasady ogólności - pominięcia szczegółów, które będą przedmiotem dalszych etapów opracowania systemu informatycznego.

Zaleca się wykorzystanie następujących metod (bądź ich kombinacji) do przeprowadzenia studium przedsięwzięcia:

- wywiady z członkami kierownictwa,
- wywiady z użytkownikami i klientami systemu,
- ankiety,
- studiowanie dokumentów formalno-prawnych, opisów i instrukcji,
- obserwacji funkcjonowania istniejącego systemu.

Zaleca się, aby Wniosek dotyczył *systemu informatycznego* stanowiącego element strategicznego planu rozwoju informatyki jednostki sporządzającej Wniosek.

### **1.3. Opiniowanie Wniosku.**

#### **1.3.1. Sposób opiniowania Wniosku.**

Zaleca się opiniowanie Wniosku przez wytypowanych użytkowników i klientów opracowywanego systemu informatycznego, których wykaz zatwierdza osoba powołująca Zespół. Opiniowanie Wniosku organizuje osoba kierująca Zespołem.

W szczególnym przypadku, gdy planowane przedsięwzięcie jest bardzo złożone lub ważne, zaleca się, aby Wniosek był opiniowany przez niezależny podmiot doradczy.

Wraz z dokumentacją Wniosku przekazywanego do zaopiniowania zaleca się określenie:

- oczekiwanej formy zgłaszania uwag,
- terminu przekazania uwag,
- terminu spotkania uzgadniającego.

#### **1.3.2. Spotkanie uzgadniające.**

Uczestnikami spotkania uzgadniającego powinni być członkowie Zespołu oraz osoby reprezentujące zainteresowane podmioty opiniujące Wniosek.

Spotkanie prowadzi osoba kierująca Zespołem. Zaleca się następujący przebieg spotkania:

- przedstawienie wszystkich zgłoszonych do Wniosku uwag,
- przedstawienie przez Zespół oceny uwag poprzez wskazanie:
  - uwag, które powinny stanowić podstawę do zmian we Wniosku,
  - uwag, które nie powinny być uwzględnione z uzasadnieniem takiej oceny,
- dyskusja,
- sporządzenie oraz podpisanie protokołu uzgodnień i rozbieżności.

#### **1.3.3. Poprawianie Wniosku.**

Poprawienie Wniosku polega na wprowadzeniu zmian wynikających z przyjętego protokołu uzgodnień.

W przypadku, gdy zapisy protokołu dotyczą spraw zasadniczych, może być konieczne ponowne przeprowadzenie analiz i badań (rozdział 4.2) i powtórzenie opiniowania.

### **1.4. Zatwierdzanie wniosku.**

Wniosek zatwierdza osoba powołująca Zespół, o której mowa w rozdziale 4.1. Przed podjęciem ostatecznej decyzji osoba ta powinna dokonać ostatecznych rozstrzygnięć wynikających z rozbieżności zawartych w protokole spotkania uzgadniającego.

Możliwe są następujące przypadki:

- zatwierdzenie Wniosku umożliwiające przejście do *opracowania systemu informatycznego*, w szczególności wykonanie działań, o których mowa w rozdziale 9 Wniosku (*zorganizowanie zarządzania przedsięwzięciem*),
- skierowanie Wniosku do poprawy przez Zespół,
- niezatwierdzenie Wniosku, co oznacza zakończenie prac zmierzających do *opracowania systemu informatycznego* w przedstawionym kształcie.

Niezatwierdzenie Wniosku powinno zostać uzasadnione.

### 1.5. Koordynacja Przedsięwzięcia

W celu spełnienia koordynacyjnej roli wytypowanego Urzędu Centralnego (zwanego dalej Urzędem), wyposażonego w ustawowe kompetencje do koordynowania rozwoju informatyki w administracji rządowej, zaleca się, aby osoba kierująca Zespołem, za zgodą osoby powołującej, przesała Wniosek do zaopiniowania przez Urząd, zwłaszcza wtedy, gdy:

- a) przewidywany szacunek nakładów rzeczowych przekracza kwotę określoną w art. 15 ust. 1 ustawy z dnia 10 czerwca 1994 r. o zamówieniach publicznych (Dz.U. Nr 76 poz. 344 z późn. zm.),
- b) przewidywany okres *opracowania systemu* przekracza 12 miesięcy,
- c) ze względu na złożoność systemu przewiduje się rozbudowaną organizację *zarządzania przedsięwzięciem*,
- d) można przypuszczać, że podobny *system informatyczny* został już opracowany w innych instytucjach i jest możliwa, po dokonaniu odpowiednich zmian, *implementacja systemu w obszarze systemu*,
- e) prawidłowe *opracowanie systemu* lub jego eksploatacja może zależeć od istnienia albo potrzeby modyfikacji innych *systemów informatycznych* w *otoczeniu obszaru* lub od terminów *implementacji* tych systemów,
- f) można przypuszczać, że w *otoczeniu obszaru systemu* są stosowane inne, niż przedstawione we Wniosku, standardy technologiczne, mogące utrudnić planowane przekazywanie strumieni danych, a także obsługę i serwis,
- g) można przypuszczać, że dla prawidłowego funkcjonowania *otoczenia obszaru systemu* uzasadnione jest rozszerzenie zakresu realizowanych procedur lub danych w planowanym systemie.
- h) Wniosek powinien być przesłany w momencie rozpoczęcia opiniowania, z podaniem przewidywanego terminu jego zatwierdzenia.

Urząd może w ciągu 30 dni od daty otrzymania Wniosku zgłosić uwagi dotyczące:

- a) zasadności *przedsięwzięcia projektowego*,
- b) konieczności integracji projektowanego *systemu informatycznego* z innymi *systemami informatycznymi* administracji rządowej, a w szczególności z rejestrami centralnymi,
- c) prawidłowości przyjętych założeń i proponowanego sposobu realizacji *przedsięwzięcia projektowego* w celu zmniejszenia ryzyka niepowodzenia,

Osoba kierująca Zespołem jest zobowiązana do zajęcia stanowiska odnośnie uwag Urzędu przed przekazaniem Wniosku do zatwierdzenia.



Brak odpowiedzi ze strony Urzędu w podanym wyżej terminie jest jednoznaczna z brakiem uwag

## 2. Część dokumentacyjna Zaleceń.

Dokumentacja *Wniosku* powinna składać się z następujących części:

### 2.1. Wstęp

Wstęp powinien zawierać:

- definicję *przedsięwzięcia projektowego*,
- podstawę podjęcia prac:
  - formalną  
należy wymienić podstawy formalne takie, jak: strategiczny plan rozwoju informatyki, pozycję planu pracy obowiązującego w *obszarze systemu*, polecenie, decyzję lub podobny akt formalno-prawny właściwego podmiotu decyzyjnego, ewentualnie inny powód formalny, np. zalecenia pokontrolne,
  - merytoryczną  
należy wyszczególnić: główne unormowania formalno-prawne aktualnie obowiązujące w *obszarze systemu*, determinujące jego kształt, oczekujące na wprowadzenie, nowe unormowania bądź zmiany dotychczasowych unormowań oraz inne uwarunkowania, np. opinie użytkowników lub klientów systemu,
- ewentualną historię przedsięwzięcia projektowego.

### 2.2. Cel przedsięwzięcia

Powinien być wyraźnie sformułowany cel jaki *użytkownik zamierza uzyskać* po wdrożeniu *systemu informatycznego* - cel strategiczny.

Powinny być zdefiniowane cele taktyczne, których realizacja składa się na cel strategiczny.

Należy dążyć do definiowania celów w postaci mierzalnej, aby była możliwa jednoznaczna ocena ich osiągnięcia.

Zestaw celów powinien stanowić podstawę do specyfikacji *wymagań użytkownika*.

### 2.3. Stan aktualny

#### 2.3.1. Struktura organizacyjna systemu

Struktura powinna mieć postać schematu organizacyjnego jednostek należących do *obszaru systemu* wybranych z aktualnych schematów organizacyjnych instytucji. Każda jednostka organizacyjna powinna zostać scharakteryzowana zakresem swoich kompetencji i obowiązków realizowanych

w *obszarze systemu*. Jednostki organizacyjne o identycznym zakresie kompetencji i obowiązków w *obszarze systemu* mogą być połączone w grupy jednostek.  
(patrz: załącznik 3.1.1)

### **2.3.2. Realizowane procedury**

Ta część opracowania powinna zawierać specyfikację głównych procedur realizowanych w *obszarze systemu* (bez specyfikacji czynności składających się na te procedury).

[Przez pojęcie procedury należy rozumieć ciąg podejmowanych decyzji (kompetencje) i realizowanych zadań (obowiązki) przez różne jednostki organizacyjne *obszaru systemu* dla realizacji istotnej dla tego *obszaru* funkcji.]  
(patrz: załącznik 3.1.2)

### **2.3.3. Struktura raportowania w obszarze systemu**

Należy przedstawić specyfikację sporządzanych raportów (klas raportów).  
[Przez pojęcie raportu należy rozumieć *informacje* przygotowywane zarówno dla jednostek organizacyjnych *obszaru systemu* jak i *otoczenia tego obszaru*.]  
(patrz: załącznik 3.1.3)

### **2.3.4. Charakterystyka powiązań z otoczeniem obszaru systemu**

Należy podać informacje charakteryzujące strumienie danych przekazywane między *obszarem systemu* i *otoczeniem obszaru systemu* oraz podmioty uczestniczące w procesie przekazywania.  
(patrz: załącznik 3.2)

### **2.3.5. Charakterystyka głównych zasobów danych**

Należy podać specyfikację podstawowych kartotek lub baz danych utrzymywanych w *obszarze systemu*.  
(patrz: załącznik 3.3)

### **2.3.6. Charakterystyka głównych strumieni danych wejściowych**

Należy przedstawić specyfikację głównych danych wejściowych (strumieni danych wejściowych).  
(patrz: załącznik 3.4)

### **2.3.7. Charakterystyka systemów informatycznych w obszarze systemu**

Należy sporządzić zestawienie *systemów informatycznych* eksploatowanych w *obszarze systemu*.  
(patrz: załącznik 3.5)

## **2.4. Ocena stanu aktualnego**

Trześcią rozdziału jest podsumowanie opisu stanu aktualnego zawierające ocenę realizacji podstawowych procedur w *obszarze systemu* z punktu widzenia kierownictwa, *użytkowników* i *klientów* w aspekcie:

- efektywności organizacyjnej i ekonomicznej,
- skuteczności,
- poprawności merytorycznej,
- niezawodności,
- bezpieczeństwa zasobów informacyjnych
- ergonomiczności.

## 2.5. Stan docelowy

Treść rozdziału 3.5. należy przygotować według zaleceń dla rozdziału 3.3, umieszczając opisy elementów proponowanego *systemu informatycznego*. Dodatkowo, w formie komentarza należy wskazać różnice w stosunku do stanu aktualnego.

### 2.5.1. Struktura organizacyjna systemu

Jak w rozdziale 3.1.1 ze wskazaniem różnic.

### 2.5.2. Proponowane procedury

Jak w rozdziale 3.1.2 ze wskazaniem różnic.

### 2.5.3. Struktura raportowania w *obszarze systemu*

Jak w rozdziale 3.1.3 ze wskazaniem różnic.

### 2.5.4. Charakterystyka powiązań z otoczeniem obszaru systemu

Jak w rozdziale 3.2 ze wskazaniem różnic.

### 2.5.5. Charakterystyka głównych zasobów danych

Jak w rozdziale 3.3 ze wskazaniem różnic.

### 2.5.6. Charakterystyka głównych strumieni danych wejściowych

Jak w rozdziale 3.4 ze wskazaniem różnic.

### 2.5.7. Charakterystyka *systemów informatycznych w obszarze systemu*

Jak w rozdziale 3.5 ze wskazaniem różnic.

## 2.6. Charakterystyka docelowego *systemu informatycznego*

Charakterystyka powinna zawierać specyfikację *wymagań* wynikających z potrzeb *użytkownika i klientów* dotyczącą:

- sprzętu komputerowego i urządzeń peryferyjnych,
- infrastruktury sieciowej,
- systemów telekomunikacyjnych,
- *oprogramowania systemowego, narzędziowego i użytkowego,*
- bezpieczeństwa *przetwarzania danych.*



## 2.7. Oczekiwane efekty

Należy wskazać spodziewane efekty z podziałem na dwa rodzaje:

- wymierne, tzn. takie, które dadzą się wyrazić w jednostkach miary,
- niewymierne, tzn. takie, które poprawiają stan zadowolenia *użytkowników i klientów systemu* poprzez wyeliminowanie uciążliwości i błędów wymienionych w opisie stanu obecnego lub stworzenie nowych możliwości.

## 2.8. Szacunek potrzebnych nakładów

Planowane nakłady w *cyklu życia systemu*:

- na *opracowanie systemu*

należy przedstawić oszacowanie sumaryczne, uwzględniając, o ile to możliwe, nakłady:

- osobowe (zmiana zakresu dotychczas wykonywanych zadań lub kwalifikacje aktualnie zatrudnionej kadry mogą wymagać przesunięć osobowych lub zatrudnienia pracowników),
- na sprzęt informatyczny: serwery, stacje robocze, urządzenia peryferyjne i inny sprzęt,
- na wyposażenie sieciowe: urządzenia, okablowanie oprogramowanie, inne,
- na zakup *oprogramowania systemowego, narzędziowego, użytkowego* i/lub na adaptację takiego oprogramowania dla potrzeb systemu
- na usługi związane z realizacją przedsięwzięcia: instalacje, konsultacje, opinie itd.,
- na szkolenia dla kadry kierowniczej, *użytkowników, administratorów* itd.,
- inne nakłady na *opracowanie systemu* (m. in. koszty materiałowe związane z wyposażeniem stanowisk pracy nie tylko informatycznym).

W miarę możliwości powyższe nakłady należy podzielić na przewidywane etapy prac.

Dla bardziej precyzyjnego określenia potrzebnych nakładów, które są na wprost pochodną stopnia złożoności systemu, można posłużyć się jedną z metod oceny takiej złożoności, np. metodę Punktów Funkcyjnych lub metodę COCOMO.

- na eksploatację *systemu informatycznego*

należy przedstawić, w miarę możliwości, oszacowanie roczne, uwzględniając nakłady:

- osobowe związane z eksploatacją systemu: administrowanie i inne, w tym szkolenie nowych *użytkowników*,

- amortyzację (umorzenia wartości) środków trwałych,
- na obsługę systemu i serwis,
- koszt materiałów eksploatacyjnych,
- inne, np. opłaty za dzierżawienie łączy telekomunikacyjnych.

Wniosek powinien wskazywać na źródła finansowania *przedsięwzięcia projektowego*.

## 2.9. Przewidywane terminy

### 2.9.1. Termin opracowania systemu

Ze względu na fakt, że szczegółowe plany *opracowania systemu* powstaną w późniejszym czasie, należy określić przewidywany termin *implementacji systemu*. Często termin ten wynika z określonych przepisów formalno-prawnych lub aktualnej sytuacji.

Jeśli jednak został określony wstępny podział na etapy prac, należy przedstawić orientacyjny harmonogram. W każdym przypadku należy ponadto przedstawić plan działań na najbliższy okres.

### 2.9.2. Termin zakończenia eksploatacji lub modernizacji systemu

Należy podać przewidywany okres eksploatacji systemu do czasu spodziewanych większych modernizacji lub zaprzestania korzystania z systemu.

## 2.10. Analiza uwarunkowań i zagrożeń

Analiza powinna być wzorowana na metodzie SWOT i powinna uwzględniać następujące czynniki:

- wewnętrzne zalety i słabości *obszaru systemu*
- zewnętrzne szanse i zagrożenia *obszaru systemu*, czyli wpływ na *obszar systemu* jego otoczenia,

ze wskazaniem najistotniejszych czynników krytycznych sukcesu przedsięwzięcia.

Wyniki analizy powinny zostać spożytkowane do określenia, w jaki sposób *system informatyczny* może:

- przeciwdziałać wewnętrznym słabościom i zewnętrznym zagrożeniom,
- wykorzystać wewnętrzne zalety i zewnętrzne szanse

Analiza powinna również pozwolić na oszacowanie ryzyka realizacji przedsięwzięcia oraz określenie działań zmniejszających to ryzyko. Jeżeli w rozdziale 5 należy zastosować metodę ilościowego pomiaru efektów przedsięwzięcia, to analogicznie można ocenić i wielkość ryzyka jego realizacji.

Analiza ma przedstawiać czynniki sprzyjające / niesprzyjające skutecznej realizacji *przedsięwzięcia projektowego* a nie bezpieczeństwa zasobów informacyjnych planowanego *systemu informatycznego*. To ostatnie powinno być określone zarówno w opisie stanu aktualnego (rozdział 3.6) jak i w charakterystyce docelowego *systemu informatycznego* (rozdział 4.6)

## 2.11. Organizacja zarządzania przedsięwzięciem

Należy określić zasady zarządzania przedsięwzięciem, z określeniem sposobów postępowania w następujących obszarach:

- planowanie przedsięwzięcia,
- nadzorowanie przedsięwzięcia,
- zapewnienie jakości.

W zależności od sposobu realizacji przedsięwzięcia (patrz. rozdział .10), należy określić strukturę zespołów własnych i obcych odpowiedzialnych za wymienione obszary, określić zakresy odpowiedzialności i kompetencji tych zespołów oraz zdefiniować metody i kierunki monitorowania postępu prac.

## 2.12. Sposób realizacji przedsięwzięcia

Realizacja przedsięwzięcia powinna być przedstawiona wariantowo, ze wskazaniem wariantu proponowanego, wraz z uzasadnieniem. Należy przyjąć następujące założenia:

- 1) nie może być wariantowany zakres przedsięwzięcia (zdefiniowany w rozdziałach 4.1 - 4.5)
- 2) mogą być wariantowo przedstawione:
  - sposoby pozyskania *oprogramowania użytkowego* (własne opracowanie, zakup pakietu komercyjnego),
  - rozwiązania dotyczące sprzętu, wyposażenia sieciowego, telekomunikacji,
  - rozwiązania organizacyjne związane z *zarządzaniem przedsięwzięciem* (z udziałem podmiotów zewnętrznych lub bez takiego udziału ze wskazaniem uprawnień i odpowiedzialności dotyczących jakości rozwiązań, terminowości realizacji prac i kosztów przedsięwzięcia).

Należy ponadto wskazać:

- elementy poszczególnych wariantów, które zostały uwzględnione w szacunku nakładów (patrz: rozdział 6),
- wpływ (zwiększający lub zmniejszający) pozostałych elementów poszczególnych wariantów przedsięwzięcia na szacunek nakładów.

Edward Gawel  
Departament Rozwoju Informatyki  
i Systemu Rejestrów Państwowych  
Ministerstwa Spraw Wewnętrznych i Administracji



## Załącznik 3.1.1

### STRUKTURA ORGANIZACYJNA SYSTEMU

Lp.	Nazwa jednostki organizacyjnej	Kompetencje	Obowiązki	Sposób realizacji

#### Sposób wypełniania:

Nazwa jednostki organizacyjnej - nazwa jednostki organizacyjnej występująca w schemacie umieszczonym w rozdziale 3.1.1.

Kompetencje - zakres kompetencji jednostki organizacyjnej określony we właściwych dokumentach formalno-prawnych

Obowiązki - specyfikacja zadań szczegółowych jednostki organizacyjnej z właściwych regulaminów organizacyjnych

Sposób realizacji zadania - określenie w jaki sposób są realizowane kompetencje i obowiązki:

- „ręcznie”
- wspomagane komputerowo całkowicie (nazwy *systemów informatycznych* wspomagających)
- częściowo (nazwy czynności wspomaganych komputerowo i nazwy *systemów informatycznych* je wspomagających)

## Załącznik 3.1.2

### REALIZOWANE PROCEDURY

Lp.	Nazwa procedury	Wykonawcy	Stan wspomagania komputerowego	Cechy procedury

#### Sposób wypełniania:

Nazwa procedury - krótka nazwa procedury zawierająca informację merytoryczną

Wykonawcy - jednostki organizacyjne realizujące procedurę

Stan wspomagania komputerowego - określenie w jaki sposób jest realizowana procedura:

- „ręcznie”
- wspomagana komputerowo całkowicie (nazwy *systemów informatycznych* wspomagających)
- częściowo (nazwy elementów wspomaganych komputerowo i nazwy *systemów informatycznych* je wspomagających)

Cechy procedury - scharakteryzowanie procedury jedną z niżej wymienionych cech:

- stabilność
- zmienność
- podatność na zakłócenia

### Załącznik 3.1.3

#### STRUKTURA RAPORTOWANIA

Lp.	Nazwa raportu (klasy raportów)	Sporządza-jący	Źródło danych	Odbiorcy	Postać raportu	Sposób przekazy-wania

#### Sposób wypełniania:

- Nazwa raportu - nazwa raportu lub krótka nazwa klasy raportów (klasy raportów) zawierająca informację merytoryczną
- łączenie raportów w klasę raportów upraszcza zestawienie i powinno mieć miejsce w przypadku możliwości takiego samego opisanie grupy raportów w tabeli, o ile nie przewiduje się zmian w proponowanych rozwiązaniach
- Sporządzający - nazwa jednostki organizacyjnej, która odpowiada za poprawne i terminowe sporządzenie raportu
- Źródło danych - nazwy jednostek organizacyjnych, które dostarczają dane sporządzającemu lub tekst „własne”
- Odbiorcy - wyszczególnienie *użytkowników systemu* lub jednostek organizacyjnych / instytucji wchodzących w skład *otoczenia systemu*
- Postać raportu - forma sporządzania raportu: maszynopis, tabulogram komputerowy itd.
- Sposób przekazywania - środki służące do przekazania raportu: poczta, gońiec, telefaks, sieć komputerowa itd.



## Załącznik 3.2

### POWIĄZANIA Z OTOCZENIEM OBSZARU SYSTEMU

Lp.	Nazwa podmiotu	Kierunek przekazu	Nazwa danych	System informatyczny	Rodzaj	Postać	Sposób przekazywania

#### Sposób wypełniania:

- Nazwa podmiotu - nazwa jednostki organizacyjnej lub innego podmiotu z otoczenia obszaru systemu
- Kierunek przekazu - określenie relacji między podmiotem z otoczenia obszaru systemu a obszarem systemu: otrzymuje lub dostarcza
- Nazwa danych - nazwa raportu (klasy raportów) lub danych wejściowych (strumieni danych wejściowych) wymienianych z otoczeniem obszaru systemu
- System informatyczny - nazwa systemu informatycznego z otoczenia obszaru systemu lub z jego obszaru z którego pochodzą lub dla którego są przeznaczone dane
- Rodzaj - określenie, jakich danych dotyczy opis w wierszu tabeli: raportu (klasy raportów) lub danych wejściowych (strumieni danych wejściowych). Rodzaj powiązania jest niezależny od kierunku przekazu.
- Postać - forma raportu (klasy raportów) lub danych wejściowych (strumieni danych wejściowych), np.: maszynopis, tabulogram komputerowy, formularz, plik na nośniku komputerowym
- Sposób przekazywania - środki służące do przekazania raportu (klasy raportów) lub danych wejściowych (strumieni danych wejściowych), np.: poczta, goniec, telefaks, sieć komputerowa

CHARAKTERYSTYKA GŁÓWNYCH ZASOBÓW *DANYCH*

Lp.	Nazwa zasobu	Sposób utrzymywania

Sposób wypełniania:

Nazwa zasobu - nazwa bazy danych lub krótka nazwa kartoteki zawierająca informację merytoryczną

Sposób utrzymywania - określenie w jaki sposób jest utrzymywany opisywany zasób *danych*:

- „ręcznie”
- komputerowo (nazwa *systemu informatycznego* utrzymującego bazę danych):
  - off-line
  - on-line

## Załącznik 3.4

### CHARAKTERYSTYKA GŁÓWNYCH STRUMIENI DANYCH WEJŚCIOWYCH

Lp.	Nazwa danych WE (strumienia danych WE)	Źródło	Odbiorca	Postać	Sposób przekazywania	Sposób wykorzystania

#### Sposób wypełniania:

- Nazwa danych WE (strumienia danych WE) - nazwa danych wejściowych lub krótka nazwa strumienia danych wejściowych zawierająca informację merytoryczną
- łączenie danych wejściowych w strumień danych wejściowych upraszcza zestawienie i powinno mieć miejsce w przypadku możliwości takiego samego opisanie grupy danych wejściowych w tabeli, o ile nie przewiduje się zmian w proponowanych rozwiązaniach
- Źródło - nazwa jednostki organizacyjnej *obszaru systemu* lub jego *otoczenia*, która odpowiada za poprawne i terminowe dostarczenie danych źródłowych do systemu
- Odbiorca - *użytkownik systemu*, odpowiedzialny za wprowadzenie danych źródłowych do systemu
- Postać - forma danych ze źródła: formularz, plik na nośniku komputerowym itd.
- Sposób przekazywania - środki służące do przekazania danych ze źródła: poczta, gońiec, telefaks, sieć komputerowa itd.
- Sposób wykorzystania - nazwa *systemu informatycznego*, jeśli dane źródłowe są przetwarzane komputerowo lub słowo „ręcznie”



### CHARAKTERYSTYKA SYSTEMÓW INFORMATYCZNYCH W OBSZARZE SYSTEMU

Lp.	Nazwa systemu	Opis	Administrator

#### Sposób wypełniania:

- Nazwa systemu - nazwa *systemu informatycznego*
- Opis - krótka charakterystyka *systemu informatycznego* istotna dla *przedsięwzięcia*
- Administrator - jednostka organizacyjna w *obszarze systemu* zarządzająca eksploatacją *systemu informatycznego*

# WYKORZYSTANIE NARZĘDZI INFORMATYCZNYCH DO OCENY I CERTYFIKACJI ZARZĄDZANIA BEZPIECZEŃSTWEM I HIGIENĄ PRACY

Janusz K. GRABARA

## Wstęp

W Polsce można spotkać trzy systemy wspomagające zarządzanie bezpieczeństwem i higieną pracy. Są one wykorzystywane w niektórych większych przedsiębiorstwach. Wdrożenie systemu zarządzania bezpieczeństwem i higieną pracy jest jakby naturalną konsekwencją wdrażania systemów zarządzania jakością zgodnie z normą ISO serii 9000 i zarządzania ochroną środowiska zgodnie z normą ISO serii 14000. Zakłady które już posiadają certyfikaty ISO zabiegają o certyfikat z zakresu zarządzania bezpieczeństwem i higieną pracy. Problem powstaje gdy trzeba określić według jakiego standardu dokonać certyfikacji. Jednym z rozwiązań jest zastosowanie Polskiej Normy serii PN 18000 lecz norma ta ma wymiar lokalny (krajowy) i to w znacznym stopniu ogranicza jej spektakularny europejski wydźwięk.

Jedną z metod uzyskania certyfikatu z zakresu zarządzania bezpieczeństwem i higieną pracy jest skorzystanie z audytu stosowanego w którymś z krajów Unii Europejskiej.

Wśród wielu metod uzyskiwania certyfikatu na szczególną uwagę metoda wykorzystująca proste narzędzie informatyczne zbudowane na bazie arkusza kalkulacyjnego.

## Metoda International Safety Rating System

Metoda International Safety Rating System (ISRS) została opracowana przez pracowników amerykańskiego instytutu naukowego International Loss Control Institute, Longanville, Georgia oraz Det Norske Veritas w 1978 roku (ISRS-1). W latach 1984, 1988, 1990 i 1994 powstały kolejne wersje tej metody o coraz wyższych wymaganiach co do bezpieczeństwa pracy.

W ramach tej metody przedsiębiorstwo zostaje podzielone na 20 obszarów podlegających badaniu. W celu dogłębnej analizy wszystkich czynników wpływających na bezpieczeństwo pracy obszary przedsiębiorstwa dzieli się na 126 podobszarów. Dla ułatwienia i zobiektywizowania oceny zakładowego systemu bezpieczeństwa pracy analiza każdego obszaru odbywa się na podstawie stałej, określonej liczby pytań

Każde z pytań ma dokładnie zdefiniowane zasady punktowania (każde pytanie ma przypisaną określoną „wartość”). Cała metoda jest oparta na arkuszu kalkulacyjnym Excel, który jest wykorzystywany do analizy i wartościowania odpowiedzi.

Poniżej podano wykaz obszarów podlegających badaniu, odpowiadające im liczby pytań i „wartości”.

Tablica 1. Obszary podlegające działaniu, odpowiadające im liczby pytań i „wartości”

Lp.	Obszar	Liczba pytań	Wartości
1	Zarządzanie i administracja	74	1310
2	Szkolenie kadry kierowniczej	26	700
3	Przegląd bhp i ocena stanu technicznego	38	690
4	Analizy bezpieczeństwa pracy i instrukcje pracy	27	650
5	Badanie wypadków i zdarzeń potencjalnie wypadkowych	33	605
6	Obserwacja pracy	15	450
7	Przygotowanie działań na wypadek katastrofy	53	700
8	Organizacja pracy w zakładzie	36	615
9	Analiza wypadków i zdarzeń potencjalnie wypadkowych	33	533
10	Szkolenie personelu	41	700
11	Środki ochrony indywidualnej	18	380
12	Higiena pracy i ochrona zdrowia	52	700
13	System oceny zarządzania bezpieczeństwem pracy	27	700
14	Bezpieczna technika	29	670
15	Komunikacja interpersonalna	20	490
16	Współpraca (praca w zespole)	20	450
17	Promocja zagadnień bezpieczeństwa pracy w zakładzie	40	380
18	Rekrutacja i dobór do pracy	18	405
19	Zakupy	39	615
20	Bezpieczeństwo poza godzinami pracy	14	240
	<b>Razem</b>	<b>653</b>	<b>11983</b>

Tabela wartości podaje wartości maksymalne jakie można uzyskać w danym obszarze analizowanym przez system.



Wybór obszarów											
OES - Optional Element Score Lth	1										
PSS - Progressive System Score Lth	n										
Level	6										
Metoda pracy	PBA	OCS_01	OCS_02	OCS_03	OCS_04	OCS_05	OCS_06	OCS_07	OCS_08	OCS_09	OCS_10
Wybór obszarów	zakumul	zakumul	zakumul	1/3	1/3	1/3	1/3	1/3	1/3	1/3	zakumul
Wskazanie metody / poziomu	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
1	Zarządzanie i administracja	1	1	1	1	1	1	1	1	1	1
2	Szkolenie kadry kierowniczej	A									
3	Przeglądy bhp i ocena stanu technicznego	1	1	1	1	1	1	1	1	1	1
4	Analiza bezpieczeństwa pracy i instrukcje	A									
5	Badanie wypadków i zdarzeń prawie wypadkowych	A	1	1	1	1	1	1	1	1	1
6	Obserwacja procesu pracy	A									
7	Przygotowanie planów na wypadek katastrofy	A									
8	Organizacja pracy w zakładzie	A	1	1	1	1	1	1	1	1	1
9	Analiza wypadków i zdarzeń prawie wypadkowych	A									
10	Szkolenie pracowników	A									
11	Środki ochrony indywidualnej	A	1	1	1	1	1	1	1	1	1
12	Higiena pracy i ochrona zdrowia	A									
13	Systemy zarządzania bezpieczeństwem pracy	A									
14	Bezpieczna technika	A									
15	Komunikacja interpersonalna	A									
16	Praca zespołowa	A									
17	Promocja zagadnień bezpieczeństwa pracy	A									
18	Rekrutacja i przygotowanie pracowników do pracy	A									
19	Zakupy	A									
20	Bezpieczeństwo poza godzinami pracy	A									
PSS	Number of elements	5-0	5-0	5-1	6-1	6-2	7-3	7-5	10-5	13-5	20-0
PSS	Element Minimum %	13	14	16	17	18	19	20	30	30	30
PSS	Total Minimum %	25	25	25	25	30	30	40	40	55	75
OCS	Element Minimum %	40	40	40	40	40	40	50	55	70	90
OCS	Total Minimum %	10	15	15	20	20	25	30	40	50	75
OCS	Total Minimum %	25	30	35	35	40	40	30	40	75	90

Rys. 1 Ekran programu pokazujący wybór metody badania.

6	Poziom	OES	wynik	max	%	min		
1	Zarządzanie i administracja	TAK	30	1310	25	328		NIE
2	Szkolenie kadry kierowniczej	NIE	0	700	25	175	TAK	
3	Przeglądy bhp i ocena stanu technicznego	TAK	0	690	25	173		NIE
4	Analiza bezpieczeństwa pracy i instrukcje	NIE	0	650	25	163	TAK	
5	Badanie wypadków i zdarzeń prawie wypadkowych	TAK	0	605	25	151		NIE
6	Obserwacja procesu pracy	NIE	0	450	25	113	TAK	
7	Przygotowanie planów na wypadek katastrofy	NIE	0	700	25	175	TAK	
8	Organizacja pracy w zakładzie	TAK	0	615	25	154		NIE
9	Analiza wypadków i zdarzeń prawie wypadkowych	NIE	0	533	25	133	TAK	
10	Szkolenie pracowników	TAK	0	700	25	175		NIE
11	Środki ochrony indywidualnej	TAK	0	380	25	95		NIE
12	Higiena pracy i ochrona zdrowia	TAK	0	700	25	175		NIE
13	Systemy zarządzania bezpieczeństwem pracy	NIE	0	700	25	175	TAK	
14	Bezpieczna technika	TAK	0	670	25	168		NIE
15	Komunikacja interpersonalna	NIE	0	490	25	123	TAK	
16	Praca zespołowa	NIE	0	450	25	113	TAK	
17	Promocja zagadnień bezpieczeństwa pracy	TAK	0	380	25	95		NIE
18	Rekrutacja i przygotowanie pracowników do pracy	NIE	0	405	25	101	TAK	
19	Zakupy	TAK	0	615	25	154		NIE
20	Bezpieczeństwo poza godzinami pracy	NIE	0	240	25	60	TAK	
74	ISRS		30	11983	40	4793		NIE
75	PCS		0	65				NIE
76	Certyfikat							NIE

Rys.2 Ekran przedstawiający wynik certyfikacji jako wynik akceptacji poszczególnych obszarów.

13.1.1	6	Czy zakład posiada długofalowy plan wprowadzenia systemu ograniczania strat, w którym zdefiniowany został cel oraz sposób jego osiągnięcia	t/n (13.2.1)
13.1.2	6	Czy pracownicy oraz ich przedstawiciele biorą udział w opracowaniu długofalowego planu wprowadzenia systemu ograniczania strat	t/n
13.1.3	6	Czy długofalowy plan wprowadzenia systemu ograniczania strat został przedstawiony wszystkim pracownikom	t/n
13.1.4	8	Czy analiza kosztów i korzyści została przeprowadzona dla każdego elementu długofalowego planu wprowadzenia systemu ograniczania strat	t/n (13.1.5)
13.1.4.1	8	Co uwzględnia analiza kosztów i korzyści	
1	8	koszt inwestycji	t/n
2	8	koszty eksploatacyjne	t/n
3	8	okres zwrotu poniesionych nakładów	t/n
4	8	potencjalne zyski z wdrożenia	t/n
5	8	potencjalne straty w przypadku zaniechania wdrożenia	t/n
6	8	niematerialne zyski i straty	t/n
13.1.4.1	8	Co uwzględnia analiza kosztów i korzyści	automat
13.1.5	8	Czy kierownictwo zakładu zapewniło środki do realizacji długofalowego planu wprowadzenia systemu ograniczania strat	t/n

Rys.3 Fragment zestawu pytań dla obszaru 13 „ System oceny zarządzania bezpieczeństwem pracy.

Po przeprowadzeniu takiego audytu przedsiębiorstwu przypisuje się notę według 10-sto-pniowego systemu oceny.

Metoda ISRS jest stosowana do oceny poziomu bezpieczeństwa pracy w energetyce. W Polsce jest realizowana w ramach programu „Evisa”. Jednym z zakładów uczestniczących w programie „Evisa” były Zakłady Energetyczne Okręgu Radomsko-Kieleckiego w Skarżysku Kamiennej, które w 1999 roku uzyskały certyfikat potwierdzający osiągnięcie 5. poziomu w zakresie bezpieczeństwa pracy według metody ISRS.

Metoda ISRS pozwala na:

- ◆ dokładną analizę przyczyn wypadków, w oparciu o którą jest możliwe opracowanie planu poprawy zakładowego systemu bezpieczeństwa,
- ◆ analizę ryzyka zawodowego,
- ◆ opracowanie lub modyfikacje planu awaryjno-ratunkowego oraz systemu oceny warunków pracy,
- ◆ regulację procedur dotyczących bhp,
- ◆ wprowadzenie nowoczesnych metod analizujących przyczyny wypadków i zdarzeń potencjalnie wypadkowych.

Oprócz wymienionych powyżej zalet, metoda ta posiada również wady, ponieważ wymaga:

- ◆ powołania specjalnych zespołów wdrożeniowych i audytorskich oraz przeszkolenia powyższych zespołów,

- ♦ zapewnienia zespołom zajmującym się metodą odpowiedniego sprzętu i oprogramowania komputerowego.

W związku z powyższym metoda ta jest dość kosztowna i możliwa do zastosowania przez duże zakłady. Wadą tej metody jest również to, że nie ujmuje ona w analizie wypadków tzw. czynnika ludzkiego, np. stresu, zmęczenia.

## Podsumowanie

Przytoczony w materiale przykład stosowanego systemu certyfikacji w zakresie zarządzania bezpieczeństwem i higieną pracy ma na celu pokazanie oprócz praktycznego zastosowania w konkretnym obszarze dla którego został stworzony, również pewnej drogi jaką można iść w kierunku opracowania systemu audytorskiego dla oceny i weryfikacji systemów informatycznych zarówno zintegrowanych jak i specjalizowanych.

Stworzenie listy pytań (w ISRS jest ich 1270) podzielonych na obszary mogłaby w sposób jednoznaczny standaryzować metodę oceny systemu zarówno z punktu widzenia wdrażającego jak i użytkownika.

Janusz K. Grabara  
Instytut Informatyki i Ekonometrii  
Wydział Zarządzania; Politechnika Częstochowska  
Ul. Armii Krajowej 19b  
42-200 Częstochowa





# JAK WDROŻYĆ METODYKĘ ZARZĄDZANIA PROJEKTAMI „PRINCE 2”<sup>®</sup>

Wiesław KOSIERADZKI

**Streszczenie:** Metodyka zarządzania projektami PRINCE 2 szybko zdobywa pozycję wiodącej metodyki w zarządzaniu projektami. W opracowaniu przedstawiono praktyczne aspekty wdrażania jej w firmach, w których obok typowych zadań rutynowych prowadzi się zadania o charakterze projektów. Wdrożenie potraktowano jako projekt, który od początku powinien być wdrażany zgodnie z tą metodyką. Dzięki takiemu podejściu można osiągnąć równocześnie dwa cele: pierwszy to wdrożenie metodyki, a drugi – zdobycie przez realizatorów projektu doświadczenia praktycznego w jej stosowaniu. W artykule wykorzystano doświadczenia zgromadzone w trakcie wdrażania PRINCE 2 w kilku polskich przedsiębiorstwach.

## 1. Wprowadzenie

Konieczność konkurowania na coraz trudniejszych rynkach zmusza przedsiębiorstwa do:

- ciągłego wprowadzania nowych produktów i skracania czasu potrzebnego na ich opracowanie i rozwój
- wzrostu produktywności zaangażowanych zasobów i ciągłej poprawy jakości oferowanych produktów
- skracania cykli produkcyjnych i cykli realizacji zamówień
- większej specjalizacji
- korzystania z zasobów zewnętrznych (outsourcing).

Wymaga to równoczesnego realizowania wielu przedsięwzięć, które:

- charakteryzują się niepowtarzalnością
- są prowadzone w celu uzyskania unikatowego wyniku, którym może być wprowadzenie zmiany lub wytworzenie produktu materialnego lub niematerialnego
- powinny się skończyć w określonym terminie
- pochłaniają określone zasoby.

---

<sup>®</sup> PRINCE jest zastrzeżonym znakiem handlowym Central Computer and Telecommunication Agency w Wielkiej Brytanii.

Takie przedsięwzięcia, w odróżnieniu od rutynowo powtarzanych działań, nazywamy projektami.

Do projektów zaliczmy wdrożenie systemu informatycznego, opracowanie programu komputerowego, uruchomienie produkcji komputerów nowego typu, zorganizowanie i przeprowadzenie kampanii marketingowej. Projektem jest także wdrożenie w organizacji wybranej metodyki zarządzania projektami.

Często kilka projektów ma przynieść organizacji określone korzyści w dłuższym okresie. Portfel projektów wybranych, planowanych i zarządzanych w sposób skoordynowany, które razem przynoszą organizacji oczekiwane korzyści, nazywamy programem.

Jako przykłady programów można wymienić:

- Program restrukturyzacji przedsiębiorstwa
- Program informatyzacji firmy
- Program redukcji kosztów operacyjnych.

Na każdy z tych programów może się składać wiele różnych projektów, których realizacja przyczynia się do osiągnięcia celów lub korzyści wymienionych zwykle w jego nazwie.

W przypadku realizacji jednego projektu, mógłby on być zarządzany w dowolny sposób i dla realizującej go organizacji nie miałyby znaczenia, jakie zastosowano procedury czy wzory dokumentów. Nawet w małych organizacjach jest jednak realizowanych wiele różnorodnych przedsięwzięć lub projektów i niezależnie od tego, czy są to projekty realizowane w ramach jednego programu, czy są to projekty indywidualne, w trakcie ich realizacji występuje wiele wspólnych elementów, na przykład:

- formaty dokumentów
- struktura danych dotyczących produktów (koszty, pracochłonność, rodzaje zasobów wykorzystywanych do ich wytworzenia)
- procesy i procedury dotyczące:
  - poszczególnych faz realizacji projektu
  - monitorowania
  - kontrolowania postępu
  - sprawozdawczości
- procedury organizowania pracy zespołów roboczych
- stosowane metody i techniki (planowanie, techniki rozwiązywania problemów).

W interesie organizacji realizującej wiele projektów, a z pewnością w interesie zespołu zarządzającego programem leży, aby we wszystkich projektach wymienione



powyżej elementy były ujednolicone i doskonalone w oparciu o doświadczenia płynące z kolejnych projektów i programów. Z tych względów wiele organizacji wprowadza jednolitą metodykę zarządzania programami i projektami. Jedną z takich metodyk jest opisana poniżej metodyka PRINCE 2, która coraz szybciej zdobywa sobie status międzynarodowego standardu zarządzania projektami.

Celem niniejszego opracowania nie jest szczegółowe przedstawienie tej metodyki, lecz pokazanie, w jaki sposób można ją skutecznie wdrożyć do praktyki zarządzania projektami, niezależnie od obszaru, których one dotyczą.

Jak wspomniano wcześniej, wdrożenie w organizacji metodyki zarządzania projektami, a więc także i metodyki PRINCE 2, może być zrealizowane jako projekt. Z tego względu w tym opracowaniu takie wdrożenie zostało przedstawione z wykorzystaniem wielu elementów tej metodyki. W ten sposób Czytelnik z jednej strony pozna nie tylko proces wdrażania metodyki PRINCE 2, lecz także kilka jej istotnych elementów.

## **2.1. Metodyka zarządzania projektami PRINCE 2 – istota i geneza**

W wyniku analiz wielkiej liczby różnych projektów zakończonych sukcesami i niepowodzeniami stwierdzono, że w większości przypadków przyczyną niepowodzeń projektów było złe zarządzanie ich przygotowaniem i realizacją, a także to, że organizacje nie wyciągały wniosków z błędów popełnionych we wcześniej zrealizowanych projektach. Tworząc PRINCE 2 wybrano te elementy, które stosowano w projektach zakończonych sukcesem i wprowadzono mechanizmy eliminujące to, co było przyczyną niepowodzeń.

Metodyka zarządzania projektami PRINCE 2 powstała w Wielkiej Brytanii. Jest ona publicznie dostępna i może być używana przez wszystkich, chociaż prawa autorskie pozostają w posiadaniu Korony Brytyjskiej a PRINCE jest marką zastrzeżoną przez Central Computer and Telecommunications Agency (CCTA).

Została oparta na wcześniejszej metodyce, znanej pod nazwą PROMPT, opracowanej w roku 1975 przez Simfact System Ltd. Od roku 1979 metodyka PROMPT była wdrażana przez CCTA jako standard używany we wszystkich projektach informatycznych wykonywanych dla Rządu Wielkiej Brytanii. W 1989 roku metodyka PROMPT została zastąpiona metodyką PRINCE, której nazwa to akronim od „**P**rojects **I**N **C**ontrolled **E**nvironments”, czyli „projekty w kontrolowanych warunkach”.

W roku 1996 powstała nowsza, unowocześniona wersja tej metodyki pod nazwą PRINCE 2, dostosowana do potrzeb różnych projektów, nie tylko

informatycznych. Na podstawie tego nowego standardu brytyjska firma SPOCE Ltd. opracowała wzorcowy sposób jej stosowania wraz z wzorami potrzebnych do tego dokumentów. Przyspieszyło to popularyzację metodyki PRINCE 2, która jest obecnie stosowana przez wielkie światowe korporacje, jak i niewielkie firmy usługowe, a umiejętność jej stosowania przez menedżerów projektów jest weryfikowana systemem egzaminów i certyfikatów. Centrum Rozwiązań Menedżerskich S.A. we współpracy z tą firmą opracowało polską wersję tego wzorca, o nazwie „PRINCE 2/SPOCE/CRM”. PRINCE 2 od kilku lat jest już z powodzeniem stosowana w Polsce. Zgodnie z tą metodyką był zarządzany bardzo duży projekt zrealizowany przez TP S.A. „Rok 2000”.

W roku 2002 wprowadzono do metodyki PRINCE 2 kolejne udoskonalenia.

Metodyka PRINCE 2 charakteryzuje się tym, że:

- wyróżnia osiem procesów, które występują w projekcie i decydują o jego powodzeniu oraz ściśle określa role odpowiedzialne za ich realizację, czyli ich właścicieli;
- w każdy z ośmiu procesów wbudowane są mechanizmy zapewnienia jakości;
- skupia się na ciągłej ocenie zasadności kontynuowania projektu (projekty, które straciły uzasadnienie powinny być przerywane niezależnie od wielkości poniesionych już nakładów);
- przed rozpoczęciem realizacji projektu oraz w trakcie jego realizacji wymagana jest analiza ryzyka zagrażającego projektowi i zarządzanie nim;
- kładzie nacisk na planowanie oparte na produktach, które mają powstać w wyniku realizacji projektu – działania są tylko sposobem ich wytworzenia lub uzyskania;
- rozdziela bieżące zarządzanie projektem, za które odpowiada kierownik projektu, od decyzji strategicznych, za które odpowiada specjalnie powoływany do tego celu komitet sterujący;
- w komitecie sterującym projektem są reprezentowane trzy strony projektu: biznes lub inwestor, czyli strona, która zamierza dzięki projektowi osiągnąć określone korzyści, użytkownik produktów powstających w ramach projektu, a także dostawcy usług lub rozwiązań, niezależnie od tego, czy pochodzą z wewnątrz, czy z zewnątrz organizacji; dzięki temu wszelkie decyzje dotyczące projektu mogą być podejmowane w trybie poszukiwania konsensu, przy decydującym głosie inwestora we wszelkich sprawach dotyczących uzasadnienia biznesowego projektu;
- precyzyjnie określa role w zarządzaniu projektem, zakresy obowiązków i odpowiedzialności przypisane tym rolom oraz ich relacje z innymi rolami w projekcie, nie dopuszczając do ich mieszania (np. nie zezwalając na to, aby menedżer projektu lub zespoły wykonawcze sami oceniali wytwarzane „produkty” projektu);

- wyróżnia w projektach etapy merytoryczne lub techniczne, związane z wytwarzaniem określonych kolejnych produktów projektu, oraz etapy zarządcze, które występują jeden po drugim i pozwalają lepiej sterować projektem oraz ograniczyć ryzyko, że się on nie powiedzie;
- określa procedury postępowania w sytuacjach awaryjnych i wynikających z nich koniecznych zmian, dzięki czemu wszelkie zmiany są wprowadzane w sposób w pełni kontrolowany;
- określa dokumenty, które powinny być stosowane w zarządzaniu projektem, a także minimalne wymagania odnośnie informacji, które powinny być w nich zawarte;
- zachowuje poziom elastyczności umożliwiający jej stosowanie do projektów różnego rodzaju i o różnej skali;
- pozwala dla dobra realizacji projektu koordynować działania podejmowane przez podwykonawców uczestniczących w projekcie, kiedy przy realizacji powierzonych im do wykonania zadań posługują się swoją własną metodyką, np. przy wdrażaniu systemów informatycznych.

Zarządzanie projektem zgodnie z PRINCE 2 ułatwia jego realizację, znacznie odciąża naczelną kadrę kierowniczą organizacji od bieżącego zarządzania projektem, zapewniając jednak skuteczny nadzór nad jego realizacją, zwiększa szanse doprowadzenia go do zaplanowanego końca, terminowo, w ramach budżetu, w pełnym zakresie i przy spełnieniu określonych wymagań jakościowych. Także w przypadku wystąpienia nieoczekiwanych trudności, podważających celowość dalszej realizacji projektu, stosując metodykę PRINCE 2 można ograniczyć straty, jakie wiążą się z koniecznością przerwania projektu.

### **3.2. Opis projektu: „Wdrożenie metodyki PRINCE 2”**

Punktem wyjścia do projektu jest określenie jego końcowego wyniku lub produktu.

Zwykle jest to zapisywane w dokumencie „Zlecenie rozpoczęcia prac”.

W naszym przypadku oczekiwanym wynikiem ma być wdrożenie metodyki PRINCE 2. Tak określony wynik byłby jednak trudny do zweryfikowania i w dalszych pracach wynik ten będzie określony w kategoriach umożliwiających późniejsze zweryfikowanie, czy projekt faktycznie zakończył się osiągnięciem zamierzonego celu.

Na zakres prac wykonywanych w projektach, a następnie na przebieg prac, wpływa wiele czynników. Część z nich jest zależna od potrzeb organizacji - np. zakres prac i termin, od ilości i dostępności zasobów potrzebnych do jego realizacji, a także od sposobu zarządzania projektem, czyli od umiejętności członków zespołu zarządzającego. Istotne jest także uzasadnienie ekonomiczne projektu i ryzyko



związane z jego realizacją. Ważne są też założenia i ograniczenia, przy jakich projekt ma być zrealizowany, w jakiej formule i jak będzie zarządzany.

Metodyka PRINCE 2 wymaga, by zagadnienia te zostały przeanalizowane i zapisane już w fazie przygotowywania projektu. Służą temu takie dokumenty jak:

- Podstawowe założenia projektu
- Formuła realizacyjna
- Organizacja zarządzania projektem

Dokumenty te wraz z *Planem konstruowania projektu*, czyli planem planowania i organizowania projektu, powinny być przedłożone do zatwierdzenia Komitetowi sterującemu.

Ze względu na ograniczoną ilość miejsca omawianie tych dokumentów ograniczymy do ich najistotniejszych elementów. Czytelników zainteresowanych dokumentami stosowanymi w PRINCE 2 odsyłamy do opisu metodyki lub do wzorów dokumentów zawartych w PRINCE 2/SPOCE/CRM.

### **3.1.a. Dokument Podstawowe założenia projektu**

W projektach realizowanych zgodnie z PRINCE 2 dokument *Podstawowe założenia projektu* stanowi fundament projektu i powinien być starannie przygotowany. Z tego względu poświęcimy mu więcej miejsca.

Dokument ten ma na celu pierwsze sformułowanie zakresu projektu, ewentualnych potrzeb inwestycyjnych, zależności od innych projektów oraz przewidywanych korzyści, jakie powinna przynieść realizacja projektu. Dopiero na tej podstawie można określić priorytety dla produktów projektu, uzgodnić sposób ich sfinansowania i zdecydować o przyjęciu projektu do realizacji. Podstawowe założenia projektu stanowią podstawę dla komitetu sterującego do podjęcia decyzji co do celowości kontynuacji prac nad projektem. Dlatego warto dobrze je przemyśleć i przygotować. Należy pamiętać, że projekty zbudowane na złych założeniach zwykle kończą się niepowodzeniem.

Poniżej przedstawiono zagadnienia, które zgodnie z PRINCE 2 powinny być omówione w dokumencie *Podstawowe założenia projektu*.

#### **1. Przesłanki projektu**

Przystępując do wdrożenia PRINCE 2 musimy określić powody, dla których projekt ma być zrealizowany, oraz określić, jakie wystąpiły przesłanki stanowiące podstawę do podjęcia prac. Powinniśmy także wiedzieć, jakie korzyści powinno przynieść to wdrożenie. Odpowiadając na te pytania warto się także zastanowić, komu z naczelnego kierownictwa firmy wdrożenie tej metodyki powinno przynieść największe korzyści, jakie i kiedy. Chodzi tu o osobę, która osobiście odpowiada za

powodzenie i niepowodzenia najważniejszych projektów realizowanych przez firmę. Może to być na przykład członek zarządu odpowiedzialny za badania i rozwój, lub dyrektor produkcji. Taką osobą nazywamy sponsorem projektu. Wskazane jest, by sponsor pełnił w projekcie rolę przewodniczącego komitetu sterującego. Jeśli takiej osoby nie jesteśmy w stanie wskazać, warto jeszcze raz przemyśleć celowość podjęcia projektu.

## **2. Cele projektu**

Wypełniając ten punkt należy określić, w jakim celu będziemy wdrażać metodykę PRINCE 2. Celem tym może być, na przykład: poprawa produktywności zespołów projektowych poprzez ujednoczenie dokumentacji i procesów zarządzania projektem. Może to być także konieczność wynikająca z coraz częściej stawianego wymagania klientów, że projekt realizowany dla nich musi być zarządzany według metodyki PRINCE 2. W tym ostatnim przypadku warto pamiętać o złych doświadczeniach firm, które za swój cel przyjęły uzyskanie certyfikatu ISO 9001. Wiele z nich certyfikat uzyskało, ale nie odniosło korzyści, jakie daje wdrożenie systemu zarządzania jakością zgodnego z tą normą. Wynika z tego, że celem projektu nie powinno być abstrakcyjne „wdrożenie metodyki”, lecz osiągnięcie mierzalnych wyników tj. ograniczenie liczby różnorodnych formatów dokumentów, wprowadzenie standardów dla dokumentacji projektowej – zarządczej, technicznej i dotyczącej jakości produktów, itp. Jednym z celów wdrożenia PRINCE 2 powinno być zorganizowanie wsparcia dla projektów w postaci biura projektów lub biura programów i projektów.

## **3. Zakres i punkty styku z innymi podmiotami**

W tym punkcie powinniśmy wskazać główne obszary, funkcje, procesy itp., jakie mają być objęte projektem. Trzeba wskazać, co będzie objęte projektem, a co pozostanie poza nim.

W rozważanym przypadku projekt powinien objąć wszystkie funkcje i procesy związane z zarządzaniem projektami realizowanymi przez firmę na swoje potrzeby, lub w których firma uczestniczy jako dostawca lub poddostawca.

W tym miejscu powinniśmy wyraźnie zaznaczyć, że projekt nie obejmuje procesów technologicznych związanych z wytwarzaniem produktów powstających w ramach projektów realizowanych przez naszą firmę. W firmach

informatycznych to oznacza, że projekt nie będzie obejmował procesów takich jak: analiza systemowa, programowanie, kodowanie, czy testowanie systemów. Tutaj także należy określić, czy w projekcie ograniczymy się do zarządzania projektami, czy też, a jeśli tak, to w jakim stopniu, zostanie uwzględnione metodyczne zarządzanie programami.

#### **4. Najważniejsze produkty**

W tym punkcie powinniśmy określić konkretne produkty, które będą wynikiem projektu, czyli pojęcie „wdrożenie PRINCE 2” przedstawić jako zestaw produktów, które składają się na nie. Przykładowo produktami takimi mogą być:

- Opisane procesy i procedury zarządzania projektami zgodne z PRINCE 2
- Wzorce dokumentów z określonymi dla nich wymaganiami informacyjnymi
- Funkcjonujące „wsparcie dla projektów” w postaci zorganizowanego biura programów i projektów
- Ludzie przygotowani do praktycznego stosowania metodyki PRINCE 2 (kierownictwo organizacji, kierownicy projektów i członkowie zespołów projektowych, użytkownicy uczestniczący w realizacji projektów)
- Określony i funkcjonujący system komunikacji w projektach
- Funkcjonujący system zarządzania wiedzą o projektach
- Zastosowanie PRINCE 2 we wszystkich projektach rozpoczynanych po określonym dniu.

#### **5. Ograniczenia**

Żadna organizacja nie jest w stanie przeznaczyć na projekt nieograniczonych środków i zasobów. Z tego względu w tym punkcie powinniśmy wpisać wszystkie ograniczenia (limity) dotyczące czasu, zasobów, funduszy oraz wskazać sposób ich ewentualnej akceptacji. Kierownik projektu, którego jednym z zadań jest przygotowanie tego dokumentu przy współpracy ze sponsorem, powinien pamiętać, że w tym punkcie należy określić ramy, w których będzie musiał zrealizować projekt, a także te obszary, na które nie będzie miał wpływu. W trakcie realizacji projektu każde wyjście poza tutaj określone granice będzie wymagało akceptacji ze strony komitetu sterującego.



## 6. Założenia

Na etapie przygotowywania projektu kierownik projektu nie zna wielu szczegółów projektu. Dlatego powinien określić założenia, jakie przyjął. W przypadku naszego projektu założeniem takim może być przyjęcie, na przykład, że:

- w realizacji projektu będą uczestniczyć najbardziej doświadczeni kierownicy projektów
- kierownictwo organizacji wyznaczy przyszłego kierownika biura programów i projektów, który będzie kierował zespołem tworzącym takie biuro
- ośrodek informatyki udostępni na potrzeby projektu określone zasoby.

## 7. Ogólne korzyści biznesowe i uzasadnienie biznesowe

W organizacjach gospodarczych każdy projekt powinien przynieść określone korzyści biznesowe. Mogą to być: dodatkowe zyski, oszczędności czasu lub/i pieniędzy, zwiększenie udziałów rynkowych poprzez pozyskanie nowych klientów. Przystępując do wdrożenia PRINCE 2 także trzeba określić korzyści, jakie ma projekt przynieść. Ich określenie i zrozumienie przez wszystkich uczestników projektu i kierownictwo firmy może mieć decydujący wpływ na jego powodzenie.

## 8. Wstępna ocena ryzyka

Oczekując korzyści należy także brać pod uwagę zagrożenia, które mogą spowodować, że projekt nie zakończy się zgodnie z oczekiwaniami. Metodyka PRINCE 2 wymaga, aby zarządzać ryzykiem związanym z realizowanymi projektami. Z tego względu w tym miejscu należy wypisać czynniki ryzyka i ocenić ich ewentualny wpływ na przebieg projektu i osiągnięcie zamierzonych celów. Można też sformułować pytania (listę tematów), jakie będą musiały być rozważone w kontekście zidentyfikowanych zagrożeń.

Najważniejsze ryzyka związane z wdrażaniem PRINCE 2 to:

- brak zrozumienia celowości wdrożenia PRINCE 2 ze strony kierownictwa firmy i determinacji w jej wdrażaniu
- niedostateczna wiedza o metodyce

- brak lidera zmian
- opór ze strony doświadczonych kierowników projektów, którzy mając własne metodyki zarządzania projektami niechętnie odnoszą się do tego, czego sami nie wymyślili
- zbyt szybkie tempo wdrożenia w stosunku do stopnia przygotowania organizacji
- opieranie się na złych wzorcach.

Warto także wskazać sposoby, jakimi będziemy zarządzać tymi ryzykami.

### 9. *Oczekiwania klienta co do jakości*

Każdy projekt związany jest z wytworzeniem produktów, którymi mogą być wytwory materialne, jak nowy system komputerowy, wykonane usługi, wartości niematerialne lub wprowadzona zmiana. Osoby, które będą je wykorzystywać, lub których będą w jakimś stopniu dotyczyć, w projektach są uważani za klientów. Przygotowując projekt do realizacji musimy poznać ich oczekiwania co do jakości produktów projektu.

W rozważanym projekcie podstawowym standardem, którego wymagania powinniśmy spełnić, jest sama metodyka PRINCE 2. Warto też pamiętać, że jej wdrożenie w organizacji, w której jest system zarządzania jakością ISO 9001: 2000, powinno uwzględniać wymagania tego systemu. W firmach współpracujących z NATO lub z instytucjami o specjalnym charakterze do takich wymagań należy zaliczyć, między innymi, zgodność systemu i wprowadzanych rozwiązań z obowiązującymi w nich standardami bezpieczeństwa.

#### **3.2.b. Dokument „Organizacja zarządzania Projektem”**

Powodzenie projektu w dużym stopniu zależy od jasnego określenia podziału zadań i odpowiedzialności oraz przypisaniu ich określonym rolom występującym w zarządzaniu projektem. Świadomie użyto tu określenia „rolom” a nie „osobom”, gdyż metodyka PRINCE 2 dopuszcza, by ta sama osoba pełniła w projekcie różne role, przy czym metodyka wyklucza łączenie niektórych ról.

Za powodzenie projektu i zarządzanie strategiczne jego realizacją będzie odpowiadał Komitet sterujący, w którego skład powinni wejść:

***Przewodniczący Komitetu sterującego (Sponsor)*** – członek naczelnego kierownictwa, który odpowiada za powodzenie głównych projektów

realizowanych przez firmę dla jej klientów lub na wewnętrzne potrzeby firmy. Powinien mieć on uprawnienia do dysponowania zasobami przeznaczonymi do i na realizację projektu.

**Główny Użytkownik** – członek kierownictwa odpowiedzialny za zarządzanie projektami realizowanymi przez firmę lub osoba o dużym doświadczeniu w kierowaniu projektami, zainteresowana wdrożeniem PRINCE 2. W Komitecie Sterującym będzie on reprezentował interesy wszystkich osób, które w przyszłości będą wykorzystywać w swojej pracy wdrażaną metodykę.

**Główny Dostawca** – kierownik pionu organizacyjnego odpowiedzialnego za wdrożenie PRINCE 2. W przypadku korzystania z pomocy firmy doradczej przedstawiciel jej kierownictwa powinien występować w Komitecie Sterującym jako jeden z Głównych Dostawców. Rola ta reprezentuje interesy osób realizujących projekt, czyli jego dostawców.

Ponieważ członkowie komitetu sterującego nie mogą zwykle poświęcić zbyt wiele czasu na sprawy projektu, funkcje związane z odbiorem gotowych produktów oraz nadzorem nad przestrzeganiem wymagań jakościowych komitet sterujący może delegować **Audytorowi**. Powinna to być osoba fizyczna lub prawna, działająca na rzecz Komitetu sterującego, biegła w metodyce PRINCE 2 i zarządzaniu projektami, której zadaniem będzie zapewnienie, by projekt był realizowany zgodnie z przyjętymi planami i w ramach przydzielonych zasobów. Powinien on także zapewnić, by w trakcie realizacji projektu był on zarządzany zgodnie z przyjętą metodyką oraz przy spełnieniu przyjętych standardów jakościowych.

Za bieżące zarządzanie projektem i za terminowe wykonanie produktów projektu, zgodnie z określonymi dla nich wymaganiami oraz w ramach przyznanym na ten cel środków, będzie odpowiadał **Kierownik projektu** - osoba doświadczona w zakresie zarządzania projektami, znająca i umiejąca stosować PRINCE 2. (Realizowany projekt może być wykorzystany jako okazja do przeszkolenia uczestniczących w nim pracowników i zdobycia przez nich podstawowych umiejętności i doświadczenia praktycznego).

Metodyka PRINCE 2 przewiduje **wsparcie dla projektu** w postaci sekretariatu lub biura. Ponieważ takie wsparcie może odegrać istotną rolę we wdrażaniu PRINCE 2, omówiono je dalej szerzej.

Realizacja projektu to nie tylko zarządzanie, ale także wykonywanie konkretnych zadań. W tym celu powinny być powołane **Zespoły zadaniowe lub realizacyjne**.

W przypadku wdrażania PRINCE 2 celowe jest utworzenie następujących zespołów:



- a) Zespół procesów, procedur i instrukcji – zamodelowanie procesów zarządczych, opracowanie procedur i instrukcji
- b) Zespół dokumentacji – przygotowanie wzorów dokumentów stosowanych w projektach
- c) Zespół wsparcia informatycznego – określenie potrzeb, wybór odpowiedniego oprogramowania oraz ewentualnie jego dostosowanie do specyficznych potrzeb firmy
- d) Zespół ds. jakości w projektach i zarządzania konfiguracją – opracowanie wzorcowego planu jakości oraz metody zarządzania konfiguracją dla najbardziej typowych projektów realizowanych przez firmę, a w przypadku, kiedy firma posiada system zarządzania jakością – zapewnienie spójności zarządzania projektami i ich realizacją z wymaganiami systemu zarządzania jakością
- e) Zespół ds. szkoleń – zorganizowanie i ewentualnie przeprowadzenie szkoleń dla wszystkich uczestników projektów w zakresie metodyki PRINCE 2 oraz produktów wyżej wymienionych zespołów
- f) Zespół ds. zorganizowania Biura programów i projektów – zorganizowanie wsparcia dla projektów realizowanych przez firmę; docelowo biuro to powinno stać się centrum wiedzy o projektach.

Wielkość zespołu zależy od rodzaju i ilości zadań, które ma on wykonać. W niektórych przypadkach może nie być potrzeby tworzenia zespołów, a poszczególne zadania lub grupy zadań mogą wykonać pojedyncze osoby. To te osoby i *członkowie zespołów realizacyjnych* będą faktycznymi realizatorami projektu. W naszym projekcie mogą to być wyłącznie pracownicy firmy wdrażającej PRINCE 2, ale w przypadku braku wiedzy i doświadczenia w tym zakresie, w skład zespołów powinien wejść konsultant lub konsultanci z firmy doradczej.

---

Metodyka PRINCE 2 wymaga, by przystępując do prac nad projektem w pierwszej kolejności powołano przewodniczącego Komitetu sterującego oraz kierownika projektu, gdyż to oni powinni odpowiednio przygotować go do realizacji, a następnie doprowadzić do pozytywnego zakończenia. Kierownik projektu powinien także określić produkty, które powinny powstać w wyniku prac poszczególnych zespołów zadaniowych i następnie wybrać osoby, które takie zespoły zorganizują i pokierują ich pracą.

### **3.3.c. Dokument „Formuła realizacyjna projektu”**

Dokument *Formuła realizacyjna* określa, w jaki sposób projekt będzie realizowany.

Przygotowując do realizacji projekt „Wdrożenie PRINCE 2” kierownik projektu musi ocenić potencjał firmy w zakresie metodycznego zarządzania projektami. Jeśli firma dysponuje osobami znającymi metodykę PRINCE 2 i posiadającymi doświadczenie w jej stosowaniu, wtedy może zrealizować taki projekt własnymi siłami z ewentualnym wsparciem przez wybraną firmę doradczą. Jeśli jednak nie dysponuje takimi osobami, powinna zrealizować projekt przy istotnym udziale konsultantów zewnętrznych. Nie może być to jednak kupienie rozwiązania wykonanego „pod klucz”. Metodyka PRINCE 2 nie narzuca bowiem rozwiązań szczegółowych, a wręcz wymaga, by wykorzystać doświadczenie firmy, w której się ją wdraża.

To oznacza, że także wtedy, kiedy firma zamierza skorzystać z doradztwa zewnętrznego, należy przewidzieć znaczący udział własnych pracowników. Do tego zagadnienia wrócimy w dalszej części opracowania.

W firmach posiadających system zarządzania jakością ISO 9001: 2000 należy zapewnić spójność wdrażanych rozwiązań z wymaganiami metodyki.

#### **4.3. Opracowanie planu projektu**

W metodyce PRINCE 2 planowanie oparte jest na produktach, czyli punktem wyjścia do planu są produkty, które mają być wytworzone w ramach projektu, półprodukty wchodzące w ich skład oraz kolejne elementy, które są konieczne do wytworzenia bardziej złożonych produktów. Na tej podstawie określane są działania konieczne do przetworzenia jednych w drugie. W przeciwieństwie do wielu podejść do planowania, w których punktem wyjścia jest struktura działań (WBS - Work Breakdown Structure), w projektach zarządzanych zgodnie z PRINCE 2, WBS powstaje na podstawie struktury produktu i strumieni przepływu elementów składowych od najdrobniejszych, poprzez bardziej złożone, aż do produktu finalnego. Warto pamiętać, że celem projektu są jego produkty, a działania są jedynie koniecznością, by mogły one powstać lub by można było je uzyskać w inny sposób. Program komputerowy potrzebny do obsługi biura projektów można wykonać własnymi siłami, albo można go kupić. Widać tu dobrze, że do oczekiwanego produktu można dojść wykonując różne działania i wykorzystując różne zasoby.

W celu zaplanowania projektu „Wdrożenie metodyki PRINCE 2” musimy dokładnie określić produkty, które na takie wdrożenie się złożą.

Poniżej przedstawiono produkt „Metodyka PRINCE 2 wdrożona w organizacji” w formie tzw. listy wciętej, pokazującej część jego struktury:

Metodyka PRINCE 2 wdrożona w organizacji:

- Opisane procesy i procedury

- Procedury zarządcze zgodne z PRINCE 2
  - schemat procedury
  - opis słowny
  - instrukcja 1
    - schemat instrukcji
    - opis słowny
- Procedura zarządzania jakością
- Procedura zarządzania konfiguracją
- Procedura zarządzania zmianami
- Procedura zgłaszania zagadnień realizacyjnych
- ...
- Wzorce dokumentów i wymagania informacyjne
  - Formularz „Zlecenie realizacji prac”
  - Formularz „Podstawowe założenia projektu”
  - Formularz „Mianowanie przewodniczącego Komitetu sterującego i Kierownika projektu”
  - ....
- Funkcjonujące „wsparcie dla projektów” – (BPiP)
  - Przygotowani pracownicy
    - Kierownik biura przygotowany do obsługi projektów zgodnie z PRINCE 2
    - Planista znający MS Project
    - Kosztorysant
    - ...
  - Pomieszczenie na biuro projektów
    - półki
    - wyposażenie komputerowe
    - szafy archiwalne
    - ...
- Pracownicy firmy przygotowani do stosowania PRINCE 2
  - kierownictwo organizacji
  - kierownicy projektów
  - kierownicy zespołów projektowych
  - członkowie zespołów projektowych
  - użytkownicy uczestniczący w realizacji projektów
- System komunikacji
  - system połączeń telefonicznych i faksowych
  - sieć LAN dostępna dla wszystkich uczestników projektu
  - oprogramowanie MS Outlook na wszystkich stanowiskach komputerowych



- poczta elektroniczna
- System zarządzania wiedzą o projektach
  - podręcznik elektroniczny PRINCE 2/SPOCE/CRM dostępny na wszystkich stanowiskach komputerowych
  - baza z wzorami wszystkich standardowych dokumentów
  - baza informacji o zakończonych projektach
    - opisy projektów A, B i C
      - lista produktów
      - plan realizacji
      - raporty z nauczek
      - ...

W celu wytworzenia lub pozyskania powyższych produktów konieczne jest wykonanie, między innymi, następujących działań (prac):

- Opisanie procesów i procedur PRINCE 2
  - procesy zarządcze
  - procesy główne
  - procesy pomocnicze
- Inwentaryzacja dokumentów wymaganych przez PRINCE 2 i stawianych im wymagań informacyjnych
- Inwentaryzacja dokumentów stosowanych w firmie na potrzeby zarządzania i realizacji projektów
- Przeprowadzenie analizy porównawczej w celu określenia dokumentów:
  - zbędnych
  - nie wymagających zmian
  - wymagających istotnych zmian
- Stworzenie bazy wzorcowych formularzy
- Przygotowanie przykładowo wypełnionych dokumentów
- ...

#### **5.4. Biuro Programów i Projektów (BPiP)**

Faktyczne powodzenie wdrożenia PRINCE 2 zależy będzie w dużym stopniu od tego, jak dobrze będą przygotowane osoby, które będą ją stosować w codziennej praktyce, oraz jakie wsparcie im zapewnimy w przypadku pojawienia się wątpliwości lub potrzeby interpretacji metodyki.

W projektach zarządzanych według PRINCE 2 taką rolę spełnia sekretariat lub biuro projektu. W organizacjach realizujących wiele projektów zamiast powoływania wielu niezależnych biur, wskazane jest utworzenie jednego biura programów i

projektów, którego zadaniem będzie wspieranie wszystkich projektów realizowanych w firmie na jej potrzeby, lub na potrzeby jej klientów. W przypadku większych projektów na czas ich realizacji do pełnienia funkcji sekretariatu lub biura projektu można oddelegować pracowników BPiP.

### ***Zadania Biura programów i projektów***

Głównym zadaniem biura programów i projektów jest pomoc administracyjna dla szefów programów i projektów, kierowników zespołów i pracowników je realizujących oraz dla ich audytorów, a w szczególności:

- usystematyzowanie i sformalizowanie procesu dochodzenia do decyzji uruchomienia projektów
- porady i interpretacja zasad zarządzania projektami, wymagań jakościowych i standardów technicznych
- zakładanie i obsługa dokumentacji projektów
- pomoc szefom programów i projektów w opracowywaniu planów i ich aktualizacji
- administrowanie zmianami wprowadzanymi w programach i projektach
- administrowanie procesem oceny jakości
- administrowanie procesem kontroli realizacji budżetów projektów
- opracowywanie wzorów dokumentów i procedur ich wykorzystywania
- określanie zasad i wspomaganie zarządzania konfiguracją
- zbieranie faktycznych danych o postępie prac oraz o prognozowanym dalszym ich przebiegu
- pomoc przy redagowaniu sprawozdań
- gromadzenie doświadczeń (nauczek) płynących z realizacji programów i projektów
- kształtowanie kultury pracy opartej na kulcie jakości i poszanowaniu partnerów i współpracowników

### ***Biuro projektu „Wdrożenie PRINCE 2”***

Także projekt „Wdrożenie PRINCE 2” powinien mieć swój sekretariat lub biuro. Ponieważ zgodnie z wcześniejszymi zaleceniami ten projekt powinien być od początku prowadzony zgodnie z PRINCE 2, biuro projektu może stać się załącznikiem przyszłego BPiP. Realizując zadania projektu pracownicy przyszłego BPiP powinni pełnić rolę biura projektu. Dzięki temu zdobędą wiedzę i doświadczenie praktyczne w stosowaniu tej metodyki, opracują wzorce dokumentów, które będą później wykorzystywane przez innych.

Podobnie inni uczestnicy tego projektu, realizując zadania projektu lub uczestnicząc w zarządzaniu jego realizacją, powinni zdobyć doświadczenie, które natychmiast mogą wykorzystać w innych projektach. Dotyczy to kierownictwa organizacji, kierowników projektów i członków zespołów zadaniowych. Wielu z nich będzie w przyszłości użytkownikami produktów tego projektu. Uczestnicząc w jego realizacji na przysłowiowej własnej skórze będą więc mogli się przekonać, jak trudna może być rola użytkownika niedostrzeżanego przez zespół projektowy.

## **6.5. Jak wdrażać w firmie metodyczne zarządzanie projektami?**

Podejście produktowe pozwoliło nam określić wynik projektu i to, co się na ten wynik składa. Problemem pozostaje jednak, w jaki sposób wykonać te produkty, czyli jak zorganizować proces zmian.

Przystępując do wdrażania PRINCE 2 musimy podjąć decyzję, czy zmiany będziemy wprowadzać rewolucyjnie, czy ewolucyjnie.

W przypadku półproduktów technicznych, tj.: wzorce dokumentów, opisy procesów i procedur, instalacja oprogramowania wspomagającego zespoły projektowe, głównym problemem jest zdobycie środków na ich zakup lub wykonanie. Jeśli mamy potrzebne na ten cel środki finansowe, półprodukty takie można kupić w krótkim czasie.

Należy jednak pamiętać, że nawet gdybyśmy, posiadając umiejętności Harry'ego Pottera, w jednej chwili mogli te środki uzyskać i zainstalować, to nie wystarczy, by metodyka zarządzania projektami została faktycznie wdrożona. Nawet najlepsze wzorce, systemy i programy komputerowe nie dadzą oczekiwanych wyników do czasu, aż będą właściwie stosowane przez właściwie przygotowanych ludzi.

Wynika z tego, że oprócz zdobycia lub wykonania środków technicznych, takich jak: opisane procedury, wzorce dokumentów itp., potrzebnych do metodycznego zarządzania projektami, musimy jeszcze do tego przygotować ludzi.

W procesie „uczenia się” wyróżnia się cztery stany:

- a) nieświadomą niekompetencję
- b) świadomą niekompetencję
- c) świadomą kompetencję
- d) podświadomą kompetencję.

Wprawdzie niektórym decydom jest trudno przyjąć do wiadomości, że czegoś nie wiedzą i preferują oni stan „nieświadomej niekompetencji”, to jednak z



doświadczenia wynika, że najtrudniej jest przejść od stanu, kiedy już wiemy, jak coś należy robić (świadoma kompetencja) do stanu, kiedy zdobyte umiejętności wykorzystujemy w sposób podświadomy (podświadoma kompetencja). Wynika to z tego, że człowiekowi najtrudniej przychodzi zmiana przyzwyczajzeń.

Wdrażając metodyczne zarządzanie projektami musimy więc pamiętać, że nie wystarczy, by pracownicy znali wybraną metodykę i wiedzieli, jak się ją stosuje. Wdrożenie polega na tym, że stosowanie takiej metodyki staje się standardem. Ponieważ w każdej organizacji zwykle jest jakaś metodyka zarządzania projektami, nawet jeśli nie jest ona sformalizowana, wdrażając PRINCE 2 musimy pamiętać, że wdrożenie polega na zmianie panujących zwyczajów, procesów, stosowanych narzędzi i dokumentów. W zależności od zakresu koniecznych zmian należy wybrać sposób ich wprowadzania. W skrajnych przypadkach może to być podejście rewolucyjne, polegające na wprowadzeniu głębokich zmian w krótkim okresie, lub podejście ewolucyjne, polegające na stopniowanym wprowadzaniu określonych zmian, które wszystkie razem dadzą oczekiwany wynik. Zmiany rewolucyjne dają pozytywny wynik wtedy, kiedy ludzie, których te zmiany dotyczą, są dobrze do nich przygotowani i chcą, by one zaszczyły.

Ponieważ tylko nieliczne instytucje w Polsce dysponują odpowiednio przygotowaną kadrą, metodyka PRINCE 2 powinna być wdrażana ewolucyjnie, ale z wyraźnym podziałem na fazy. Trzeba unikać przypadku kiedy kilka projektów rozpoczynanych w tym samym okresie jest zarządzanych według różnych metodyk.

W ewolucji systemów zarządzania projektami można wyróżnić następujące stany i punkty przełomowe:

1. Niektóre narzędzia lub techniki są stosowane przez pojedyncze osoby
2. Formalne wprowadzenie wybranych metod i narzędzi metodycznego zarządzania projektami
3. Stopniowy wzrost zainteresowania naczelnego kierownictwa doborem projektów do realizacji i ich integracją ze strategią firmy, zgodnie z ustalonymi priorytetami
4. Uruchamianie wszystkich nowych projektów zgodnie z PRINCE 2
5. Zamykanie projektów rozpoczętych według dotychczasowej metodyki, które powinny się zakończyć w terminie do 3 miesięcy od dnia przyjęcia metodyki PRINCE 2 za obowiązującą, bez wprowadzania zmian w ich zarządzaniu
6. Stopniowe wbudowywanie elementów PRINCE 2 w zarządzanie projektami, które rozpoczęto według dotychczasowej metodyki, a których termin zakończenia przekracza 3 miesiące od przyjęcia metodyki PRINCE 2.
7. Stosowanie metodyki PRINCE 2 we wszystkich projektach.

## 6.1. Plan działań organizacyjnych

Oprócz planu projektu, w którym określono działania służące wytworzeniu produktów składających się na wynik „wdrożenie PRINCE 2” poniżej przedstawiono działania służące wytworzeniu jednego z „półproduktów”, który określiliśmy wcześniej jako „ludzie przygotowani do stosowania metodyki PRINCE 2”. Działania te można zrealizować w następującej kolejności:

1. Przekonać zarząd o celowości wdrożenia metodycznego zarządzania projektami według PRINCE 2
2. Wybrać lidera przemian, czyli osobę, która jest uznawana przez współpracowników za autorytet w zakresie zarządzania projektami
3. Podjąć decyzję odnośnie wyboru firmy doradczej lub realizacji własnymi siłami (udział osób spoza danej organizacji sprzyja zmianie kultury organizacyjnej i przełamywaniu występujących zwykle barier.)
4. Przeszkolić lidera przemian w zakresie metodyki zarządzania projektami PRINCE 2
5. Wybrać grupę kierowników projektów, lub kandydatów do pełnienia tej roli, i przeszkolić ich w zakresie metodyki PRINCE 2
6. Zorganizować wsparcie dla projektów – tworząc biuro programów i projektów
7. Opracować wzorcowy podział zadań dla najbardziej typowych dla organizacji projektów
8. Dobrać narzędzia, techniki i programy komputerowe wspierające zarządzanie projektami, dostosowane do przyjętej polityki, struktury organizacyjnej i procesów zarządzania projektami
9. Przeszkolić osoby biorące udział w zarządzaniu projektami z ramienia kierownictwa organizacji (członkowie komitetów sterujących i audytorzy projektów) i włączyć ich w proces zarządzania projektami
10. Przeszkolić członków zespołów realizacyjnych w zakresie metodycznego zarządzania projektami
11. Stworzyć forum zarządzania projektami
12. Podjąć decyzję o obowiązku stosowania metodyki PRINCE 2 z równoczesnym określeniem czasu trwania okresu przejściowego dla projektów już realizowanych o terminie zakończenia ponad trzy miesiące od dnia podjęcia tej decyzji
13. Stopniowo obejmować metodycznym zarządzaniem coraz mniejsze przedsięwzięcia
14. Wprowadzić macierzową strukturę organizacyjną, łączącą elementy struktury funkcjonalnej z organizacją zadaniową

15. Przekształcić organizację w organizację zarządzaną przez projekty (Przez zarządzanie przez projekty rozumiemy stosowanie przyjętej metodyki zarządzania projektami także w odniesieniu nawet do pozornie prostych zadań, na przykład: do zarządzania zadaniami realizowanymi w ramach większych projektów.)

## 7. Uwagi praktyczne

Świadomie w tym opracowaniu nie wspomniano o harmonogramie realizacji projektu i jego kosztach. Wynika to z istotnego ich uzależnienia od początkowego stanu organizacji i kultury realizacji projektów, a także od dysponowania odpowiednio przygotowanymi ludźmi.

Na podstawie obserwacji procesu wdrażania PRINCE 2 w kilku polskich firmach można sformułować następujące zalecenia:

1. Kandydaci na kierowników projektów, którzy jako pierwsi będą stosować PRINCE 2, powinni nie tylko mieć doświadczenie w zarządzaniu projektami, ale także powinni poznać dobrze tę metodykę, umieć negocjować, potrafić rozwiązywać konflikty i komunikować się z innymi. Takie umiejętności pozwolą im pełnić rolę promotorów nowych rozwiązań i w ten sposób pozyskiwać zwolenników dla nowej metodyki.
2. Opracowanie wzorcowego podziału zadań dla najbardziej typowych projektów jest istotnym elementem procesu zarządzania zmianami. Zespoły złożone z kierowników projektów, kierowników funkcjonalnych, personelu i ewentualnie konsultanta, analizując projekty typowe dla organizacji i poszukując jednolitych ról i związanych z nimi zadań, które odpowiadałyby potrzebom większości realizowanych projektów, ujednolicają sposób myślenia o projektach. Wykonanie takiej pracy stanowi istotny element procesu zarządzania zmianą i służy kształceniu członków zespołów projektowych.
3. Skuteczną formą zarządzania zmianami jest stworzenie forum wymiany doświadczeń w dziedzinie zarządzania projektami. Może mieć to formę spotkań dyskusyjnych lub internetowej grupy dyskusyjnej, służących:
  - o dyskusowaniu problemów i wymianie doświadczeń
  - o omawianiu sukcesów i porażek
  - o proponowaniu nowych rozwiązań i zmian organizacyjnych
  - o promowaniu zarządzania przez projekty
  - o popularyzowaniu metod, technik i narzędzi użytecznych w zarządzaniu projektami.
4. Dobierając narzędzia i programy komputerowe wspierające zarządzanie projektami należy pamiętać, że powinny być one zgodne z przyjętą metodyką



oraz ułatwiać pracę osób zarządzających projektami i wykonujących zadania w projektach. Należy unikać sytuacji, kiedy rozwiązania organizacyjne dostosowywane są do posiadanych narzędzi.

5. Przystępując do wdrażania metodyki PRINCE 2 warto uświadomić wszystkim zainteresowanym, że to wdrożenie powinno zaowocować:
  - o ułatwieniem pracy i wzrostem produktywności poszczególnych pracowników, zespołów wykonawczych i całej organizacji
  - o ukształtowaniem nowej kultury pracy opartej na myśleniu procesowym i zarządzaniu przez projekty
  - o stanowić istotny krok w procesie przekształcania organizacji w samouczącą się organizację.
6. Dla powodzenia projektu konieczne jest włączenie naczelnego kierownictwa w proces zarządzania projektami, gdyż:
  - o zmiany, które nie mają wsparcia ze strony naczelnego kierownictwa, zwykle kończą się niepowodzeniem.
  - o zmiana struktury organizacyjnej na strukturę macierzową lub projektową wymaga decyzji na najwyższym szczeblu.
  - o metodyczne zarządzanie projektami nakłada na członków naczelnego kierownictwa szczególną rolę, np. jako przewodniczących komitetów sterujących, dlatego ta rola musi być im znana.
7. Decydując się na wdrożenie metodyki PRINCE 2 bez pomocy firmy doradczej, w pierwszej kolejności należy zadbać o to, by pozyskać pracownika, który tę metodykę znał i stosował. W firmach, które zdecydowały się na wdrażanie PRINCE 2 własnymi siłami, korzystając jedynie z podręczników, na ogół wdrożenia zakończyły się niepowodzeniem, lub wdrożeniem niezgodnym z zamierzonym celem.

## Literatura

1. „*Managing Successful Projects with PRINCE 2*”, CCTA, London, Stationary Office. 1999
2. K. Bradley „*Podstawy metodyki PRINCE 2*”, Centrum Rozwiązań Menedżerskich S.A. Warszawa. 1999

Autor:

Dr inż. Wiesław KOSIERADZKI

Centrum Rozwiązań Menedżerskich S.A. Warszawa

ul. Pożarowa 9/11 Tel. (0-22) 811-44-40

e-mail: w.kosieradzki@crm.com.pl



# METODYKI WDRAŻANIA KM

Kazimierz KRUPA

**Streszczenie:** Jednym z zasadniczych problemów skutecznego zarządzania wiedzą jest wykorzystanie profesjonalnych narzędzi wdrożeniowych. Proces wdrożenia KM zaliczamy do kluczowych czynników sukcesu. Opracowano wiele metodyk wspomagających implementację zarządzania wiedzą. Wybór właściwej, najbardziej adekwatnej dla określonego podmiotu gospodarczego ułatwia kompleksowa ich charakterystyka.

## 1. Skuteczne zarządzanie wiedzą w podmiotach gospodarczych należy do bardzo trudnych zadań

Jednym z ważnych i szczególnych zadań Zarządzania Wiedzą jest wspomaganie handlu i marketingu. W tym głównie umożliwienie zrozumienia danych dotyczących wyborów rynkowych klientów i wykorzystanie ich do budowy pętli decyzyjnych oraz wtórnych pętli poznawczych. Wyrafinowane bazy wiedzy a także rozwój metod inteligentnych umożliwiły bardzo duży postęp w CRM. Zarządzanie wiedzą (ZW<sup>1</sup>) należy do najtrudniejszych, jednocześnie najbardziej obiecujących, aspektów kreatywnej aktywności podmiotów gospodarczych. Opiera się ono na założeniu, że wiedza osobowa i bezosobowa, zgromadzona w głowach, dokumentach, schematach, procedurach, regułach, modelach, może poprawić współpracę personelu, zredukować redundancję, zmobilizować pracowników do dzielenia się wiedzą, co w efekcie pozwoli na powstanie nowych wartości. Wdrażanie ZW nie należy do zadań prostych, przekonywano o tym również w czasie seminariów na Targach Knowledge Management World 2001 w Santa Clara Convention. Larry Prusak z IBM twierdzi, że z kilkuset wdrożeń, które analizował większość była nietrafna, bowiem koncentrowano się głównie na aspektach technicznych i technologicznych<sup>2</sup>. Wydaje się, że prawidłowy, bardzo skodyfikowany proces wdrażania ZW należy rozpocząć od:

- specyfikacji wiedzy którą dysponujemy,
- zweryfikowania potrzeb w tym zakresie,
- wybrania lub zbudowania właściwej metodyki wdrożeniowej,
- opracowania spójnego systemu nagród za dzielenie się wiedzą.

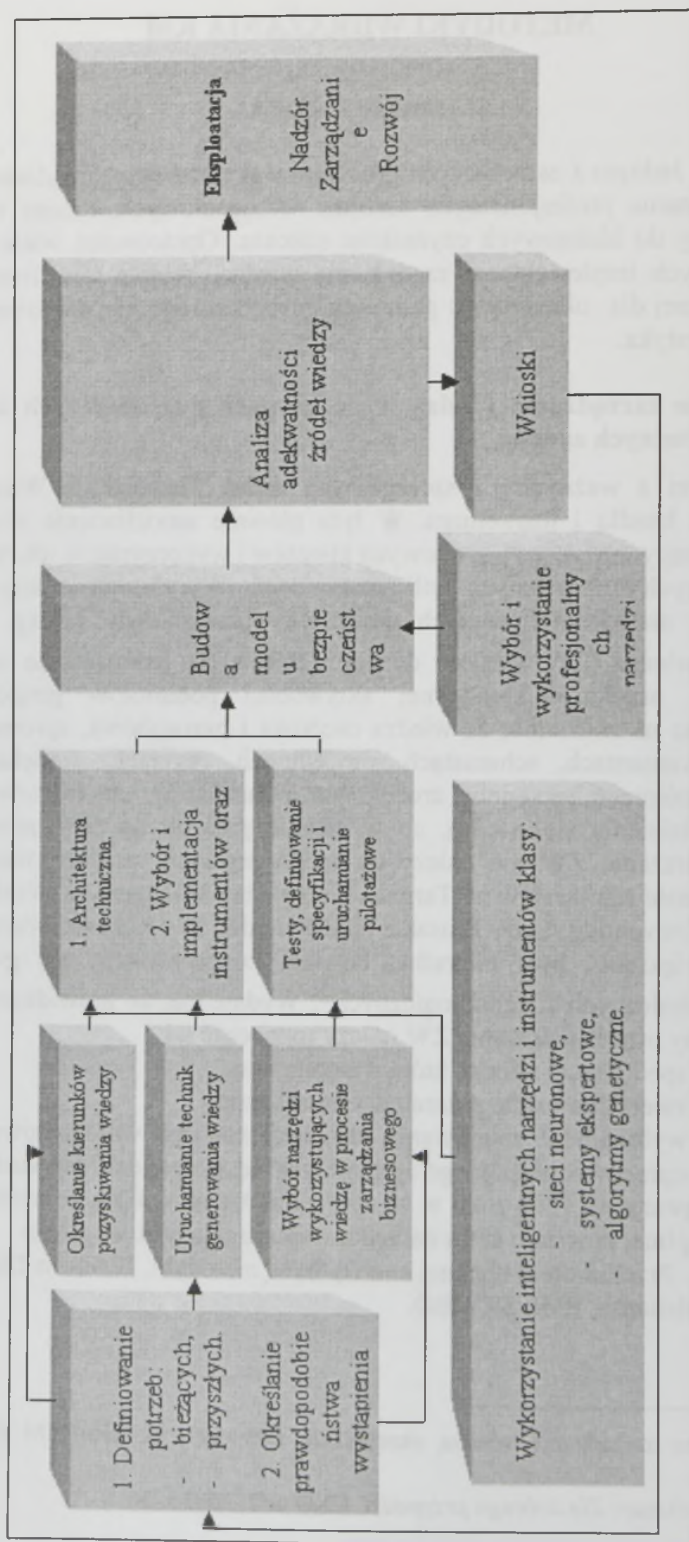
Począwszy od 1994 roku w którym zarządzanie wiedzą pojawiło się jako osobna dyscyplina, powstało kilka narzędzi wspomagających jej proces wdrożeniowy. Przedmiotem obecnej analizy będą metodyki: Business Dimensional Lifecycle, Applehansa, PMI, STWBO.

---

<sup>1</sup> W literaturze zarządzanie wiedzą określa się zazwyczaj skrótem KM (Knowledge Management).

<sup>2</sup> Zobacz E. Berkman: *Złe dobrego przypadki*. CXO nr 2, 2001 s. 30.



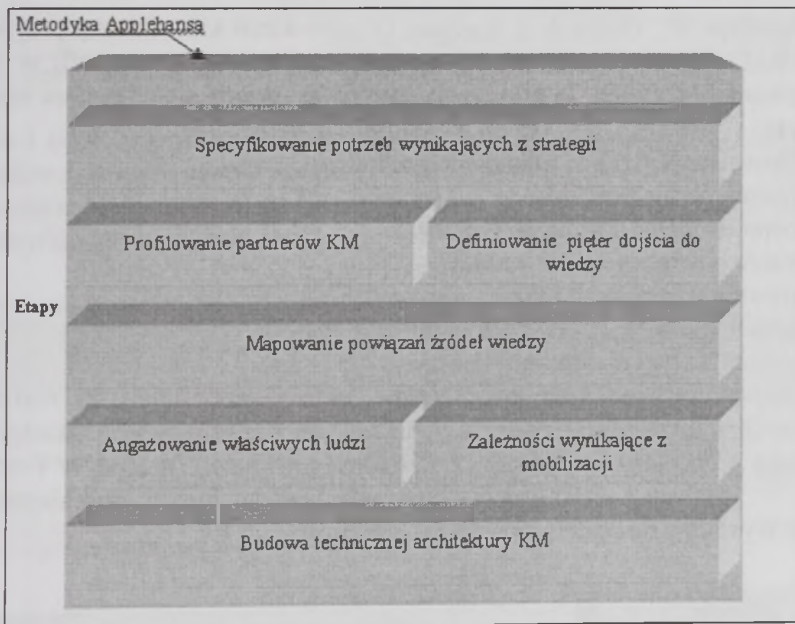


Rys. 1. Metoda budowy bazy wiedzy Business Dimensional Lifecycle  
 Źródło: Opracowanie własne na podstawie [19 s. 5]

## 2. Metodyka Business Dimensional Lifecycle

Zdaniem ekspertów Delphi Group w skutecznym zarządzaniu wiedzą bardzo dobrze zdaje egzamin koncepcja warstwowa i wykorzystanie metodyki Business Dimensional Lifecycle (BDL). Metodyka BDL umożliwia kompleksowe zarządzanie wiedzą oraz wyrafinowane kodyfikowanie (codification strategy), i składa się z dwunastu narzędzi (rys. 1), które służą do:

1. Definiowania potrzeb.
2. Określania prawdopodobieństwa ich wystąpienia.
3. Ustalenia kierunków pozyskania wiedzy.
4. Uruchamiania technik generowania wiedzy.
5. Selekcji narzędzi wykorzystujących wiedzę w procesie zarządzania biznesowego.
6. Wyboru inteligentnych narzędzi i instrumentów wspomagających pozyskiwanie wiedzy.
7. Budowy architektury technicznej.
8. Definiowania testów i specyfikacji.
9. Budowy modelu bezpieczeństwa.
10. Analizy adekwatności źródeł wiedzy.
11. Eksploatacji zgromadzonej wiedzy.
12. Formułowania wniosków [19 s. 4-6].



Rys. 2. Metodyka Applehansa – etapy i ich zależności  
Źródło: Opracowanie własne na podstawie [2 s.8]

Szczegółowość metodyki BDL (która ma cechy modelu linearnego) w zakresie pozyskiwania wiedzy od personelu, ekspertów, konsultantów oraz wiedzy bezosobowej, z źródeł wewnętrznych i zewnętrznych nie jest zbyt duża. Metodyka BDL wspomaga kilka podfunkcji ZW. Była ona zweryfikowana z pozytywnym skutkiem w licznych podmiotach gospodarczych. Uwzględnia jednak w małym zakresie elementy „miękkie” i wymagania personelu, który może być w bardzo wielu przypadkach źródłem wiedzy lub jej odbiorcą. Zakres wykorzystania instrumentów wspomagających proces wdrożeniowy w tym: procedury, modele, narzędzia informatyczne, systemy inteligentne (algorytmy genetyczne, sieci neuronowe, pakiety ekspertowe) jest ograniczony. Metodyka BDL posiada pełną kompletność – wspomaga cały cykl ZW. Ma możliwość instalacji narzędzi dzielenia się wiedzą (portale, forum dyskusyjne, foldery, listy dyskusyjne, blogi, narzędzia klasy groupware, narzędzia do współpracy). Dostarcza również instrumenty zachęcające do korzystania z zgromadzonej wiedzy oraz pełny system oceny wartości wiedzy, nie dysponuje natomiast pakietem nagród za dzielenie się wiedzą, do którego należą między innymi punkty i ich zamiana na z góry ustalone w specjalnym regulaminie środki finansowe, bonusy. Pozwala natomiast na budowę modelu bezpieczeństwa zgromadzonej i przechowywanej wiedzy oraz umożliwia generowanie efektu kumulacyjnego.

### 3. Metodyka Applehansa

Applehans W., Globe A. i Langer G. prowadzili kilkadziesiąt projektów wdrażania KM. Doświadczenia zaprezentowali w obszernej monografii w 1998 [2]. Opracowana przez nich interaktywna metodyka składa się z siedmiu etapów (rys. 2). Są to:

1. Specyfikowanie potrzeb wynikających z strategii.
2. Profilowanie partnerów KM.
3. Definiowanie piętér dojścia do wiedzy.
4. Mapowanie powiązań źródeł wiedzy.
5. Angażowanie właściwych ludzi.
6. Badanie zależności wynikających z mobilizacji.
7. Budowa technicznej architektury KM.

Każdy z etapów posiada kompletny zestaw zadań koniecznych do realizacji. Tabela 1 zawiera specyfikację potrzeb wynikających z zdefiniowanej strategii. Są to: a/ Budowa procedury dochodzenia od strategii informacji do KM, b/ Portfolio zawartości – sprawne, automatyczne definiowanie co jest najistotniejsze, c/ Ponawianie wysiłków na polu tworzenia architektury

Tablica 1. Specyfikowanie potrzeb wynikających z strategii

Lp.	Nazwa zadania koniecznego do realizacji
1	Budowa procedury dochodzenia od strategii informacji do KM
2	Portfolio zawartości – sprawne, automatyczne definiowanie co jest najistotniejsze



3	Ponawianie wysiłków na polu tworzenia architektury wiedzy
4	Audit – analiza dopasowania projektu KM do biznesowych obiektów
5	Analiza możliwości inwestowania projektów KM

Źródło: Opracowanie własne na podstawie [2 s. 4]

Tablica 2. Profilowanie ludzi

Lp.	Nazwa zadania koniecznego do realizacji
1	Komu należy utworzyć profil?
2	Jaki profil należy utworzyć?

Źródło: Opracowanie własne na podstawie [2 s. 4] wiedzy, d/ Audit – analiza dopasowania projektu KM do biznesowych obiektów, e/ Analiza możliwości inwestowania projektów KM.

Tabela 2 zawiera determinanty - pytania związane z właściwym „profilowaniem” ludzi właścicieli wiedzy i ludzi odbiorców wiedzy. Są to pytania: *Komu należy utworzyć profil?, Jaki profil należy utworzyć?*

Tablica 3. Kroki dochodzenia do wiedzy. Mapowanie ludzi i zawartości

Krok i	Nazwa zadania koniecznego do realizacji
1	Identyfikacja cyklu strategii i biznesu
2	Mapowanie punktów dźwigni informacyjnych
3	Dołączanie zespołów ekspertów
4	Identyfikacja kontekstu

Źródło: Opracowanie własne na podstawie [2 s. 5]

Tabela 3 zawiera kroki dochodzenia do wiedzy czyli mapowanie ludzi i zawartości. Zdefiniowano tutaj cztery kroki: 1. Identyfikacja cyklu strategii i biznesu. 2. Mapowanie punktów dźwigni informacyjnych. 3. Dołączanie zespołów ekspertów. 4. Identyfikacja kontekstu.

Tablica 4. Mapowanie sieci połączeń źródeł wiedzy

Krok i	Nazwa zadania koniecznego do realizacji
1	Identyfikacja centrów kontekstu
2	Dołączanie zewnętrznych (satelitarnych) kontekstów
3	Kompletowanie personelu, przydzielanie atrybutów i posiadania
4	Architektura wiedzy

Źródło: Opracowanie własne na podstawie [2 s. 5]

Tabela 4 dotyczy mapowania sieci połączeń źródeł wiedzy. Wyodrębniono w niej: identyfikację centrów kontekstu, dołączanie zewnętrznych (satelitarnych) kontekstów, kompletowanie personelu, przydzielanie atrybutów i posiadania, architekturę wiedzy.

Tablica 5. Angażowanie ludzi dla trzech poziomów zarządzania wiedzą

Lp.	Nazwa poziomu zadań zarządzania wiedzą
1	Analizujący wiedzę
2	Autorzy (źródła) wiedzy
3	Zewnętrzne zespoły

Źródło: Opracowanie własne na podstawie [2 s. 6]

Tab. 6. Mobilizowanie kontekstów

Lp.	Nazwa zadania koniecznego do realizacji
1	Unikanie wszystkich indeksów schyłkowych
2	Tworzenie słownika ogólnego
3	Wykorzystanie ogólnych podstaw
4	Typy kontekstów
5	Utrzymanie ustalonej dyscypliny
6	Przydzielanie właścicieli

Źródło: Opracowanie własne na podstawie [2 s. 6]

Tabela 5 poświęcona jest angażowaniu ludzi dla trzech poziomów zarządzania wiedzą. Poziomy te to: 1. Analizujący wiedzę. 2. Autorzy (źródła) wiedzy. 3. Zewnętrzne zespoły.

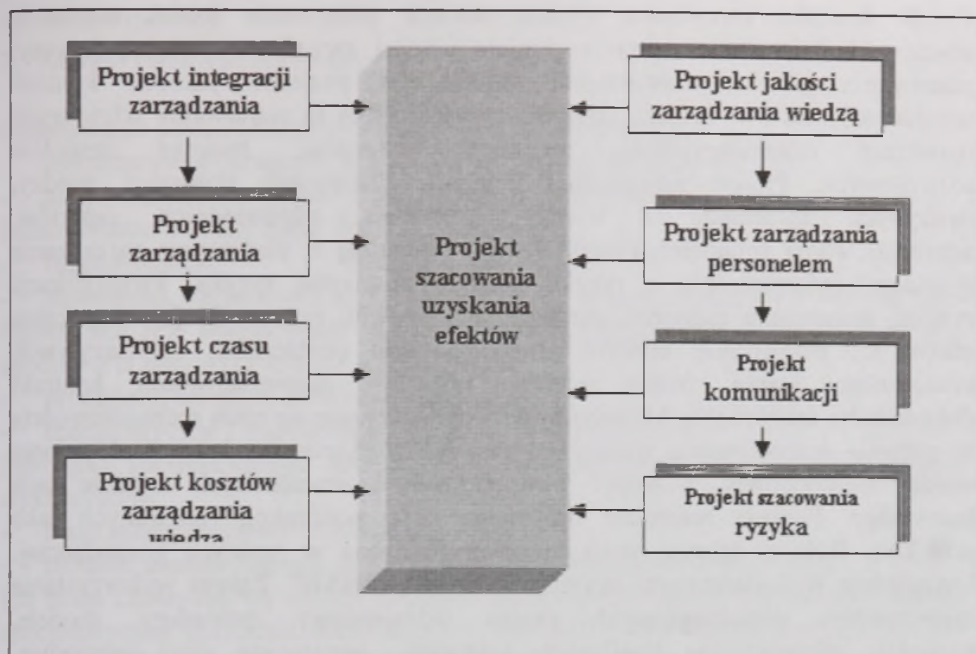
Tabela 6 dotyczy mobilizowaniu kontekstów. Zadania realizowane tutaj to: a/ Tworzenie słownika ogólnego, b/ Unikanie wszystkich indeksów schyłkowych, c/ Tworzenie wyrafinowanych i złożonych (wieloaspektowych) kontekstów, d/ Wykorzystanie ogólnych podstaw, e/ Przydzielanie właścicieli, f/ Utrzymanie ustalonej dyscypliny.

Tablica 7. Budowa kompleksowej technicznej architektury

Poziomy	Nazwa zadania koniecznego do realizacji
1	Dojście do wiedzy
2	Interfejs (sprzęg)
3	Inteligencja
4	Transmisja wiedzy
5	Odpowiedzialność
6	Aplikacje umożliwiające dysponowanie wiedzą

Źródło: Opracowanie własne na podstawie [2 s. 6]

Tabela 7 poświęcona jest aspektom budowy kompleksowej technicznej architektury zarządzania wiedzą. Najważniejsze aspekty w tym zakresie to: 1. Procedury dojścia do wiedzy. 2. Interfejs (sprzęg). 3. Inteligencja. 4. Transmisja wiedzy. 5. Odpowiedzialność. 6. Aplikacje umożliwiające dysponowanie wiedzą.



Rys. 3. Projekty metodyki PMI  
 Źródło: Opracowanie własne na podstawie [1 s. 10]

#### 4. Dziewięć projektów KM – metodyka Project Management Information

Metodyka PMI (Project Management Information) zawiera modele prezentujące sposób przekształcania podmiotów gospodarczych w rezultacie procesu uczenia. Nakreśla procedury konieczne *przed, podczas i po* oraz dostarcza narzędzia i techniki (specyficzne procedury) wykonawcze typu:

1. Ktoś ma już zgodę na uczenie się z twego doświadczenia.
2. Pozwolić sobie na uczenie – czas do rozważań.
3. Uczenie po doświadczeniu – kiedy jest ono wszędzie możliwe.
4. Szukanie właściwych ludzi – kto tylko ma na ten temat wartościową wiedzę [1 s. 11].

Autorzy metodyki PMI twierdzą, że stanowi ona próbę łatwej drogi zmiany kultury organizacji podmiotów gospodarczych. Project Management Information składa się z (jak to nazywa Collison) dziewięciu projektów (rys. 3). Każdy z nich posiada po kilka procedur odcinkowych, które są przedmiotem szczegółowej operacjonalizacji oraz pomiaru. Projekt integracji zarządzania wiedzą składa się z modułów: rozwoju, realizacji, kontroli integracji zmian. Projekt zakresu zarządzania wiedzą dotyczy: intencji, planowania zakresu, zmian kontroli zakresu, definicji zakresu, weryfikacji określonego zakresu. Projekt czasu to: definicja aktywności, sekwencje aktywności, zestawienie rozwoju, zestawienie kontroli.



Projekt kosztów zarządzania wiedzą zawiera: planowanie źródeł, estymacje kosztów, budżetowanie, kontrolę. Projekt jakości zarządzania wiedzą dotyczy: planowania jakości, zapewnienia określonego poziomu jakości, kontroli uzyskanych efektów. Projekt zarządzania personelem to planowanie właściwych rozwiązań organizacyjnych, znajomość personelu, budowa zespołów rozwojowych. Projekt komunikacji dotyczy: planowania transmisji wiedzy, dystrybucji informacji i wiedzy, tworzenia odpowiednich raportów, administrowania zmianami. Projekt ryzyka składa się z: planowania zarządzania ryzykiem, informowania o ryzyku, analizy pomiarów ryzyka, kwantyfikacji ryzyka, planowania zagrożeń, zarządzania i kontroli ryzyka. Projekt uzyskania efektu to: planowanie efektów, rozwiązywanie problemów, rozwiązywanie konieczności starań, źródła selekcji, zawartość administrowania, kontrakt dokumentów źródłowych. Metodyka PMI charakteryzuje się małą szczegółowością w zakresie pozyskiwania wiedzy od personelu, ekspertów, konsultantów oraz wiedzy bezosobowej, z źródeł wewnętrznych i zewnętrznych a także tacit knowledge. Posiada natomiast optymalną ilość podfunkcji (nazwanych jako projekty). Była z dobrym skutkiem zweryfikowana w praktyce gospodarczej. Uwzględnia w dostatecznym stopniu elementy „miękkie”. Zakres wykorzystania instrumentów wspomagających proces wdrożeniowy: procedury, modele, narzędzia informatyczne, inteligentne (algorytmy genetyczne, sieci neuronowe, pakiety ekspertowe) jest bardzo mały. Kompletność metodyki jest odpowiednia. Nie posiada ona jednak narzędzi do dzielenia się wiedzą i instrumentów zachęcających do korzystania z zgromadzonej wiedzy. System oceny wartości wiedzy (wg kryterium: stopnia korzystania, ilości wejść, udzielania punktów za wykorzystanie) oraz system nagród za dzielenie się wiedzą (punkty i ich zamiana na z góry ustalone nagrody materialne lub bonusy) będą dopiero budowane. Brak w niej narzędzi pozwalających na budowę modelu bezpieczeństwa zgromadzonej i przechowywanej wiedzy<sup>3</sup>.

## 5. Metodyka sekwencyjnego tworzenie bazy wiedzy organizacji

Metodyka sekwencyjnego tworzenie bazy wiedzy organizacji (STBWO) jest klasycznym modelem interaktywnym i należy do grupy inteligentnych narzędzi wspomagających realizację głównych zadań Knowledge Management. STBWO wykorzystuje kilkanaście specjalistycznych narzędzi i składa się z siedmiu etapów wykonawczych. Są to:

1. Generowanie potrzeb.
2. Poszukiwanie źródeł wiedzy.
3. Gromadzenie wiedzy.
4. Przechowywanie źródeł i wiedzy wraz z inteligentną klasyfikacją wg ustalonych kryteriów.

---

<sup>3</sup> Więcej zobacz w *PMBOK - Project Management Body of Knowledge* (1997). Q&A. Business Management, NY

5. Przetwarzanie.
6. Wykorzystanie
7. Model: kwantyfikacji, motywowania i oceny<sup>4</sup>.

Generowanie potrzeb może być przez podmiot gospodarczy:

- uświadamiane (strategia, oczekiwania rynku, globalizacja),
- nie uświadamiane (cyberkultura, wyzwania zewnętrzne).

Poszukiwanie wiedzy może dotyczyć źródeł:

- bezpośrednich,
  - pośrednich, szczególnie zewnętrznych traktowanych jako zasób wiedzy per se.
- System ten jest jednak mało kognitywny. W przechowywaniu źródeł i wiedzy zgodnie z tą metodyką można wykorzystać:

- procedury, intencje, dane nienumeryczne,
- specjalistyczne i inteligentne algorytmy do klasyfikacji.

Metodyka sekwencyjnego tworzenia bazy wiedzy organizacji w przetwarzaniu wykorzystuje:

- mało skomplikowane algorytmy,
  - modele, analizy regresji, drzewa decyzyjne, narzędzia ekstrapolacji,
- lub może cechować się dużym stopniem złożoności, a także eksploatować bardzo specjalistyczne narzędzia matematyczne, narzędzia do budowy kontekstów oraz instrumenty ułatwiające komparatystkę wiedzy wewnętrznej i zewnętrznej.

Wykorzystanie wiedzy może być bezpośrednie lub pośrednie oraz może uwzględniać personalizację, uznane formaty transmisji (XML, HTML WWW), programy użytkowe (BI, OLAP) i służyć do wspomaganie podejmowania decyzji. Wzrost skuteczności i efektywności metodyki STBWO umożliwia wykorzystanie pętli zamkniętej wraz z sprzężeniami zwrotnymi. Pętla zamknięta dotyczy etapów:

- pierwszego i drugiego,
- drugiego i szóstego,
- czwartego i piątego,

Natomiast etapy: trzeci i czwarty oraz piąty i szósty posiadają możliwości sprzężeń zwrotnych.

Każdy z etapów metodyki STBWO eksploatuje specjalistyczne metody i techniki w zależności od:

- okresu czasu,
- zakresu aktywności,
- realizowanej strategii,
- potencjału personelu,
- oceny szans i zagrożeń otoczenia,
- etapu rozwoju organizacji i jej kultury,
- poziomu kultury informacyjnej personelu.

Szczegółowość metodyki sekwencyjnego tworzenia bazy wiedzy organizacji w zakresie pozyskiwania wiedzy od personelu, ekspertów,

<sup>4</sup> Podobną koncepcję prezentują specjaliści KPMG Consulting wyróżniając pięć faz ZW [Research Report 2000 s 22 - 23].

konsultantów oraz wiedzy bezosobowej, z źródeł wewnętrznych i zewnętrznych jest znaczna. Metodyka ta wyróżnia się również dużą ilością podfunkcji. Uwzględnia elementy „miękkie” i mierzy MBTI. Zwraca uwagę na cechy charakterystyczne personelu który może być źródłem wiedzy lub jej odbiorcą. Zakres wykorzystania instrumentów wspomagających proces wdrożeniowy w metodyce STBWO obejmuje: procedury, modele, narzędzia informatyczne, systemy inteligentne w tym algorytmy genetyczne, sieci neuronowe, pakiety ekspertowe. Jest to metodyka kompletna – wspomaga zdobywanie wiedzy, jej kreowanie, przetwarzanie, segmentację, wykorzystanie, eksport.

Posiada narzędzia do dzielenia się wiedzą:

- portale: firmowe (ogólne, dziedzinowe), ogólnodostępne,
- forum dyskusyjne,
- foldery,
- listy dyskusyjne,
- narzędzia klasy groupware,
- narzędzia do współloceny.

Metodyka STBWO dostarcza również instrumenty zachęcające do korzystania z zgromadzonej wiedzy oraz system:

- oceny wartości wiedzy,
- nagród za dzielenie się wiedzą, na który składają się punkty, bonusy i premie finansowe.

## **Wnioski**

Zaprezentowane metodyki wdrażania ZW zostały opracowane przez różne zespoły badawcze posiadające bogate doświadczenia w zakresie kreatywnego wykorzystania wiedzy w prowadzeniu biznesu. Analiza głównych cech tych metodyk może ułatwić dobór właściwego narzędzia do implementacji KM w każdym podmiocie gospodarczym.

## **Literatura**

1. A Guide to the Project Management Body of Knowledge. PM Book Guide. PMI, 2001
2. Applehans W., Globe A., Langer G: Managing Knowledge. A Practical Web – Based Approach. Addison – Wesley Information Technology Series. N.Y, 1998
3. Brook A. : Corporate Memory : Strategies for Knowledge Management. International Thomson Business Press, 1998
4. Carter L., Harkins P., Tmmines A., Onge H.: Linkage, Inc.’s Best Practices in Knowledge Management &Organizational Learning Handbook. Linkage, Ma, 2001
5. Collison CH., Parcell G.: Learning to Fly: Practical Lesson from one of the World’s Learning Knowledge Companies. Capstone Publishing Ltd., UK, 2001



6. Dixon N.M.: Common Knowledge. How Companies Thrive by Sharing What They Know. Harvard Business School Press, 2000
7. Foster I., Kesselman C.: The Grid - Blueprint for a New Computing Infrastructure. Morgan Kaufman, 1999
8. Hattan K., Rosenthal S.: Reaching for the Knowledge Edge: How the Knowing Corporation Seeks, Stares & Uses Knowledge for Strategic Advantage. AMACOM, NY, 2001
9. Knowledge Discovery for Business Information Systems. Red. W. Abramowicz, J. Zurada. Kluwer Academe Publishers, Boston, Dordrecht, London, 2001
10. Knowledge Management Research Center. CIO Magazine, 2001
11. Koulopoulos T.M., Frappado C.: Smart Things to Know About, Knowledge Management. Capstone Publishing Ltd. 2001
12. Krupa K.: Metodyka STBWO i akcelerator ekspansji. Red. P.W. Fuglewicz, J.K. Grabary. W Informatyka w gospodarce wiedzy. WNT, Warszawa, 2001
13. Liebowitz J.: Building Organizational Intelligence. A Knowledge Management Primer. 1999
14. Liebowitz J.: Knowledge Management Handbook. International Thomson Business Press, NY, 1999
15. Lei D., Slocum J. Building Cooperative Advantage: Managing Strategic Alliances to Promote Organizational Learning. Journal of World Business, no 3, 1997
16. Miller W., Morris L.: Fourth Generation R & D : Management Knowledge Technology, and Innovation. John Wiley & Sons, Inc. NY, 2001
17. Pfeffer J., Satton R.I.: The Knowing – Doing Gap: How Smart Companies Turn Knowledge into Action. Harvard Business School Press, 2000
18. Tiwan A.: The Knowledge Management. Toolkit. Principal Technical for Building a Knowledge Management System. Prentice Hall PTR, NJ, 2000
19. Business Intelligence. Materiały informacyjne IDG. Warszawa 2001  
[www.cio.com](http://www.cio.com)  
[www.kmtoolkit.com](http://www.kmtoolkit.com)

dr inż. Kazimierz Krupa  
 Uniwersytet Rzeszowski  
 email [kkrupa@pf.pl](mailto:kkrupa@pf.pl)



# METODY WDRAŻANIA SYSTEMÓW INFORMATYCZNYCH W URZĘDZIE MIASTA I GMINY

Jarosław ŁADYGA

**Streszczenie:** Praca jest poświęcona naszkicowaniu koncepcji podejścia do informatyzacji gminy oraz administracji gminnej w oparciu o doświadczenie wdrożeniowe w Łomiankach. Projekt wciąż trwa i zaczyna dopiero przynosić wymierne rezultaty. Jego istotna część – kompleksowy system informatyczny urzędu, rozpoczyna dopiero fazę wdrożeniową. W moim przekonaniu przedstawiona koncepcja i metoda są słuszne i warto je zaprezentować już teraz. W trakcie dotychczasowych działań popełniono szereg błędów. Przedstawiam je po to, aby ewentualni naśladowcy mogli ich uniknąć. Wdrożenie w Łomiankach nie jest wybitnym osiągnięciem z punktu widzenia efektywności wdrażania projektów IT. Samorząd gminny jest jednak szczególnie dla informatyki sferą, posiadającą szereg uwarunkowań nie występujących gdzie indziej i – powiedzmy szczerze – często absurdalnych. Nie można jednak ich pomijać lub nie uwzględniać, jeśli wdrożenie ma przebiegać zgodnie z regulacjami prawnymi.

## Wstęp

Gmina zajmuje szczególne miejsce w systemie funkcjonowania samorządu terytorialnego w naszym kraju. Jest to, bowiem jedyny byt rzeczywisty. Gmina jest wspólnotą mieszkańców zasiedlających określony, niewielki z reguły obszar, poddający się podświadomemu postrzeganiu i dostępny wyobraźni człowieka. Pozostałe szczeble, powiat i województwo to byty na swój sposób „wirtualne”. Ich istnienie nie ma w powszechnym odczuciu wielkiego związku z rzeczywistością w miejscu zamieszkania. Przeciętny obywatel w praktyce nie ma najmniejszego wpływu na decyzje zapadające na szczeblu powiatowym lub wojewódzkim. Co więcej – na ogół nie wie nawet, kto i dla czego podejmuje takie, a nie inne decyzje. W gminie związek władzy samorządowej z mieszkańcami jest znacznie bardziej bezpośredni i czytelny.

W wyborach wspólnota wybiera swoich przedstawicieli – Radę Gminy, lokalny odpowiednik Sejmu, a ta z kolei wybiera Zarząd Gminy, lokalną władzę wykonawczą. Zarówno Rada jak i Zarząd działają w ogólności dla dobra wspólnoty, ale są przede wszystkim zobowiązane do wykonywania ustawowo określonych zadań własnych i zleconych. Takie usytuowanie Rady i Zarządu powoduje, że znajdują się one z jednej strony pod ustawiczną presją mieszkańców, z drugiej zaś mają do wykonania szereg ustawowych obowiązków. Należy podkreślić, że środki, jakimi dysponują samorządy są zazwyczaj niedostateczne do wywiązania się z tych obowiązków i sprostaniu żądaniom mieszkańców. Żądania te dotyczą zwykle rozwiązania podstawowych problemów bytowych wynikających ze złego stanu infrastruktury technicznej i społecznej (brak wodociągów i



kanalizacji, utwardzonych i oświetlonych dróg, złołków i przedszkoli, mieszkań komunalnych itp.).

Do tego należy dodać szereg wymogów i ograniczeń formalnych, ustawowych i proceduralnych w zakresie dysponowania budżetem gminnym (ustawy o samorządzie, finansach publicznych, procedury przetargów publicznych itd.), przeciętnie niskie kompetencje i kadencyjność decydentów. W tej perspektywie przedsięwzięcia informatyczne w sferze samorządowej są trudniejsze i obciążone większym ryzykiem niepowodzenia od projektów realizowanych w sferze biznesu. Głównym celem informatyzacji w samorządach jest zwiększenie efektywności wykonywania zadań i rozwiązywania problemów. Podstawowym aparatem realizacji zadań władzy wykonawczej w gminie – Zarządu jest Urząd Gminy. Podniesienie sprawności jego działania jest w praktyce celem większości projektów informatycznych w samorządach. Podobnie było w rozpatrywanym przykładzie, choć od początku celem w Łomiankach była informatyzacja całej gminy, a nie tylko Urzędu.

Gmina Łomianki jest gminą szczególną ze względu na położenie, nagromadzenie konfliktów i ogrom potrzeb infrastrukturalnych. Bliskość metropolii warszawskiej, położenie w obszarze chronionym (z jednej strony Puszcza Kampinoska, z drugiej obszar chroniony doliny Wisły), przecięcie gminy drogą krajową o ogromnym natężeniu ruchu, konieczność ochrony środowiska, duża presja migracyjna i urbanizacyjna, konflikty między starymi i nowymi mieszkańcami, brak przestrzeni społecznej dla młodzieży, brak wodociągów i kanalizacji, brak szkół, to tylko niektóre z problemów.

## 1. Stan wyjściowy w 1998 roku

Urząd Miasta i Gminy w Łomiankach od powstania miasta w 1990 roku podlegał stopniowej i „naturalnej” komputeryzacji. Proces ten przebiegał chaotycznie, inwestycje, a raczej doraźne zakupy odbywały się w miarę potrzeb i możliwości finansowych, bez planu, wytyczonego kierunku i celu nadrzędnego.

W efekcie było za mało sprzętu komputerowego i był on w przeważającej części przestarzały. Sieć zbudowana w nienowoczesnej technologii ethernetu koncentrycznego funkcjonowała w bardzo ograniczonym zakresie w 2 niezależnych fragmentach. Systemowo występowało mieszane środowisko SCO Unix, Novell Netware i MS Windows 95. Brakowało aplikacji użytkowych, a z posiadanych, nie wszystkie zostały wdrożone. Prawidłowo funkcjonowały jedynie moduł księgowości podatkowej i finansów budżetowych oraz niezależny terminal dostępu do systemu ewidencji ludności.

W Urzędzie zatrudnionych było 2 informatyków, każdy na pół etatu. W praktyce nie było to jednak rozwiązanie pozwalające efektywnie wprowadzać jakieś istotne zmiany w funkcjonowaniu systemu przetwarzania informacji. Jednocześnie stan finansów gminnych był, delikatnie mówiąc – zły. Gmina osiągnęła górny, ustawowo określony, 60% pułap zadłużenia. Złe umowy kredytowe i inwestycyjne doprowadziły do niemal 20% luki budżetowej zaś

poziom inwestycji przekraczał 40% w budżecie na 1999 rok, odziedziczonym po poprzedniej kadencji. Niezbędne inwestycje gminne ugrzęzły w paraliżu finansowym, nieprawidłowo kierowane i nieodpowiednio nadzorowane. Nie była to dobra sytuacja wyjściowa do podejmowania przedsięwzięcia informatycznego.

Mimo to, a raczej właśnie, dlatego, że stan gminy był niezadowolający na tak rozległym obszarze, rozpoczęto projekt informatyzacji gminy. Projekt zakładał z tego względu, od początku konieczność informatyzacji nie tylko Urzędu, ale szersze podejście – informatyzację gminy. W efekcie zaangażowania Zarządu i zaangażowania grupy informatyków z firm TORN i ASTECH została zaproponowana wstępna koncepcja informatyzacji Łomianek, zaś ostateczna wersja w postaci Kompleksowej Informatyzacji Gminy (KIG) – GMIS/OTIS została opracowana przez firmę TORN.

Należy podkreślić, że były również pozytywne przesłanki projektu, stwarzające korzystne dla przedsięwzięcia perspektywy, głównie dobrze rozwinięta sieć telekomunikacyjna na terenie gminy, potencjał lokalnych informatyków oraz możliwość przeprowadzenia działań wstępnych i przygotowawczych bez nakładów z budżetu gminnego, co określono jako *Opcję Zero*.

## 2. Plan ogólny

Na wstępie Zarząd Gminy podjął decyzję o potrzebie unowocześnienia systemu przetwarzania informacji w Urzędzie. Wobec trudności finansowych przyjęto *Opcję Zero* jako rozwiązanie polegające na wykonaniu koncepcji informatyzacji i rozpoczęciu jej wdrożenia bez nakładów z budżetu gminy, własnymi siłami.

W ramach *Opcji Zero* ostatecznie wykonano:

- Koncepcję wstępną informatyzacji gminy – GMIS/OTIS
- Plan działania na 1999 rok, zakładający:
  - Stworzenie podstawy technicznej systemu (strukturalne okablowanie sieci komputerowej, zasilanie awaryjne, częściową wymianę i rozbudowę sprzętu komputerowego),
  - Połączenie sieci LAN w 3 budynkach Urzędu w jedną sieć kampusową,
  - Opracowanie wytycznych do budowy zintegrowanego systemu zarządzania gminą (GMIS) dla Urzędu,
  - Opracowanie wytycznych do budowy systemu informacyjnego gminy (OTIS),
  - Instalacja i uruchomienie systemu kontroli spraw i obiegu dokumentów,
  - Opracowanie projektu i uruchomienie gminnej strony internetowej (element OTIS),
- Materiały przetargowe do uruchomienia przetargów publicznych na:
  - Okablowanie Urzędu,

- Wyposażenie stanowisk w nowoczesny sprzęt komputerowy wraz z zasilaniem awaryjnym,
- System poczty elektronicznej oraz kontroli spraw i obiegu dokumentów.
- Zarys planu działania na rok 2000, zakładający:
  - Dalsze unowocześnienie sprzętu komputerowego – wymianę i uzupełnienie wyposażenia stanowisk w stacje robocze,
  - Wybór i rozpoczęcie wdrożenia zintegrowanego systemu informatycznego dla UMiG,
  - Uruchomienie eksploatacji systemu kontroli spraw i obiegu dokumentów.

Wobec ograniczeń wynikających ze:

- skromnego budżetu, trudności jego realizacji - nierytmiczność i brak rzeczywistych środków,
- braku możliwości stałego nadzoru nad kierowaniem projektem,
- niskiego poziomu opanowania technologii informacyjnych przez kadre UMiG,
- dynamicznie rozwijającego się potencjału IT (nowe rozwiązania informatyczne i rozwój telekomunikacji na terenie gminy),

przyjęto szereg założeń warunkujących realizację projektu, z których najważniejsze przedstawiam poniżej.

### **Założenia warunkujące**

#### 1. Dopasowanie zdolności użytkowników do wdrażanych technologii

- Stopniowe wdrażanie nowych systemów dla uniknięcia dezorganizacji w UMiG i umożliwienie nauczania się tych systemów przez pracowników (lub zatrudnienie pracowników o wyższych kwalifikacjach),
- Zapewnienie szkoleń w trakcie wdrożeń,
- Umożliwienie samodoskonalenia się pracowników - zapewnienie opieki autorskiej producentów oprogramowania.

#### 2. Dopasowanie technicznej platformy sprzętowej i telekomunikacyjnej do rozwoju aplikacji, potrzeb i możliwości

- Wykorzystanie nowoczesnych technologii telekomunikacyjnych w miarę ich dostępności na terenie gminy,
- Wymiana urządzeń technicznych w miarę rozwoju technologii dla uniknięcia sytuacji, gdy cały sprzęt komputerowy będzie przestarzały,
- Unikanie dużych zmian ilościowych i jakościowych dla eliminacji:
  - czynników dezorganizujących pracę UMiG,
  - konieczności dużych wydatków jednorazowych,
  - barier w przyswajaniu przez użytkowników nowych rozwiązań.

#### 3. Dopasowanie wzajemne wdrażanych technologii i zdolności finansowych

- Wybór metody finansowania nakładów na urządzenia komputerowe za pomocą leasingu, ze względu na:



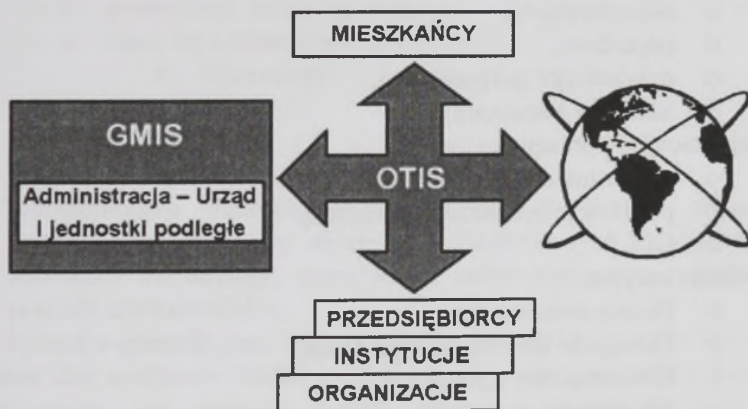
- ograniczenia budżetowe - brak środków całość zadania inwestycyjnego w zakresie informatyzacji (brak odpowiedniego budżetu dla potrzeb GMIS i OTIS),
- brak zdolności kredytowej gminy – niemożność kredytowania nakładów na sprzęt,
- stopniowe ponoszenie nakładów na zadanie, umożliwiające dopasowanie do stopniowo uzyskiwanych dochodów gminy,
- Poszukiwanie innych rozwiązań finansowania IT oferowanych przez dostawców.

### 3. Koncepcja KIG - Projekt GMIS/OTIS

Koncepcja informatyzacji gminy GMIS/OTIS zakłada stworzenie dwóch, komplementarnych podsystemów - płaszczyzn przetwarzania i wymiany informacji w gminie:

- Zintegrowanego Systemu Wspierania Zarządzania Gminą – GMIS
- Otwartego Systemu Dostępu i Wymiany Informacji – OTIS

Obydwa podsystemy realizowane są w ramach Kompleksowej Informatyzacji Gminy (KIG) – zadania inwestycyjnego wpisanego do budżetu miasta i gminy Łomianki. Każdy z nich ma odmienne zadania i cele, zaś ich efekty uzupełniając się, mają dać nową jakość. Ideę koncepcji przedstawia poniższy schemat.



Rys.1. Idea Kompleksowej Informatyzacji Gminy (KIG) w modelu GMIS/OTIS

#### Zintegrowany System Wspierania Zarządzania Gminą – GMIS

##### Cele systemu:

1. Poprawa efektywności:
  - 1.1. Zarządzania urzędem (kierowanie ludźmi, sprawami i zadaniami),
  - 1.2. Załatwiania spraw mieszkańców,
  - 1.3. Realizacji zadań własnych gminy,
2. Usprawnienie zarządzania finansowego gminą:

- 2.1. Wymiana oprogramowania finansowo-księgowego (dostosowanie do nowych przepisów),
- 2.2. Instalacja systemu do zarządzania budżetem i jego realizacją (z uwzględnieniem wymogów sprawozdawczych dla nadzoru RIO, klasyfikacji budżetowej i zadaniowej oraz rozliczania WPI),
- 2.3. Monitorowania finansów gminy z uwzględnieniem wymagań ustawowych i przystosowawczych do integracji z UE,
- 2.4. Rozliczania i egzekwowania podatków lokalnych;
3. Utworzenie ewidencyjnych baz danych do monitorowania stanu i planowania odtworzenia infrastruktury technicznej gminy oraz jej rozwoju;
4. Dostosowanie do wymagań elektronicznej formy przekazywania dokumentów do jednostek organów Państwa (RIO), ZUS, ich publikacji oraz wykonywania transakcji finansowych z bankami;
5. Poprawa wydajności pracy urzędników (muszą wykonywać więcej zadań, a nie ma możliwości odpowiedniego zwiększenia zatrudnienia);
6. Oszczędności w kosztach funkcjonowania administracji lokalnej.

#### Budowa wewnętrzna GMIS – podsystemy:

- Obiegu dokumentów i informacji oraz kontroli spraw,
- Wspierania zarządzania i planowania,
- Finansów i podatków,
- Ewidencyjne:
  - infrastruktury technicznej,
  - ludności,
  - nieruchomości,
  - pojazdów,
  - działalności gospodarczej,
  - zasobów komunalnych,
- Publikacyjne i prezentacyjne:
  - archiwum elektroniczne,
  - publikacja informacji publicznej,
  - GIS,
- Komunikacyjne:
  - Poczta elektroniczna,
  - Dostęp do informacji prywatnej,
  - Elektroniczne formularze,
  - Monitorowanie przestrzeni,
  - Ankiety, sondaże, referenda i głosowanie.

### **Otwarty System Dostępu i Wymiany Informacji – OTIS**

#### Cele systemu:

1. Realizacja regulacji ustawowych o jawności i dostępie do informacji,
2. Poprawa poziomu i jakości usług administracji lokalnej, świadczonych mieszkańcom,
3. Promocja gminy w Polsce, Europie i Świecie,
4. Wspieranie rozwoju gospodarczego gminy,

5. Propagowanie pożądaných kierunków rozwoju,
6. Umożliwienie szerokiej i szybkiej konsultacji społecznej w istotnych dla gminy sprawach,
7. Oszczędność kosztów realizacji zadań 1-6,
8. Podnoszenie poziomu świadomości społecznej w korzystaniu z demokracji lokalnej (jawność spraw, ustaw, propagowanie demokratycznych rozwiązań),
9. Integracja społeczności lokalnej przez tworzenie przestrzeni informacyjnej dla wymiany myśli, poglądów i organizowania się społeczności i środowisk,
10. Aktywizacja jednostek i środowisk lokalnych – wyposażenia ich w narzędzia wymiany, propagowania i ochrony dorobku kulturalnego i intelektualnego,
11. Budowa podstaw społeczeństwa informacyjnego – umożliwienie powszechnego dostępu do publicznej przestrzeni informacyjnej (np. przez publiczne punkty dostępowe),
12. Przygotowanie społeczności lokalnej do aktywnej obecności w UE – rozwoju w zintegrowanej Europie.

### Struktura systemu OTIS

Koncepcja OTIS zakłada wykorzystanie Internetu jako podstawy całej konstrukcji, której centralnym filarem jest internetowy portal gminny, a raczej technologia portali wirtualnych, zainstalowana na gminnym serwerze (docelowo – serwerach).

#### □ Portal internetowy gminy:

- obsługa spraw mieszkańców przez sieć,
- płaszczyzna prezentacji spraw publicznych,
- płaszczyzna aktywnego udziału w życiu gminy,
- prezentacja gminy na świecie,
- okno do komunikacji ze światem:
  - informacje
  - biznes
  - dialog
  - edukacja i kultura,

#### □ Udostępnienie technologii – wirtualnych portali organizacjom i mieszkańcom w miarę możliwości (technicznych – obecnie do 128 i zdolności do obsługi, opanowania technologii przez użytkowników) i potrzeb użytkowników,

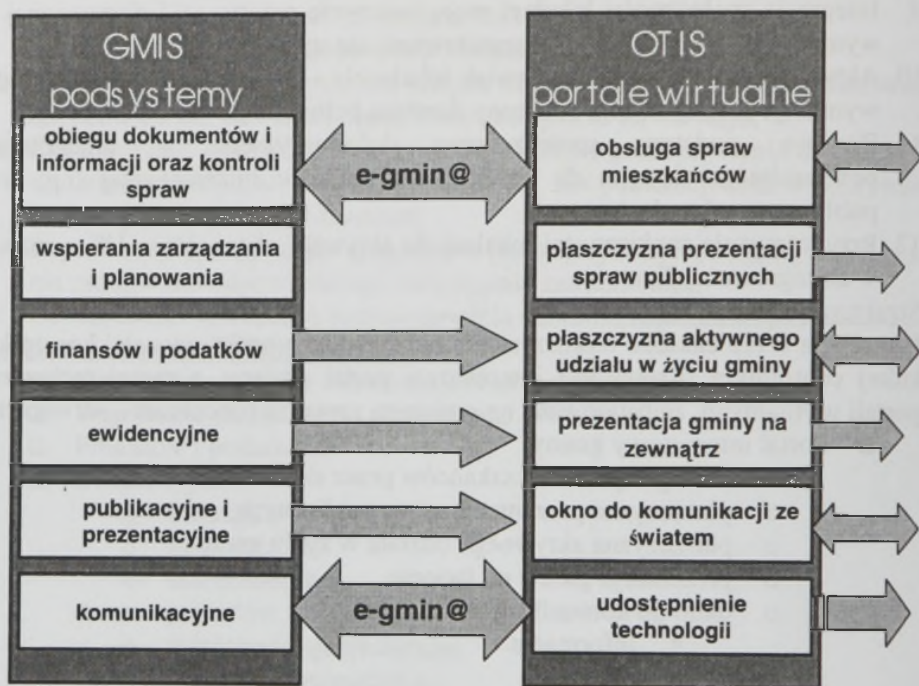
#### □ Moduł **e-gmin@**, pozwalający przez portal gminny „wejść” do Urzędu.

Połączenie obu systemów GMIS i OTIS tworzy wirtualną gminę, której trzonem konstrukcyjnym jest gminny portal. Istotnym elementem funkcjonowania połączenia systemów jest **e-gmin@**. Moduł ten pozwala na „inteligentny” dostęp do zasobów i baz danych, do których dany użytkownik chce się dostać (autoryzacja petenta). Pozwala on także na uruchamianie spraw i sprawdzanie ich statusu w Urzędzie. Umożliwia również dwustronną komunikację z konkretnymi pracownikami Urzędu oraz kontakt Urzędu z mieszkańcem w przypadku zapytań, ankiet, sondaży i głosowań. Oczywiście pojawia się tu szereg problemów prawnych i technicznych, które wymagają rozwiązania (autoryzacja i zabezpieczenie dostępu, ochrona danych osobowych i zapewnienie prywatności, podpis elektroniczny i wiele innych).



### Wstępne założenia wdrożeniowe:

- Zagwarantowanie środków budżetowych na lata 1999-2002,
- Wykonanie audytu funkcjonalnego instytucji administracji publicznej (UMiG) i dokonanie zmian organizacyjnych w tych instytucjach,
- Zatrudnienie menedżera (szefa) projektu do nadzoru wdrożenia zintegrowanego systemu dla Urzędu.



Rys.2. Ogólny schemat funkcjonalny modelu GMS/OTIS

## 4. Przebieg procesu

Proces informatyzacji przebiegał oczywiście inaczej niż planowano. Życie zawsze przynosi niespodzianki. Ponadto popełniono szereg błędów, a problemy budżetowe nie pozwalały mieć pewności, co do rzeczywiście posiadanych środków w kasie Urzędu na realizację przedsięwzięcia. Taki stan zmuszał do ciągłej weryfikacji założonych planów. W budżecie 1999 roku udało się zaplanować środki na wykonanie niektórych elementów planu, ale w niedostatecznej wysokości. W tym miejscu należy zwrócić uwagę, że nawet najlepiej przygotowany budżet jest tylko planem finansowym. Budżet gminy jest w dodatku zależny od wielu czynników, na które Zarząd Gminy nie ma wpływu (dochody realizowane przez budżet Państwa, terminy przekazywania dotacji i subwencji, wpływy z podatków lokalnych i ich terminowość). Poniżej przedstawiony został rzeczywisty przebieg procesu realizacji KIG.

## **Rok 1999**

### Stworzenie podstawy technicznej systemu

- Wykonano sieć komputerowa w nowoczesnej technologii - MODTAP (standard CAT5) na 90 stanowisk komputerowych we wszystkich trzech budynkach UMiG,
- Wyposażono kluczowe stanowiska pracy w dobrej jakości urządzenia (komputery PC Hewlett Packard i IBM) aczkolwiek w niewystarczającej liczbie (dla docelowego zintegrowanego systemu),
- Połączono sieci LAN w 2 budynkach Urzędu w jednolitą sieć,
- Rozpoczęto prace nad połączeniem sieci UMiG z pozostałym budynkiem.

### Stworzenie podstawy systemu informacyjnego gminy - OTIS

- Opracowano projekt, zarejestrowano domenę i uruchomiono gminną stronę internetową [www.lomianki.pl](http://www.lomianki.pl),
- Weryfikowano i modyfikowano projekt i serwis internetowy wraz z wdrażaniem i testowaniem procedur jego uaktualniania – nastąpiło kilka istotnych zmian w ciągu roku,
- Skutkiem powyższego rozpoczęto działania nad fizycznym umieszczeniem witryny www na serwerze w UMiG - budowa stanowiska SDI w porozumieniu z TP S.A.

### Budowa podstawowych baz danych - systemów ewidencyjnych

- Opracowano wytyczne dla zintegrowanego systemu zarządzania gminą (GMIS),
- Wobec potrzeb (obowiązek ustawowy) w zakresie budowy i remontów dróg gminnych - rozpoczęto budowę ewidencji sieci drogowej – program i wykonanie numerycznej mapy sieci drogowej oraz rozpoczęcie zakładania metryk dróg gminnych,
- Instalacja oprogramowania do weryfikacji baz danych płatności podatku drogowego.

### Implementacja aplikacji ZARZĄDZANIA INFORMACJĄ I OBIEGIEM DOKUMENTÓW

- Podstawowy system kontroli realizacji zadań – stworzenie założeń ogólnych aplikacji (platforma systemowa, wybór aplikacji i dostawcy),
- Stworzenie platformy systemowej wymiany informacji i dokumentów - wybrano Lotus Notes i rozpoczęto wdrożenie wewnętrznej poczty elektronicznej, systemu rejestracji i obiegu dokumentów.

## **Rok 2000**

### Budowa podstawy technicznej systemu

- Wykonano system zasilania awaryjnego dla sieci komputerowej w 2 budynkach w oparciu o awaryjne zasilacze bezprzerwowe i agregaty prądowórcze,
- Wyposażono dalsze stanowiska pracy w urządzenia (komputery PC Hewlett Packard) chociaż nadal w niewystarczającej liczbie (dla docelowego zintegrowanego systemu).

### Budowa systemu OTIS

- Dalsze modyfikacje serwisu, zmiana ISP, i opracowanie koncepcji gminnego portalu jako bazowej platformy GMIS/OTIS,
- Działania wspierających koncepcję OTIS - uruchomienie programu XXI,
- Przygotowano i uruchomiono przetarg na kierowanie projektem GMIS/OTIS – przetarg unieważniony (tylko 1 oferta).

#### Budowa systemu GMIS

- Uruchomiono przetarg na wdrożenie zintegrowanego systemu zarządzania gminą (GMIS) w UMiG - przetarg unieważniony z powodu braku odpowiednich aplikacji na rynku i braku środków - przesunięcie zadania na 2001 rok,
- Wdrożono system ewidencji sieci drogowej – wykonanie numerycznej mapy sieci drogowej założenie metryk dróg gminnych,
- Dalsze uzupełnienie oprogramowania weryfikacji baz danych płatności podatku drogowego,
- Kontynuowano wdrażania poczty elektronicznej w UMiG (zwiększenie liczby użytkowników, dalsze szkolenia),
- Prace modyfikacyjne systemu rejestracji i obiegu dokumentów (zw. ze zmianami w obowiązującej instrukcji kancelaryjnej dla urzędów administracji publicznej).

Zadania w zakresie GMIS, a zatem realizacja projektu KIG jako całości była praktycznie wstrzymana w 2000r ze względu na brak środków. Pojawiły się one dopiero w samym końcu roku, co nie pozwoliło na ich uruchomienie w ramach budżetu roku 2000.

#### **Rok 2001**

Ograniczenia budżetowe 2001 roku spowodowały kolejne zmodyfikowanie planu informatyzacji i ograniczenie go do części OTIS w postaci zaimplementowania technologii portalowej. Przeznaczone w budżecie środki nie pozwoliły na uruchomienie przetargu na zintegrowany system zarządzania gminą GMIS i zadanie to zostało po raz kolejny przesunięte na następny rok.

#### Rozbudowa podstawy technicznej systemu

- Rozwój platformy telekomunikacyjnej i systemowej UMiG – instalacja łącza stałego 128 Kbit,
- Instalacja serwera internetowego na systemie Linux w UMiG,
- Przejęcie prowadzenia serwisu internetowego od ISP.

#### Budowa systemu OTIS

- Wybór technologii portalowej – przygotowanie przetargu i wyłonienie dostawcy
- Instalacja technologii portalowej *TORN Palio* – produkcji firmy TORN (9 portali wirtualnych)
- Prace projektowe (szczegóły zawartości i formy ekranowe) i uruchomieniowe
- Rozwiązania organizacyjne w UMiG (administracja serwisami, zarządzanie usługami, support techniczny)
- Szkolenie użytkowników – administratorów, zarządzających i autorów



- Szkolenia dostawców informacji i publikujących
- Uruchomienie elektronicznej wersji dwutygodnika Urzędu – „Gazety Łomiankowskiej” na portalu: <http://gazeta.lomianki.pl>

#### Budowa systemu GMIS

- Technicznie wdrożono SYSTEM ZARZĄDZANIA INFORMACJĄ I OBIĘGIEM DOKUMENTÓW – nastąpiło dalsze opóźnienie w rozpoczęciu eksploatacji ze względu na brak odpowiednich decyzji kierownictwa Urzędu
- Uruchomienie komunikacji dwukierunkowej z portalem gminnym – modułu e-gmin@ w trybie testowym

### **Rok 2002**

#### Budowa systemu GMIS

- Późne uchwalenie budżetu na 2002 rok spowodowało uruchomienie przetargów na uzupełnienie sprzętu i zintegrowany system dla Urzędu – trzon GMIS, dopiero w maju.
- Na początku roku zatrudniono wreszcie szefa projektu i przeorganizowano w Urzędzie funkcjonowanie informatyki i służby informacyjnej.
- Wdrożono testowo SYSTEM ZARZĄDZANIA INFORMACJĄ I OBIĘGIEM DOKUMENTÓW dla 3 referatów Urzędu

#### Budowa systemu OTIS

- Od końca 2001r. intensywnie postępowaly prace nad implementacją technologii portalowej. Wykonano szereg projektów i wybrano ostatecznie wersję graficzną do wdrożenia głównego portalu gminnego
- Trwa wypełnianie danymi inicjalnymi portalu <http://server.lomianki.pl>
- Opracowywane i uruchomiane są procedury aktualizacji portalu gminnego.
- Trwają prace uruchomieniowe pozostałych 7 portali wirtualnych (nasze, nowe, biznes, uksw, gimnazjum1, gimnazjum2, liceum).

### **5. Popelnione błędy**

Z perspektywy z górą trzech lat od rozpoczęcia projektu można krytycznie oceniać jego przebieg, sformułować szereg uwag i określić popelnione w trakcie wdrożenia błędy. Oto najważniejsze z nich:

- Nie spełnienie wstępnych założeń wdrożeniowych
- Zbyt słaby nacisk na publiczne prezentowanie konieczności inwestowania w IT i korzyści płynących z efektów tych inwestycji.
- Brak w gminie Wieloletniego Planu Inwestycyjnego.
- Brak szefa projektu wdrożeniowego, odpowiedzialnego za poszczególne zadania.
- Słabość organizacji wewnętrznej funkcjonowania Urzędu Gminy

*Nie spełnienie wstępnych założeń wdrożeniowych* było w zasadzie przyczyną większości następných problemów i źródłem kolejnych błędów. Nie było jednak

praktycznych możliwości ich spełnienia, a oczekiwanie na poprawę sytuacji przesunęłoby w czasie rozpoczęcie projektu, o co najmniej 2 lata.

*Zbyt słaby nacisk na publiczne prezentowanie konieczności inwestowania w IT i korzyści płynących z efektów tych inwestycji dotyczy przekonania większości radnych do zatwierdzenia budżetu na informatyzację, pozwalającego sprawnie i szybko ją przeprowadzić. Co roku budżet IT „przegrywał” z ważniejszymi, w odczuciu radnych zadaniami – np. kanalizacją i drogami.*

*Brak Wieloletniego Planu Inwestycyjnego powodował, że co roku podchodzono do finansowania informatyzacji od początku, mimo formalnej kontynuacji zadania. Brak WPI nie pozwalał na uchwalenie całego budżetu informatyzacji i np. przeprowadzania przetargów w terminach odpowiednich do wdrożenia, lecz dopiero po uchwaleniu budżetu w poszczególnych latach.*

*Brak szefa projektu wdrożeniowego – człowieka, który byłby odpowiedzialny personalnie i merytorycznie za postępy prac wdrożeniowych, powodował opóźnienia w procesie decyzyjnym, niejasność kompetencyjną i nieporozumienia w klasyfikowaniu środków budżetowych. Zakończona niepowodzeniem próba zatrudnienia odpowiedniego człowieka w 2000 roku była obok problemów finansowych główną przyczyną opóźnień i niskiej efektywności wdrożenia.*

*Słabość organizacji Urzędu, polegająca m.in. na braku jasno sformułowanych zadań i kompetencji dla pracowników zaangażowanych w projekt w połączeniu z brakiem jego szefa zmniejszała zdolność egzekwowania decyzji, dalsze opóźnienia i w ostateczności znacznie osłabiała dynamikę realizacji. Szczególnie dało się to odczuć w tempie wdrożenia SYSTEMU ZARZĄDZANIA INFORMACJĄ I OBIĘGIEM DOKUMENTÓW. System ten, z naturalnych powodów nie spotkał się z entuzjazmem większości urzędników, którzy mnożyli mniej lub bardziej uzasadnione powody, uniemożliwiające uruchomienie systemu.*

## **6. Postulaty metodologiczne**

Doświadczenia z wdrażanej koncepcji wirtualnej gminy – GMIS/OTIS pozwalają sformułować szereg postulatów ogólnych, dotyczących informatyzacji samorządów gminnych.

W dobie lansowanej w Unii Europejskiej koncepcji e-Europe – społeczeństwa informacyjnego, wydaje się słuszne podejście kompleksowe do informatyzowania całej gminy, nie tylko jednostek administracji. Inicjatywa e-Europe jest wdrażana w krajach Unii Europejskiej ze względu na to, że technologia stała się na tyle tania i dostępna, aby uczynić tę drogę najbardziej ekonomicznie uzasadnioną dla rozwoju demokracji europejskiej.

Kompleksowa informatyzacji społeczności lokalnych pozwala uzyskać nową jakość z wdrażanych technologii, opartych na Internecie – nie tylko określone oszczędności finansowe i wzrost efektywności pracy urzędników.

Co więcej – ustawa o dostępie do informacji publicznej nakłada na urzędy obowiązek do uruchomienia internetowych publikacji do lipca 2003 roku. Czasu nie jest dużo, a doświadczenia z Łomianek pokazują, że prace wdrożeniowe w administracji publicznej przeciągają się bardziej niż w innych sferach.

Koncepcja wirtualnej gminy zakłada nie tylko budowę zintegrowanego systemu przetwarzania informacji w urzędzie gminy i udostępniania informacji publicznej mieszkańcom, ale budowę sfery informacyjnej w gminie w oparciu o Internet.

Oto ogólny postulowany zarys metody budowy wirtualnej gminy:

1. Analiza:

- 1.1. zasobów rzeczowych – jaki sprzęt, systemy, aplikacje, możliwości telekomunikacyjne są w dyspozycji i jak można je wykorzystać,
- 1.2. potencjału ludzkiego – jakie zdolności przyswojenia zmian w korzystaniu z IT i posługiwania się technologią posiadają pracownicy administracji i potencjalni użytkownicy,
- 1.3. zasobów i możliwości finansowych – ile należy zainwestować w IT i jak to sfinansować.

2. Analiza funkcji:

- 2.1. wyodrębnienie obszarów i procesów do informatyzacji poza sferą urzędowo – administracyjną,
- 2.2. określenie zakresu informatyzacji w jednostkach administracji i rozłożenie procesu w czasie (szczególnie ze względu na pp. 1.2 i 1.3),
- 2.3. określenie niezbędnych zmian w systemie funkcjonowania administracji gminnej – dostosowanie organizacji urzędu do planowanej technologii pracy,

3. Sformułowanie celu – określenie modelu docelowego

- 3.1. Obieg, gromadzenie i udostępnianie informacji – OTIS,
- 3.2. Przetwarzanie, analizowanie i syntetyzowanie informacji – GMIS,
- 3.3. Elementy styczne – część wspólna, czyli koła zębate całej maszinerii,
- 3.4. Jak to wszystko ma działać za 2-4 lata,
- 3.5. Jakie to przyniesie korzyści.

4. Wybór technologii

- 4.1. Jakość – podstawowe kryterium wyboru,
- 4.2. Cena – podrzędne kryterium wyboru,

5. Plan wdrożenia

- 5.1. Promocja projektu – zorganizowanie poparcia „politycznego” na terenie samorządu (Rada Gminy),
- 5.2. Metoda finansowania – wybór metod optymalnych z punktu widzenia budżetu gminnego, nie koniecznie najtańszych (np. leasing),
- 5.3. Harmonogram – dopasowanie tempa wdrożenia do zdolności adaptacyjnych zaangażowanych pracowników, zdolności do finansowania projektu (cykliczność i specyfika realizacji budżetu), ustawowych i



formalnych ograniczeń dynamiki realizacji (wymogi przetargów publicznych, konieczność podejmowania niektórych decyzji przez Radę Gminy – czas między sesjami itp.),

- 5.4. Ludzie – wybór kadry, która ma odpowiednie predyspozycje do realizacji poszczególnych zadań wdrożeniowych (czasem odpowiednie zmotywowanie pracowników zdolnych do przyswajania nowości daje znacznie lepsze rezultaty niż poleganie na pracownikach pełniących określone funkcje, czy zajmujących stanowiska kierownicze).
6. Podział konkretnych zadań realizacyjnych między ludzi jednostki – rozpoczęcie wdrożenia.
7. Kontrola wdrożenia:
  - 7.1. sprawozdawczość i rozliczanie zadań (koszty, budżet),
  - 7.2. Weryfikacja założeń i zadań,
  - 7.3. Weryfikacja harmonogramu.

## 7. Wnioski

W skali kraju istnieje zasadniczy problem informatyzacji samorządów gminnych. Mimo opisanych trudności wdrażania informatyki w gminie Łomianki, należy podkreślić duży potencjał intelektualny mieszkańców Łomianek, który można było zaangażować w ten proces, co w znacznym stopniu się powiodło. Gmina posiadała na wyjściu bardzo dobrą, jak na polskie warunki – sieć telekomunikacyjną, na europejskim poziomie. Ponadto trudności finansowe gminy nie zmieniają zasadniczo jej potencjału ekonomiczno – finansowego. Gmina Łomianki nie jest gminą ubogą.

Nie oznacza to jednak, że gminy ubogie o słabym potencjale intelektualnym i słabo rozwiniętej sieci telekomunikacyjnej znajdują się na straconych pozycjach. Inwestycja w IT przeprowadzona wg przedstawionej koncepcji wirtualnej gminy GMIS/OTIS, może przynieść takim gminom korzyści nieosiągalne dotychczas, poprzez tradycyjne podejście informatyzację urzędu gminy. Warunkiem jest odpowiednie przygotowanie i wdrożenie projektu oraz unikanie opisanych wyżej błędów.

Autor: Jarosław Ładyga, niezależny konsultant,  
współpracownik firmy TORN Sp.z o.o., ul. Sosnkowskiego 17, 02-495 Warszawa,  
e-mail: [jladyga@torn.com.pl](mailto:jladyga@torn.com.pl)

# ZARZĄDZANIE JAKOŚCIĄ W IT W ŚWIETLE NORM ISO

Andrzej Niemiec

**Streszczenie** : W pracy zostaną omówione najważniejsze standardy związane z projektowaniem, rozwojem i wdrażaniem systemów informatycznych. Zagadnienia te są istotne dla firm posiadających systemy zarządzania oparte o normy ISO, firm produkujących i wdrażających oprogramowanie oraz osób zainteresowanych w prawidłowym funkcjonowaniu systemów informatycznych w kontekście zaleceń norm ISO/IEC. Praca przedstawia podstawowe normy ISO związane z produkcją i oprogramowaniem. Opisano wymagania ISO 9001, ISO 9000-3, ISO 12207, ISO 15504 i norm związanych.

## Dlaczego normy?

Istnieją dwa podstawowe powody ustanawiania i stosowania norm:

- ochrona odbiorcy/ użytkownika = jakość
- pomoc producentowi = działanie

Potrzeba stosowania norm wynika z:

- ułatwienia pracy producenta - normy zawierają przemyślenia i doświadczenia wielu organizacji
- zabezpieczenia interesów odbiorcy - wyrażają wymagania i potrzeby i chronią przed samowolą dostawcy
- ułatwienia stosowalności produktów

Działalności normalizacyjnej nie oparła się informatyka. Informatyka polska ma długie tradycje w normalizacji. Już w 1923 - Polska została członkiem Międzynarodowej Komisji Elektrotechnicznej IEC; a w 1947 - Polski Komitet Normalizacyjny został członkiem-założycielem Międzynarodowej Organizacji Normalizacyjnej ISO. Jedną z pierwszych norm związanych z informatyką opublikowaną w Polsce była PN-71 T- 01016 „Przetwarzanie danych i komputery- Podstawowe nazwy i określenia” (16 stron).

Jednym z warunków integracji Polski z UE jest udział PKN w Europejskim Komitecie Normalizacyjnym (CEN) oraz Europejskim Komitecie Normalizacyjnym Elektrotechniki (CENELEC).

Do końca 2002 roku Polska powinna przyjąć co najmniej 80% norm europejskich. Jest ich około 9,5 tys, ale trwają prace nad nowymi. PKN wdrożył 4,7 tys. norm europejskich (przetłumaczone). Uznano 3,6 tys tzn. przyjęto je jako normy polskie ale w języku angielskim [Jerzy Marcinek, prezes PKN]. Normy są dokumentami technicznymi, które powszechnie i dobrowolnie mogą stosować producenci, eksporterzy, importerzy, administracja państwowa, samorząd

terytorialny. Normy są opracowywane oraz przyjmowane na zasadzie konsensu - porozumienia między przedstawicielami danej branży.

Normy nie są

- uniwersalną receptą na sukces - niestety nie istnieje :-(
- utrapieniem i utrudnieniem - gdyż systematyzują i ułatwiają pracę,
- jednym słusznym sposobem działania - bowiem zostawiają bardzo dużą swobodę interpretacji zwracając uwagę na najistotniejsze zagadnienia, kładąc nacisk na obszary, które twórcy norm uznali za szczególnie istotne.

Szczególną pozycją jest przewodnik ISO / IEC 2: 1996 „Standardization and related activities -- General vocabulary” służący do interpretacji słownictwa norm. Stosowany w normach język zawiera wiele pułapek językowych, np. słowa powinien, może, należy, utrzymywanie mają w normach ścisłe znaczenie, nie pokrywające się z językiem potocznym.

Normy dotyczące oprogramowania skupiają się na trzech obszarach

- jakość oprogramowania - w sensie zapewnienia jakości usługi, zagadnienia organizacyjne
- procesy związane z cyklem życia wyrobu programowego - opanowanie problemów związanych ze zmiennością wymagań zmianami środowiska i postępem technicznym
- stosowane techniki, definicje języków i środowisk - zagadnienia „techniczne”

Stosowanie norm sprzyja:

- Zapewnieniu odpowiedniej funkcjonalności
- Zapewnieniu zgodności
- Zapewnieniu zamienności
- Regulowaniu różnorodności
- Uzyskaniu wysokiego stopnia bezpieczeństwa wyrobów, procesów i usług,
- Ochronie środowiska
- Ochronie wyrobu

Normy związane z IT sprzyjają:

- wydajniejszemu tworzeniu oprogramowania
- stosowaniu "dobrych praktyk"
- systematyczności działań i podejmowanie właściwych
- ułatwieniu zrozumienia zaangażowanych stron

Stosowanie norm daje też jednoznaczność pojęć, terminów i definicji – zwłaszcza w przypadku słów niejednoznacznych (np. program, produkt, system, aplikacja) co znakomicie ułatwia kontakty z dostawcami i klientami.



## Pojęcia jakości

Istnieje wiele definicji jakości - począwszy od Platona: Jakość jest to pewien stopień doskonałości, poprzez Arystotelesa Jakość jest jedną z dziesięciu kategorii, które umożliwiają podział wszystkich pojęć na grupy logiczne aż do E. Deminga : Przewidywany stopień jednorodności i niezawodności przy możliwie niskich kosztach i dopasowaniu do wymagań rynku i J. Jurana Przydatność użytkowa.

Funkcjonują również definicje potoczne np.: Jakość to wszystko co można poprawić. Jakość to stopień uwolnienia wyrobu od wad i błędów lub Jakość- Spełnienie potrzeb klienta.

Definicje ekonometryczne starają się podać mierzalne cechy pojęcia jakości z uwzględnieniem cech produktu i kosztów ich uzyskania.

Normy ISO określają jakość jako zgodność z wymaganiami (ISO 9001:1994) a współczesna jej edycja ISO 9001:2000 mówi o ciągłym dostarczaniu wyrobu spełniającego wymagania Klienta, stałym doskonaleniu i zapobieganiu niezgodnościom.

Tak ogólne definicje jakości są stosunkowo mało przydatne w określaniu jakości prac informatycznych.

## Geneza norm ISO/ IEC

Normy pochodzą z różnych źródeł- np. ze stowarzyszeń inżynierskich (IEEE, Open Source), krajowych komitetów normalizacyjnych (np. BS, ANSI), stowarzyszeń producentów (QS, VDA) oraz prac własnych międzynarodowych komitetów normalizacyjnych. Proces tworzenia norm w ISO polega na uzgadnianiu treści normy w komitetach technicznych (ISO TC) i podkomitetach (ISO SC). Norma międzynarodowa powstaje w fazach:

Tablica 1. Cykl życia norm międzynarodowych

Oznaczenie	Etap	nazwa	dokument
10	Propozycja	FORMULATION	Proposal (TR)
20	Przygotowanie	DEFINITION	Working Draft (WD)
30	Uzgodnienia w komitetach	COMMENT	Committee Draft (CD)
40	Zapytania	ENQUIRY	Draft International Standard (DIS)
50	Aprobata	APPROVAL	Final Draft International Standard (FDIS)
60	Publikacja	IMPLEMENTATION	International Standard
90	Przegląd	REVIEW	
95	Wycofanie	WITHDRAWAL	

Tablica 2. Podetapy cyklu życia norm

00	Rejestracja
20	Początek działań
60	Zakończenie działań
90- 99	Decyzja

Procesu tworzenia normy sprawia, że każde zdanie w niej zawarte jest wynikiem licznych kompromisów, przemyśleń i uzgodnień ze wszystkimi zainteresowanymi stronami. Powoduje to, że zwykle normy zawierają wymagania minimalne, podstawowe a w wielu sytuacjach niewystarczające. Dotyczy to szczególnie zagadnień informatycznych gdzie istnieje dużo niespójności i dowolności interpretacji norm. Należy również zaznaczyć, że proces normalizacyjny jest długotrwały, a firmy szybko wytwarzają własne standardy często niezgodne z uznanymi normami ISO.

### Normy serii ISO 9000

Normy serii ISO 9000 (jakościowe) mogą być stosowane do produkcji dowolnego wyrobu lub dostarczania dowolnej usługi materialnej i niematerialnej a w szczególności do zagadnień informatycznych.

Tablica 3. Komitet Techniczny TC 176

TC 176	Zarządzanie jakością i zapewnienie jakości
TC 176/SC 1	Zasady i terminologia
TC 176/SC 2	Systemy jakości
TC 176/SC 3	Technologie związane

Normy dotyczące systemów zapewnienia jakości - serii ISO 9000 są wynikiem ewolucji norm związanych z zamówieniami wojskowymi dla US Army i NATO (system AQAP z lat 60-70). Wtedy pojawiła się w przemyśle koncepcja zapewnienia jakości- daleko szersze pojęcie niż wcześniejsze pojęcie kontroli jakości. Zapewnienie jakości rozumiane jest jako całokształt działań mających na celu definiowanie, kształtowanie i zarządzanie działaniami mającymi na celu spełnianie udokumentowanych wymagań Klienta. Bezpośredni wpływ na kształt normy ISO 9001 miała norma (brytyjska) BS 5750 (z 1979 roku) Obecnie obowiązującym dokumentem jest norma PN ISO 9001:2000 zastępująca normy ISO 9001, 9002 w wydaniu z roku 1996. Normy serii PN ISO 9000:1994, 1996 składają się z 20 (dla 9001, 19 dla 9002) rozdziałów. Norma ISO 9001:2000 ma 8 rozdziałów, określających zakres wymagań w procesie produkcji lub dostarczania usług.

W przypadku producentów towarów dla szerokiego kręgu konsumentów, można założyć, że ich przedstawicielem są jednostki certyfikujące.

Od strony formalnej uzyskanie certyfikatu zgodności działań z wymaganiami ISO 9001 jest umową pomiędzy jednostką certyfikującą a firmą. Umowa ta zobowiązuje firmę do stosowania metod pracy zgodnych z wymaganiami normy w interpretacji jednostki certyfikującej. Szczegółowe wymagania jednostek certyfikujących różnią się nieco mimo że wywodzą się z tej samej normy. Dobór jednostki certyfikacyjnej zależy od klientów firmy, mają oni zwykle pewne preferencje (np. dostawcy na rynki Beneluxu współpracują z KEMA, rynek niemiecki ceni certyfikaty TUV lub DQS, w okrętownictwie preferowane są certyfikaty BVQI, DNV i GL. Firmy certyfikacyjne mają zakresy kompetencji, określane przez krajowe centra akredytacji (w Polsce – Polskie Centrum Akredytacji, PCA [<http://www.pca.gov.pl>]).

Cele stosowania systemów zarządzania jakością wg ISO 9001:2000

- Trwałe zaspakajanie udokumentowanych potrzeb Klientów
- Stałe doskonalenie jakości działań i metod zarządzania
- Zaufanie, że wymagania jakościowe są i będą spełniane
  
- Wymagania dotyczące systemu jakości są i będą spełniane.

Tablica 4. Strony zainteresowane w utrzymaniu systemu jakości

Strona	Oczekiwanie
Klient	pożądana jakość wyrobów
Właściciel	korzystna inwestycja
Pracownicy	satysfakcja z pracy i korzyści materialne
Poddostawcy	stabilność kontraktu
Institucje nadzorujące	spełnienie obowiązków/ wymagań, zgodność z wymaganiami prawnymi

## Struktura norm serii ISO 9000

Obszerna literatura poświęcona jest normom jakościowym w modelu do roku 2000 (seria ISO 9000:1994) Ta grupa norm określa wymagania, związane z zapewnieniem jakości, terminologię, oraz wytyczne do akredytacji i certyfikacji. Normy te będą aktualne do 15 grudnia 2003, po tym czasie nastąpi ich wycofanie a firmy będą musiały się certyfikować na zgodność z nową edycją normy. Aktualna norma ISO 9001 nosi tytuł „Zarządzanie jakością”, podczas gdy poprzednia edycja była zatytułowana „Zapewnienie jakości” Oddaje to charakter zmian: od zapewnienia, że wszystkie czynności przebiegają zgodnie z planem działań do aktywnego zarządzania uzyskiwaniem założonego i coraz wyższego poziomu jakości.

Podstawowym wymaganiem norm jest stworzenie systemu jakości tj. struktury organizacyjnej, podziału odpowiedzialności, określenia procedur, procesów i zasobów umożliwiających utrzymanie i doskonalenie jakości.



System jakości powinien obejmować: wszystkie fazy życia wyrobu oraz określać:

- strukturę organizacyjną
- ogólną i szczegółową odpowiedzialność za jakość
- uprawnienia do wykonywania zadań
- sposób przepływu poleceń, informacji, dokumentów
- działania wykonywane w zakresie jakości

Ponadto, system powinien być efektywny, zapewniać, że produkty spełniają życzenia klientów a główny wysiłek działań ukierunkowany jest na prewencję.

Do roku 2000 normy jakościowe obejmowały

ISO 8402	zarządzanie jakością i zapewnienie jakości – terminologia
ISO 9000	Wytyczne – wybór modelu modele systemu jakości
9000-1	Wybór modelu ISO 9001
9000-2	Zastosowanie modelu ISO 9002
9000-3	Wytyczne do stosowania ISO 9001 w produkcji oprogramowania
9000-4	Niezawodność

ISO 9001:1994,	System jakości - Model zapewnienia jakości w projektowaniu, pracach rozwojowych, produkcji, instalowaniu i serwisie.
ISO 9002:1994	Systemy jakości - Model zapewnienia jakości w produkcji, instalowaniu i serwisie.
ISO 9003:1994	System jakości - Model zapewnienia jakości w kontroli i badaniach końcowych.

akredytacja i certyfikacja	wytyczne systemów jakości
EN 45001 EN 45011	ISO 9004-1 wytyczne
EN 45002 EN 45012	ISO 9004-2 usługi
EN 45003 EN 45013	ISO 9004-3 materiały
EN 45014	ISO 9004-4 poprawa jakości

ISO 10011	Audit ISO	ISO 10012	Pomiary
ISO 10011-1	Audit ISO	ISO 10012-1	Ap. kontr – pom.
ISO 10011-2	Auditorzy	ISO 10012-2	Pewność
ISO 10011-3	Program Auditów		

Wytyczne	
PN-ISO 10005:1998	Zarządzanie jakością. Wytyczne dotyczące planów jakości
PN-ISO 10007:1998	Zarządzanie jakością. Wytyczne dotyczące zarządzania konfiguracją
PN-ISO 10012-1:1998	Wymagania dotyczące zapewniania jakości wyposażenia pomiarowego - System potwierdzania metrologicznego wyposażenia pomiarowego
PN-ISO 10013:1998	Wytyczne opracowywania ksiąg jakości

Norma ISO 9001:1994 oddawała typowy cykl produkcyjny wyrobu i w 20 rozdziałach zestawiała wymagania, dotyczące kontaktów z Klientem, projektowania, zakupów i depozytów, zarządzania procesem produkcyjnym, kontroli jakości i dostarczania towaru do nabywcy. Norma określała również wymagania dotyczące procesów nie związanych bezpośrednio z produkcją (szkolenie, audit wewnętrzny, serwisowanie)

Tablica 5. Przegląd wymagań normy ISO 9001:1994

Rozdział	Tytuł
1	Zakres normy
2	Normy powołane
3	Definicje
4.01	Odpowiedzialność kierownictwa
4.02	System jakości
4.03	Przegląd umowy
4.04	Sterowanie projektowaniem
4.05	Nadzór nad dokumentacją i danymi
4.06	Zakupy
4.07	Wyrób dostarczony przez nabywcę
4.08	Oznaczanie wyrobu i jego identyfikowalność.
4.09	Sterowanie procesem
4.10	Kontrola i badania
4.11	Nadzorowanie wyposażenia do kontroli, pomiarów i badań
4.12	Status kontroli i badań
4.13	Nadzorowanie wyrobu niezgodnego z wymaganiami
4.14	Działania korygujące i zapobiegawcze
4.15	Transport, przechowywanie, pakowanie, dostarczanie
4.16	Zapisy dotyczące jakości
4.17	Wewnętrzne audyty jakości
4.18	Szkolenie
4.19	Serwis
4.20	Metody statystyczne

Zastosowanie tej normy bezpośrednio w IT było mało oczywiste, dlatego w 1996 roku opublikowano „Wytyczne do stosowania normy ISO 9001 do produkcji oprogramowania” (Guidelines for the Application of ISO 9001 to the development, supply and maintenance of software).

Wytyczne te wymagają podejścia procesowego do tworzenia oprogramowania, zastosowanego w później wydawanych normach. Podejście procesowe związane jest z tzw. cyklem Deminga: PDCA = plan, do, check, act, który powinien być „wbudowany” we wszystkie działania organizacji. Cykl Deminga jest włączony w normę ISO 9001:2000.

Tablica 6. Cykl Deminga

Plan (planowanie)	identyfikacja problemów
	Określenie ich przyczyn i skutków
	Zaplanowanie działań doskonalących
Do (realizacja- spróbuj)	Realizacja projektu zmiany
Check (weryfikacja)	Zebrańnię wyników działania
	Ocena rezultatów
	Określenie rzeczywistej poprawy
	Wykazanie usunięcia zaplanowanych problemów
Act (upowszechnienie)	Ustalenie procedur eliminujących występowanie problemu
	Zmiana procedur w celu wykluczenia ponownego problemu
	Szkolenie

Tablica 7. Przegląd normy ISO 9000 - 3: 1996

rozdz.	Tytuł	Odniesienie do ISO 9001:1994	Odniesienie do ISO 9001:2000
1	Przedmiot normy		
2	Normy związane		
3	Definicje		
4	System jakości – Struktura		
4.1	Odpowiedzialność kierownictwa	4.1	5.1
4.2	System jakości	4.2	5.2, 5.3
4.3	Wewnętrzne audyty systemu jakości	4.17	8.2, 8.5
4.4	Działania korygujące	4.14	8.3
5	System jakości – Działania dotyczące cyklu życia		
5.1	Postanowienia ogólne	4.2	4.1
5.2	Przegląd umowy	4.3	5.2
5.3	Specyfikacja wymagań nabywcy	4.3	7.2
5.4	Planowanie opracowywania	4.4	5.4, 7.1
5.5	Planowanie jakości	4.4	5.3, 7.1
5.7	Badania i zatwierdzanie	4.9	8.2
5.8	Odbiór	4.12	7.1, 7.5
5.9	Kopiowanie, dostarczanie i instalowanie	4.15	7.5,
5.10	Obsługiwanie	4.19	7.5
6	System jakości – Działania wspomagające		
6.1	Zarządzanie konfiguracją	ISO 10007	ISO 10007
6.2	Nadzór nad dokumentacją	4.5	4.2
6.3	Zapisy dotyczące jakości	4.16	4.2
6.4	Pomiary	4.10, 4.11, 4.12	8.2, 7.6
6.5	Zasady, praktyki i reguły	4.9	6.4, 7.3, 7.5
6.6	Narzędzia i techniki	4.9	6.1, 7.3, 7.5
6.7	Zakupy	4.6	7.4
6.8	Wyrób programowy włączany do dostaw	4.7	6.1, 7.2
6.9	Szkolenie	4.18	6.2

Norma ISO 9000-3 jest zestawieniem typowych zadań, które powinny być zarządzane w firmie produkującej oprogramowanie. Jak pokazuje praktyka,



niepełnienie tych wymagań jest powodem niezadowolenia Klienta albo kłopotów finansowych. Podobne wytyczne daje The TickIT Guide: A guide to Software Quality System Construction and Certification using ISO 9001:1994 Schemat TickIt jest brytyjskim schematem wdrażania systemu jakości dla firm produkujących oprogramowanie.

### Przykłady wymagań ISO 9000 - 3

#### 4.1.2 Odpowiedzialność kierownictwa nabywcy

Zaleca się, aby nabywca współpracował z dostawcą przez dostarczanie w porę niezbędnych informacji i uczestniczył w rozwiązywaniu pojawiających się problemów. Zaleca się, aby nabywca wyznaczył swojego przedstawiciela, odpowiedzialnego za współpracę z dostawcą w zakresie spraw dotyczących umowy. Zaleca się, aby przedstawiciel ten posiadał uprawnienia współmierne do potrzeb wynikających z zajmowania się sprawami dotyczącymi umowy, które obejmują, nie wykluczając innych, następujące zagadnienia:

- a) określanie wymagań nabywcy w stosunku do dostawcy;
- b) odpowiadanie na pytania dostawcy;
- c) zatwierdzanie propozycji dostawcy;
- d) zawieranie porozumień z dostawcą;
- e) zapewnienie, że organizacja przestrzega porozumień zawartych z dostawcą;
- f) określanie procedur i kryteriów odbioru;
- g) postępowanie z jednostkami programowymi dostarczonymi przez nabywcę, które okazały się nieprzydatne.

lub

#### 5.2.1 Przegląd umowy - Postanowienia ogólne

Zaleca się, aby dostawca dokonał przeglądu każdej umowy w celu zapewnienia, że:

- a) zakres umowy i wymagania są określone i udokumentowane;
  - b) ewentualne nieprzewidziane okoliczności i ryzyko są znane;
  - c) informacje dotyczące praw własności są odpowiednio zabezpieczone;
  - d) sprawy wszelkich wymagań różniących się od zawartych w ofercie są rozwiązane;
  - e) dostawca jest zdolny do spełnienia wymagań zawartych w umowie;
  - f) określona jest odpowiedzialność dostawcy w odniesieniu do prac wykonanych przez podwykonawców;
  - g) terminologia jest uzgodniona przez obie strony;
  - h) nabywca jest zdolny do spełnienia zobowiązań wynikających z umowy.
- Zaleca się, aby zapisy dotyczące przeglądów umów były przechowywane

Ważnym punktem wytycznych jest „6.1 Zarządzanie konfiguracją” zawierający m.in. takie wymaganie:

#### 6.1.1 Postanowienia ogólne

(...) Zaleca się, aby system zarządzania konfiguracją:

- a) identyfikował w niepowtarzalny sposób poszczególne wersje każdej jednostki programowej;

- b) identyfikował wersje każdej jednostki programowej, które wspólnie stanowią określoną wersję kompletnego wyrobu;
- c) identyfikował status wyrobów programowych w trakcie ich opracowywania lub dostarczanych i instalowanych;
- d) kontrolował równoczesne uaktualnianie danej jednostki programowej przez więcej niż jedną osobę;
- e) zapewniał koordynację uaktualniania różnych wyrobów w jednym lub kilku miejscach, jeśli jest to wymagane;
- f) identyfikował i śledził wszystkie działania i zmiany wynikające ze zmian zapotrzebowania, od etapu początkowego, aż do wydania kolejnej wersji.

Obecnie norma ISO 9000-3 jest zmieniana w celu uzyskania zgodności z obowiązującą ISO 9001:2000

Edycja roku 2000 zmniejszyła ilość norm jakościowych. Obecnie obowiązują następujące normy:

Tablica 8. Rodzina norm jakościowych ISO z uwzględnieniem ISO 9001:2000 i normy uzupełniające

<b>Norma ISO</b>	<b>Tytuł</b>
9000:2000	Systemy zarządzania jakością - podstawy i definicje
9001:2000	Systemy zarządzania jakością - wymagania
9004:2000	Systemy zarządzania jakością- Wytyczne dla prowadzenia doskonalenia
9000-3 /DIS	Wytyczne do stosowania normy ISO 9001 w produkcji oprogramowania
19011/DIS	Wytyczne do auditowania systemów jakości i/lub środowiska (w opracowaniu)
10005:1995	Zarządzanie jakością - wytyczne do planów jakości
10006:1997	Zarządzanie jakością - wytyczne do jakości w zarządzaniu projektami
10007:1995	Zarządzanie jakością - wytyczne do zarządzania konfiguracją
10012/DIS cz. 1	Zapewnienie jakości - wymagania dla wyposażenia pomiarowego - zgodność metrologiczna systemów pomiarowych
10012-2:1997	Zapewnienie jakości - wymagania dla wyposażenia pomiarowego - Wytyczne do prowadzenia pomiarów procesów
10013:1995,	Wytyczne do tworzenia ksiąg jakości
TR 10014:1998	Wytyczne do zarządzania kosztami jakości
10015:1999	Wytyczne do szkolenia
TS 16949:1999	Systemy jakości - Dostawcy przemysłu samochodowego - szczególne wymagania dla stosowania normy ISO 9001: 1994 w przemyśle samochodowym

Główne zasady zawarte w ISO 9001:2000 to:

- Ukierunkowanie na klienta: .
- Przywództwo
- Zaangażowanie ludzi
- Podejście procesowe
- Systemowe podejście do zarządzania.
- Ciągłe doskonalenie
- Oparte na faktach podejście do podejmowania decyzji
- Wzajemnie korzystne stosunki z dostawcami
- Spełnienie wymagań prawnych

Tablica 9. Przegląd wymagań normy ISO 9001:2000

Rozdział	Tytuł	Cel
0	Wprowadzenie	
1	Zakres normy	
2	Normy związane	
3	Terminy i definicje	
4.1	Wymagania ogólne	Ustanowienie, udokumentowanie, wdrożenie, utrzymanie i doskonalenie w sposób ciągły systemu zarządzania jakością
4.2	Ogólne wymagania dotyczące dokumentów	Utrzymywanie w stanie aktualnym i zapewnienie dostępności wszystkich dokumentów i danych odnoszących się do postanowień normy.
5.1	Zaangażowanie kierownictwa	Zapewnienie, aby naczelne kierownictwo odnosiło się z uwagą do tego, jak należy działać, ciągle troszcząc się o jakość.
5.2	Rozpoznawanie potrzeb klienta	Jednoznaczne i jasne zrozumienie potrzeb nabywcy (klienta) przez wszystkich w organizacji przedsiębiorstwa
5.3	Polityka jakości	Adekwatność polityki jakości do celów organizacji i do stałego doskonalenia
5.4	Planowanie jakości	Wprowadzanie zmian w sposób kontrolowany i utrzymywanie spójności systemu zarządzania jakością w czasie tych zmian
5.5	Sterowanie	Określenie i ogłoszenie funkcji i ich powiązań, w celu wzmocnienia skuteczności zarządzania jakością
5.6	Przegląd wykonywany przez kierownictwo	Zaprezentowanie osiągnięcia wymaganej jakości oraz skuteczności działania systemu jakości.
6.1	Zaopatrywanie w zasoby	Określenie i terminowe dostarczanie zasobów, w celu wdrażania i doskonalenia procesy systemu zarządzania jakością, i uzyskania zadowolenia klienta
6.2	Zasoby ludzkie	Ciągłe doskonalenie kwalifikacji całego personelu zaangażowanego w działania związane z jakością
6.3	Infrastruktura	Dostarczanie i zarządzanie urządzeniami niezbędnymi do osiągnięcia zgodności wyrobu



6.4	Środowisko pracy	Identyfikowanie i zarządzanie czynnikami ludzkimi i fizycznymi środowiska pracy niezbędnymi do osiągnięcia zgodności wyrobu
7.1	Planowanie realizacji procesów	Ustalenie i zaplanowanie procesów produkcyjnych, instalowania i serwisu mających bezpośredni wpływ na jakość.
7.2	Procesy związane z klientem	Określenie udokumentowanych i nieudokumentowanych potrzeb klienta oraz wymagań prawnych i innych
7.3	Projektowanie i/lub prace rozwojowe	Zapewnienie takiej jakości projektowej wyrobu, aby ryzyko podjęcia jego produkcji było zredukowane do minimum
7.4	Zakupy	Zapewnienie, aby nabywane od podwykonawców zakupy spełniały określone wymagania
7.5	Działania produkcyjne i prowadzenia obsługi	Nadzorowanie produkcji i prowadzenia obsługi a także zabezpieczenie wyrobu na wszystkich etapach jego realizacji
7.6	Nadzorowanie urządzeń do pomiarów i monitorowania	Uzyskanie zaufania do danych otrzymanych z kontroli, pomiarów i badań.
8.1	Planowanie	Określenie, planowanie i wdrożenie działań pomiarowych i monitorowania w celu zapewnienia zgodności i osiągnięcia doskonalenia
8.2	Pomiary i monitorowanie	Wdrożenie i utrzymanie metod pomiaru zadowolenia klienta, zgodności systemu zarządzania, procesów i wyrobów w celu ich doskonalenia
8.3	Nadzór nad niezgodnościami	Zapewnienie, że wyroby niezgodne nie zostaną wykorzystane lub zainstalowane przez przeoczenie razem z tymi, które są zgodne.
8.4	Analiza danych	Podejmować decyzje co do stosowanych procesów i wytwarzanych wyrobów w oparciu o realne liczby i fakty.
8.5	Doskonalenie	Ułatwienie ciągłego doskonalenia systemu zarządzania jakością

Edycja roku 2000 normy lepiej stosuje się do działań w IT, ale jej wdrożenie bez dodatkowych uściśleń jest trudne. ISO 9001 pozostawia zdefiniowanie co to jest jakość samej organizacji.

## ISO/IEC JTC1

Normy ISO związane z informatyką rozwijane są przez JTC1 (Joint Technical Committee - Information technology) - wspólny komitet techniczny, tworzony przez ISO oraz IEC (International Electrotechnical Commission) wraz z podkomitetami:

Tablica 10. Podkomitety JTC1

JTC 1/SC 2	Coded character sets
JTC 1/SC 6	Telecommunications and information exchange between systems
JTC 1/SC 7	Software and system engineering
JTC 1/SC 11	Flexible magnetic media for digital data interchange
JTC 1/SC 17	Identification cards and related devices
JTC 1/SC 22	Programming languages, their environments and system software interfaces
JTC 1/SC 23	Optical disk cartridges for information interchange
JTC 1/SC 24	Computer graphics and image processing
JTC 1/SC 25	Interconnection of information technology equipment
JTC 1/SC 27	IT Security techniques
JTC 1/SC 28	Office equipment
JTC 1/SC 29	Coding of audio, picture, multimedia and hypermedia information
JTC 1/SC 31	Automatic identification and data capture techniques
JTC 1/SC 32	Data management and interchange
JTC 1/SC 34	Document description and processing languages
JTC 1/SC 35	User interfaces
JTC 1/SC 36	Learning technology

Inżynierią oprogramowania zajmuje się podkomitet JTC1/SC7 z grupami roboczymi.

Tablica 11. Grupy robocze JTC 1/SC 7

WG2	SW system documentation
WG4	support tools and environments
WG6	SW evaluations and metrics
WG7	SW life cycle
WG8	Life cycle support process
WG9	SW integrity
WG10	SW processes assessment
WG11	SW engineering daat
WG12	SW functional measurement
WG13	SW measurement process
WG17	ODP enterprise lang
WG18	Quality management
sWG18	Revision ISO 9000 – 3
WG19	ODP and modelling lang
SWG1	Planning
SWG2	Vocabulary
SWG3	Process architecture
SWG4	WEB and communication support

Wynikiem prac JTC1 jest ponad 800 norm związanych z różnymi aspektami IT- od szczegółowych, technicznych do bardzo ogólnych - cyklu życia oprogramowania i systemu .

Podstawową normą terminologiczną, opracowaną w latach 1970 i stale aktualizowaną jest rodzina ISO/IEC 2382:

Tablica 12. ISO/IEC 2382 Information technology , Data processing Information processing systems

Część	rok wydania	Tytuł
1	1993	Fundamental terms
2 :	1976	Arithmetic and logic operations
3	1987	Equipment technology
4	1987	Organization of data
5	1989	Representation of data
6	1987	Preparation and handling of data
7 FDIS	1996	Computer programming;
8	1998	Security
9	1995	Data communication
10	1979	Operating techniques and facilities
11	1987	Processing units
12	1988	Peripheral equipment
13	1996	Computer graphics
14 :	1997	Reliability, maintainability and availability
15	1998	Final DIS Programming languages; Rev 1985
16 :	1996	Information theory
17 :	1999	Databases; Corrected first edition
18	1987	Distributed data processing
19	1989	Analog computing
20	1990	System development
21	1985	Interfaces between process computer systems and technical processes
22	1986	Calculators (kalkulatory)
23	1994	Text processing
24	1995	Computer-integrated manufacturing
25	1992	Local area networks
26	1993	Open systems interconnection
27	1994	Office automation
28	1995	Artificial intelligence - Basic concepts and expert systems
29 :	1999	Artificial intelligence - Speech recognition and synthesis Final DIS
30 FDIS	1999	Artificial Intelligence - Computer vision
31	1997	Artificial intelligence - Machine learning
32 :	1999	Electronic mail
33	1999	CD Hypermedia and multimedia
34	1999	FDIS Artificial intelligence - Neural networks



## Jakość oprogramowania

Definicję jakości oprogramowania podaje czteroczęściowa norma ISO 9126. Prace nad tą normą trwały od 1985 roku. Norma określa cechy użytkowe oprogramowania definiując jakość oprogramowania w sześciu kategoriach i 22 charakterystykach. Norma nie narzuca producentowi konieczności monitorowania wszystkich charakterystyk, wskazuje jednak na różne aspekty oceny jakości oprogramowania.

Tablica 13. Charakterystyki jakości oprogramowania wg ISO 9126 (część 1)

Charakterystyka	Podcharakterystyka	Cecha oprogramowania które dotyczy:
Functionality Funkcjonalność	Suitability	Wyglądu i dostosowania zestawu funkcji do określonych zadań
	Accuracy	Zdolności do uzyskania określonych rezultatów
	Interoperability	Zdolności do interakcji z określonymi systemami Zdolność do pracy na wielu platformach ma zastosowanie pozwala uniknąć możliwych niejasności związanych z zastępowalnością
	Compliance	Zgodności ze standardami aplikacji, konwencjami lub regulacjami w prawach i podobnych nakazach
	Security	Zdolności do zapobiegania nieautoryzowanemu dostępowi, przypadkowemu bądź umyślnemu, do programów i danych
Reliability niezawodność	maturity	Częstotliwość pomyłek powodowanych przez błędy oprogramowania
	fault tolerance	Zdolności do utrzymania określonego poziomu wyników w przypadkach błędów oprogramowania lub naruszenia jego specyficznego interfejsu
	recoverability	Zdolności do ponownego ustalenia poziomu osiągnięć i przywrócenia danych bezpośrednio narażonych w wypadku błędu i wyгоды ich odzyskiwania
	Understandability	Wysiłku użytkownika koniecznego na rozpoznanie systemu i jego odpowiedniego zastosowania
Usability użyteczność	Learnability	Wysiłku użytkownika koniecznego do nauczenia się aplikacji (na przykład, przebieg operacji, wejścia, wyjścia).
	Operatibility	Wysiłku użytkownika konieczny do sterowania programem i przeprowadzanie procesów
	Attractiveness	Atrakcyjności wizualnej
	time- behaviour	Czas obróbki, reakcji i przeprowadzanych operacji

Efficiency wydajność	resource behaviour	Ilość użytych zasobów i czasu trwania funkcji opartych o ich zużycie
	Analysability	Wkład pracy potrzebny na diagnozę niedoborów bądź przyczyn błędów, lub na rozpoznanie modułów do modyfikacji
Maintanability pielęgnownalność	Changeability	Wkładu pracy potrzebnego na dokonanie zmian, usunięcie błędu lub na zmianę środowiska.
	Stability	Ryzyka wystąpienia niespodziewanych efektów zmian
	Testability	Wkładu pracy potrzebnego na zatwierdzenie zmodyfikowanego oprogramowania
	Adaptability	Możliwości adaptacji do innych środowisk bez stosowania innych działań
Portability przenośność	Instalability	Wysiłku potrzebnego na instalację oprogramowania w określonym środowisku
	Coexistence/ Conformance	Dostosowania do standardów lub norm odnoszących się do przenośności i współpracy
	Replaceability	Możliwości i wysiłku potrzebnego do zastąpienia tego oprogramowania (np. przeniesienie danych, dokumentów baz danych)

Tablica 14. Części normy ISO 9126

ISO 9126 (część 1)	model jakości i charakterystyki
ISO 9126 (część 2)	miary wewnętrzne
ISO 9126 (część 3)	miary zewnętrzne
ISO 9126 (część 4)	charakterystyki i miary

Producenci oprogramowania do szerokiego korzystania, oraz oprogramowania wbudowanego w sprzęty codziennego użytku powinni zwrócić uwagę na normę ISO WD 20282: Usability of everyday products (powinna obowiązywać od 2001). Norma ta określa informacje, które powinien otrzymać konsument aby mógł ocenić przydatność produktu do jego wymagań. Norma określa wymagania testowe, cechy charakterystyczne "normalnego użytkownika" i zamierzony sposób korzystania z produktu. Norma zawiera również wytyczne do projektowania produktu dla możliwie szerokiej grupy użytkowników w celu osiągnięcia.

- Uniwersalności
- Dostępności
- Uwzględnienia poziomu wykształcenia użytkownika.

Norma zwraca uwagę na produkty, które powinny być stosowane bez szczególnego wykształcenia i skomplikowanych instrukcji i szkolenia w powiązaniu z ergonomią, dostosowaniem gabarytów, projektem wizualnym, komunikacją, kulturą, wiekiem i płcią użytkownika. Norma określa również metody testowania produktów codziennego użytkowania.

Norma ISO 14598:1999 „Proces oceny jakości oprogramowania” jest przeznaczona dla nabywców oprogramowania i zwraca uwagę na następujące cechy systemów:

Funkcjonalność	Efektywność
Niezawodność	Produktywność
Użyteczność	Bezpieczeństwo
Efektywność	Satysfakcję klienta
Utrzymywalność	Przenośność

Tablica 15. ISO 14598:1999 Information technology - Software product evaluation

Rok wydania	Tytuł	Przeznaczenie
1999	1: Przegląd	Określenie celów
2000	2: Planowanie i zarządzanie	Identyfikacja produktu
2000	3: Procesy dotyczące dostawcy	Specyfikacja jakościowa, dobór miar jakości
1999	4: Procesy związane z nabywcą	Poziomy (rating)
1998	5: Procesy ewaluacji oprogramowania	Poziom jakości, pomiar wydajności
DIS	6: Dokumentowanie procesów ewaluacji	Analiza wyników

Obecnie trwają prace zmieniające normę 9126 – planowane jest połączenie jej z ISO 20282 (WD), ISO 14598, ISO 11219 i częścią SPICE w model SQUARE. Model SQUARE ma na celu ukazanie szerokiego przeglądu cyklu życia oprogramowania - dostawa, rozwój, eksploatacja i pielęgnacja.

SQUARE= Software Product Quality Requirement and Evaluation

Tablica 16. Zmiany w normie 9126 - rozszerzenie i uzupełnienie ISO 25000- 25010 25020

25000	Wstęp, model odniesienia i przewodnik po SquaRE.
25001	Planowanie i zarządzanie
25010	Model jakości
25020	Metryczny model odniesienia i przewodniki.
25021	Podstawy pomiarów .
25022	Miary wewnętrzne
25023	Miary zewnętrzne
25024	Jakość przy użyciu miar.
25025	Dokumentacja modułu oceniającego.
25030	Wymagania jakościowe.



25040	Przegląd procesu oceny .
25041	Wymagania (procesy) dla programistów .
25042	Wymagania (procesy) dla rozwijających projekt (sw developers)
25043	Wymagania (procesy) dla oceniających.

## Tworzenie oprogramowania

W 1995 ISO/IEC opublikował normę 12207 cykl życia oprogramowania. Norma ta określa cykl życia oprogramowania jako części większego systemu, samodzielnego produktu oraz usługi. Norma opisuje procesy związane z tworzeniem oprogramowania: zakupy, dostawy, opracowywanie, korzystanie i pielęgnację oraz procesy wspomagające tworzenie oprogramowania i procesy organizacyjne.

Norma określa wymagania dla pięciu procesów podstawowych ośmiu procesów pomocniczych oraz czterech procesów organizacyjnych

Tablica 17. Procesy w ISO 12207:1995 Software life cycle processes

Procesy	
podstawowe	pomocnicze
5.1 Zakup	6.1 Dokumentowanie
5.2 Dostawa	6.2 Zarządzanie konfiguracją
5.3 Rozwój	6.3 Zapewnienie jakości
5.4 Korzystanie	6.4 Weryfikacja
5.5 Pielęgnacja	6.5 Walidacja
	6.6 Wspólne przeglądy
	6.7 Audit
	6.8 Rozwiązywanie problemów
Organizacyjne	
7.1 Zarządzanie	7.2 Infrastruktura
7.3 Doskonalenie	7.4 Szkolenie

Norma ISO/IEC 12207 określa zasady współpracy między organizacjami i może z powodzeniem być stosowana do tworzenia skomplikowanych projektów informatycznych w dużej skali. Pewne procesy zachodzą w organizacji dostawcy oprogramowania, inne u użytkownika, niektóre dokonywane są wspólnie. Norma ukazuje najważniejsze zagadnienia, które powinny być nadzorowane. Każdy proces cyklu życia podzielono na zbiór działań, a działania rozłożono na zadania. Dla 12207 opracowano przewodnik ISO IEC TR Information technology – Guide for ISO/IEC 12207.

Norma ISO/IEC 12207 nie narzuca żadnego modelu jakości oprogramowania ani modelu tworzenia oprogramowania (kaskadowy, V, spiralny, inne) pokazując podstawowe procesy, które zachodzą niezależnie od wybranego w organizacjach modelu tworzenia oprogramowania.

Norma 12207, podobnie jak 17799 jest przeglądem powodów niepowodzeń różnych projektów programistycznych i poradnikiem, jak ich unikać.

Uzupełnieniem ISO/ IEC 12207 jest przewodnik ISO/IEC TR 16326: Software Engineering - Guide for the application of ISO/IEC 12207 to project management. Podane są w nim praktyczne uwagi do zarządzania projektem oraz powiązania z ISO 10006. Przewodnik cytuje uwagi Watta Humphrey (1992, cytuje za ISO 16326) pokazujące czym różni się zarządzanie projektem programistycznym od innych:

- Oprogramowanie jest bardziej skomplikowane
- Zmiany w oprogramowaniu wydają się być łatwe do wprowadzenia
- Wiele problemów sprzętowych wynika ze zmian oprogramowania
- Niskie koszty reprodukcji powodują, że w oprogramowaniu nie ma naturalnej dyscypliny przy zwalnianiu do produkcji
- Produkcja oprogramowania nie jest oparta o podstawy inżynierskie, brakuje technik testowania wydolności procesu i prototypowania.
- Oprogramowanie jest często elementem integrującym systemy, komplikując je i narażając na dokonywanie zmian w ostatniej chwili.
- Oprogramowanie jest czytelne dla użytkowników, przez co jest narażone na zmiany wymagań użytkownika

Przewodnik ISO/IEC TR 15846:1998 Information technology - Software life cycle processes - Configuration management określa wymagania dla zarządzania konfiguracją w projekcie softwarowym.

ISO 12207 wskazuje wymagania dotyczące procesu tworzenia oprogramowania. Standard CMM (Capability Maturity Model for Software) opracowany przez Software Engineering Institute przy Carnegie Mellon University. Mówi on o dojrzałości organizacji do realizacji projektu. Model CMM był wykorzystywany w procedurach klasyfikacji potencjalnych wykonawców oprogramowania dla Departamentu Obrony USA.

Tablica 18. Poziomy dojrzałości w modelu CMM

Poziom	Opis	Praktyka
Początkowy	procesy chaotyczne, nie zarządzane	Wdrożono podstawowe praktyki
Powtarzalny	procesy zindywidualizowane – wdrożono zarządzanie projektami	Planowana, dyscyplinowana, weryfikowana i śledzona wydajność
Zdefiniowany	procesy zinstytucjonalizowane-zdefiniowane	Zdefiniowane standardowe procedury Wykonywane standardowe procesy
Zarządzany	procesy i informacje zwrotne dla sterowania	Określone mierzalne cele jakości Zarządzanie wydajnością
Optymalizowany	procesy + informacje zwrotne wpływające na doskonalenie	Doskonalenie zdolności organizacji Doskonalenie efektywności procesów

ISO/IEC 12207 zwraca uwagę na cykl życia oprogramowania odseparowany jest on jednak od procesów zachodzących w organizacjach Rozszerzeniem tej normy jest ISO/IEC 15228 CD która umiejscawia cykl życia oprogramowania w szeroko rozumianym kontekście współpracujących przedsiębiorstw i integruje wykorzystanie oprogramowania z innymi procesami biznesowymi.

Tablica 19. Procesy w ISO 15288 /CD2 :2000 Standard on System Life Cycle Processes.

Enterprise processes	Project processes	Technical Processes
		Stockholders definition
Enterprise processes	Planning	Requirements process
Investment processes	Assessment	Architectural
System life Management	Control	Implementation
Resource processes	Decision Making Process	Integration
	Risk Management process	Verification
<b>Agreement processes</b>	Configuration Management	Transition
Acquisition		Validation
Supply		Operation and Process
		Disposal

Podstawowe	Projektowanie	Procesy techniczne
Przedsiębiorstwo Inwestycje Cykl życia Zasoby	Planowanie Szacowanie Sterowanie Procesy decyzyjne Zarządzanie ryzykiem Zarządzanie konfiguracją	Udziałowcy Wymagania Architektura Implementacja Integracja Weryfikacja Zmiany Walidacja Eksploatacja Zakończenie eksploatacji
<b>Kooperacja</b>		
Zakupy Dostawy		

## SOFTWARE PROCESS ASSESSMENT TR ISO 15504 (SPICE)

Raport techniczny ISO 15504 TR jest połączeniem wymagań ISO 12207 i ISO 15228 z modelem CMM. Zawiera wytyczne do oceniania dojrzałości procesów:

Tablica 20. Model oceny dojrzałości wg ISO 15504 TR (SPICE)

Poziomy	Początkowy	Powtarzalny	Zdefiniowany	Zarządzany	Optymalizowany
Procesy związane z Klientem					
Rozwój oprogramowania					
Procesy wspomagające					
Zarządzanie					
Organizacja					



Tablica 21. Struktura normy ISO 15504 CD SPICE

1 Wstęp i przewodnik		9 Słownik	
7 Doskonalenie procesów	8 Dostawca	6 Szkolenie auditorów	
3 Ocena		4 Przewodnik do auditu	
5 Narzędzia do auditowania	2 Zarządzanie procesem		

Aktualnie ISO 15504 jest zamieniane w pięcioczęściowy standard posiadający cechy zarządzania procesowego.

Tablica 22. ISO Procesy wg 15504:1999 SPICE

Customer		Engineering		Support	
CUS.1 Acquire software		ENG.1 Develop system requirements and design		SUP.1 Develop documentation	
CUS.2 Manage customer needs		ENG.2 Develop software requirements		SUP.2 Perform configuration management	
CUS.3 Supply software		ENG.3 Develop software design		SUP.3 Perform quality assurance	
CUS.4 Operate software		ENG.4 Implement software design		SUP.4 Perform work product verification	
CUS.5 Provide customer service		ENG.5 Integrate and test software		SUP.5 Perform work product validation	
		ENG.6 Integrate and test system		SUP.6 Perform joint reviews	
		ENG.7 Maintain system and software		SUP.7 Perform audits SUP.8 Perform problem resolution	
Management					
MAN.1 Manage the project		MAN.2 Manage quality		MAN.3 Manage risks MAN.4 Manage subcontractors	
Organization					
ORG.1 Organizational alignment		ORG.1 Engineer the business		ORG.2 Define the process	
ORG.3 Improve the process		ORG.4 Provide skilled human resources		ORG.5 Provide software engineering infrastructure	
ORG.2 Improvement process					
ORG.2.1 Process establishment		ORG.2.2 Process assessment		ORG.2.3 Process improvement	
ORG.3 Human resource management		ORG.4 Infrastructure		ORG.5 Measurement ORG.6 Reuse	

## Bezpieczeństwo informacji - TR ISO 13335 i ISO 17799

Wytyczną dla zarządzania bezpieczeństwem informacji daje Raport Techniczny - ISO/IEC TR 13335. System zabezpieczeń chroni ustalony zbiór zasobów przed stratami, które mogłyby powstać w wyniku podatności zasobów na zagrożenia. Zagrożeniom przeciwdziałają zabezpieczenia ograniczające ryzyko. Niektóre zagrożenia mogą wymagać zastosowania kilku zabezpieczeń. Zabezpieczenia rzadko bywają pełne, pozostawiając ryzyko resztkowe. System zabezpieczeń funkcjonuje w pewnym środowisku podlegając ograniczeniom prawnym, finansowym, i technicznym.

Środowisko		Ograniczenia		
Zasoby	podatności	zabezpieczenie	ryzyko	zagrożenie

Tablica 23. ISO/IEC/TR 13335

Część 1 (PN-I-13335-1) Wytyczne do zarządzania bezpieczeństwem systemów informatycznych	terminologia, związki między pojęciami
	podstawowe modele.
Część 2: Planowanie i zarządzanie bezpieczeństwem systemów informatycznych:	różne podejścia do prowadzenia analizy ryzyka
	plany zabezpieczeń
	rola szkoleń i działań uświadamiających
	stanowiska pracy w instytucji związane z bezpieczeństwem
Część 3: Techniki zarządzania bezpieczeństwem systemów informatycznych:	sformułowanie trójpoziomowej polityki bezpieczeństwa,
	rozwinięcie problematyki analizy ryzyka,
	rozwinięcie problematyki implementacji planu zabezpieczeń,
	reagowanie na incydenty.
Część 4: Wybór zabezpieczeń:	klasyfikacja i charakterystyka różnych form zabezpieczeń,
	dobór zabezpieczeń ze względu na rodzaj zagrożenia i rodzaj systemu.
Część 5: Zabezpieczenie dla połączeń z sieciami zewnętrznymi	dobór zabezpieczeń stosowanych do ochrony styku systemu z siecią zewnętrzną.

Rozwinięciem TR ISO 13335 jest ustanowiona w 2000 roku norma ISO 17799 (przewidziana do ustanowienia w Polsce w 2003 roku). Norma ISO/IEC 17799:2000 **Praktyczne zasady zarządzania bezpieczeństwem informacji** określa sposoby postępowania z informacją w firmie, zwracając uwagę na poufność, dostępność i spójność danych. Jest to szczególnie ważne w organizacjach przetwarzających dane poufne, prowadzących produkcje specjalna

oraz obawiających się nieuczciwych działań konkurencji. Norma wskazuje podstawowe zagadnienia i określa metody i środki konieczne dla zabezpieczenia informacji.

Norma ta jest szczególnie istotna dla firm przetwarzających dane, które z różnych powodów powinny być chronione. Dotyczy to m.in. dostawców usług internetowych, firm przetwarzających dane klientów, instytucji państwowych, finansowych, firm notowanych na giełdach i wielu innych. Zasady zabezpieczania informacji są również istotne dla firm tworzących oprogramowanie.

System zarządzania bezpieczeństwem informacji nie jest kosztowym systemem zarządzania i w przeciwieństwie do ISO 9001 lub ISO 14001 można go wdrożyć stosunkowo niewielkim wysiłkiem organizacyjnym i finansowym przez wprowadzenie zmian w procedurach i urzędzeniach.

Korzyści płynące z wdrożenia systemu zabezpieczenia informacji:

- Zgodność z wymaganiami prawnymi;
- Pewniejsze działanie w sytuacji zagrożeń powodowanych przez czynniki wewnętrzne i zewnętrzne;
- Ochrona żywotnych interesów organizacji;
- Wysoka sprawność pracowników osiągnięta poprzez odpowiednie szkolenia;
- Lepsza organizacja pracy
- Oszczędności finansowe;
- Utrzymanie przewagi nad konkurencją;
- Poprawa wizerunku firmy;

Straty wynikające bezpośrednio z błędów związanych z niezabezpieczoną informacją:

- Straty materialne (przeciek informacji do konkurencji, gorsze warunki kontraktu);
- Odpowiedzialność karna lub cywilna;
- Bałagan organizacyjny
- Niezgodność lub sprzeczność danych;
- Niewłaściwe decyzje;
- Niezgodne z wymaganiami wykonanie pracy, niekompetencja;

Norma ISO 17799 definiuje bezpieczeństwo informacji w trzech kategoriach:

- Poufność
- Integralność
- Dostępność

ISO 17799 skupia się na następujących zagadnieniach (łącznie ok. 250 zadań szczegółowych):

- Polityka bezpieczeństwa



- Struktura organizacyjna zabezpieczeń
- Klasyfikacja i kontrola zasobów
- Personel – przebieg zatrudnienia
- Zabezpieczenia fizyczne i środowiskowe - zasady dostępu do obiektów
- Zabezpieczenia sieci i komputerów
- Kontrola dostępu do systemów i do aplikacji
- Pielęgnacja systemu - zasady wymiany oprogramowania, kontroli wersji itp
- Planowanie ciągłości pracy - postępowanie w wypadku zagrożeń
- Zgodność z wymaganiami prawnymi - zapewnienie zgodności działania organizacji z wymaganiami prawnymi i kontraktowymi.

Tablica 24. ISO 17799:2001/BS 7799 Code of Practice for Information Security Management

Rozdział	Tytuł	Cel
3	Policy	
3.1	Protection Policy	To achieve top-level support and control of information protection.
4	Organizational Information Protection	
5.1	Organizational Infrastructure	Effectively manage information protection throughout the organization.
5.2	Third Party Access Controls	To control the potential effects of access by third parties.
5	Control and sensitivity of assets	
5.1	Accountability	Appropriate control of information assets.
5.2	Information classification	Provide assurance that protection is appropriate for all information assets.
6	People Issues	
6.1	Protection in personnel specifications and decisions	To control people-related risks.
6.2	Security Training	Provide assurance that the nature of threats to information are understood by those with legitimate access and that individuals with said access have a proper understanding of controls.
6.3	Incident Response	To minimize harm from protection-related incidents and to adapt protection over time.

7	Physical Protection	
7.1	Physically Protected Places	Prevent inappropriate interaction between unauthorized and authorized people and systems.
7.2	Hardware security	Assure business continuity and reduce exposures to loss.
8	System and Infrastructure Management	
8.1	Roles and Responsibilities	Properly operate facilities.
8.2	System planning and acceptance	To minimize the risk of systems failure. Advance planning and preparation are required to ensure the availability of adequate capacity and resources.
8.3	Protection from malicious software	To safeguard the integrity of software and data. Precautions are required to prevent and detect the introduction of malicious software.
8.4	Housekeeping	To maintain the integrity and availability of IT services. Housekeeping measures are required to maintain the integrity and availability of services.
8.5	Network management	To ensure the safeguarding of information in networks and the protection of the supporting infrastructure.
8.6	Media handling and security	To prevent damage to assets and interruptions to business activities. Computer media should be controlled and physically protected.
8.7	Data and software exchange	To prevent loss, modification or misuse of data. Exchanges of data and software between organizations should be controlled.
9	System access control	
9.1	Business requirement for system access	To control access to business information.
9.2	User access management	To prevent unauthorized computer access.
9.3	User responsibilities	To prevent unauthorized user access.
9.4	Network access control	Protection of networked services.
9.5	Computer access control	To prevent unauthorized computer access.

9.6	Application access control	To prevent unauthorized access to information held in computer systems.
9.7	Monitoring system access and use	To detect unauthorized activities.
10	Systems development and maintenance	
10.1	Security requirements of systems	To ensure that security is built into IT systems.
10.2	Security in application systems	To prevent loss, modification or misuse of user data in application systems.
10.3	Security of application system files	To ensure that IT projects and support activities are conducted in a secure manner.
10.4	Security in development and support environments	To maintain the security of application system software and data.
11	Business continuity planning	
11.1	Aspects of business continuity planning	To have plans available to counteract interruptions to business activities.
12	Compliance	
12.1	Compliance with legal requirements	To avoid breaches of any statutory, criminal or civil obligations and of any security requirements.
12.2	Security reviews of IT systems	To ensure compliance of systems with organizational security policies and standards.

### Inne normy jakościowe związane z IT

Inne normy jakościowe są oparte na koncepcjach ISO 9001, określają podwyższone wymagania dotyczące specyficznych dziedzin (np. w przemyśle farmaceutycznym - GMP, w przemyśle samochodowym VDA 6.1, QS9000, w przemyśle lotniczym AS9000). Wymagania specyficzne dla przemysłu samochodowego i lotniczego dotyczące całego procesu wytwarzania oprogramowania zawarto w normie ISO TS 16949

Grupa QUEST (Quality Excellence for Suppliers of Telecommunications) opracowało podręcznik jakości dla oprogramowania znany jako TL 9000:1999 zawierający wymagania dla dostawców sprzętu, oprogramowania i usług dla telekomunikacji. Jest on rozszerzeniem wymagań ISO 9001:1994 o szczególne potrzeby telekomunikacji, m.in. o wymaganie doskonalenia oraz analizowanie



zadowolenia Klientów. W drugiej części TL9000 określone są miary jakości produktów, odnoszą się one w większości do niezawodności i dostępności systemów, kładąc nacisk głównie na sprzęt telekomunikacyjny. QUEST prowadzi prace nad aktualizacją TL9000 do zgodności z ISO 9001:2000.

Wytyczne do auditów informatyki acz znacznie mniej rozbudowane podaje również organizacja ISACA Information Systems Audit And Control Association (skupiająca w większym stopniu auditorów systemów informatyki bankowej i finansowej)

Własne standardy tworzenia i odbioru oprogramowania określa wojsko. Normy wojskowe pochodzą z dwóch podstawowych źródeł: amerykańskie wymagania wojskowe – MIL i standardy NATO – AQAP.

Wymagania NATO bazują na standardach ISO dając dodatkowe wymagania. Uzupełnienia dotyczą szczególnych wymagań związanych z mniejszą rolą czynników ekonomicznych na rzecz użytkowych oraz zwiększonymi wymaganiami dotyczącymi poddostawców i ochrony poufności. Wymagania NATO kładą większy nacisk na zarządzania konfiguracją i wymagają prawa do auditów na każdym etapie produkcji przez rządowego przedstawiciela. Podstawowe normy jakościowe NATO to AQAP 110 i AQAP 120 a w zakresie produkcji oprogramowania AQAP 150 (NATO Quality Assurance Requirements for Software Development) AQAP 160 (NATO integrated quality requirements for software throughout the life cycle), uzupełnione o STANAG 4107 Standardization Agreement (luty 199) i AQAP 170. W Polsce certyfikacją AQAP zajmuje się Wojskowe Biuro Służb Normalizacyjnych.

## Zakończenie

Mimo dużej ilości norm związanych z informatyką w praktyce mam dyskomfort związany z brakiem powszechnie uznawanych standardów dotyczących zagadnień praktycznych takich jak standardy kodowania ogólnie przyjętych struktur danych, zasady nazewnictwa i korzystania z obiektów programistycznych (istniejące zalecenia nie są normami i w praktyce nie obowiązują). Przykładem nieszczęśliwego braku norm jest np. comment, C, \*, //, \*, #, {, (\*, /\* itp.... lub łączenie tekstów: +, &, ., concat, strepy, czy jeszcze jakoś inaczej :-)

Można oczekiwać że w najbliższym czasie działalność normalizacyjna rozwine się w inżynierii oprogramowania i ułatwi życie programistom.

## Literatura

1. The TickIT Guide: A guide to Software Quality System Construction and Certification using ISO 9001:1994
2. Bolesław Szomański Zapewnienie jakości w środowisku IT Wydział Inżynierii Produkcji Politechniki Warszawskiej

3. David H. Kitson ISO/IEC 15504 - Overview and Status Software Engineering Institute Carnegie Mellon University Pittsburgh Software Engineering Institute
4. Baza danych norm [www.perinorm.com](http://www.perinorm.com) - zestawienie norm dotyczących informatyki
5. Nigel Bevan Quality in Use: Meeting User Needs for Quality Journal of System and Software, 1999

## Normy

CMM Capability Maturity Model for Software	ISO 19011 DIS Wytyczne do auditowania systemów jakości i/lub środowiska
ISACA Information Systems Audit And Control Association	ISO 20282 WD Usability of everyday products
ISO 10005:1998 Zarządzanie jakością. Wytyczne dotyczące planów jakości	ISO 25000 WD Ewaluacja oprogramowania
ISO 10006:1997 Zarządzanie jakością - wytyczne do jakości w zarządzaniu projektami	ISO 8402 zarządzanie jakością i zapewnienie jakości – terminologia
ISO 10007:1998 Zarządzanie jakością. Wytyczne dotyczące zarządzania konfiguracją	ISO 9000:2000 Systemy zarządzania jakością - podstawy i definicje
ISO 10011 Audit ISO	ISO 9000-1 Wybór modelu ISO 9001
ISO 10012:1998 Wymagania dotyczące zapewniania jakości wyposażenia pomiarowego	ISO 9000-2 Zastosowanie modelu ISO 9002
ISO 10013:1998 Wytyczne opracowywania ksiąg jakości	ISO 9000-3 Wytyczne do stosowania ISO 9001 w produkcji oprogramowania
ISO 10014:1998 TR Wytyczne do zarządzania kosztami jakości	ISO 9000-4 Niezawodność
ISO 10015:1999 Wytyczne do szkolenia	ISO 9001:1994, System jakości - Model zapewnienia jakości w projektowaniu, pracach rozwojowych, produkcji, instalowaniu i serwisie.
ISO 12207:1995 Software life cycle processes	ISO 9001:2000 Systemy zarządzania jakością –wymagania
ISO 14598:1999 „Proces oceny jakości oprogramowania”	ISO 9002:1994 Systemy jakości - Model zapewnienia jakości w produkcji, instalowaniu i serwisie.
ISO 15288/CD2 :2000 Standard on System Life Cycle Processes.	ISO 9003:1994 System jakości - Model zapewnienia jakości w kontroli i badaniach końcowych.
ISO 15504 TR SPICE	ISO 9004:2000 Systemy zarządzania jakością- Wytyczne dla prowadzenia doskonalenia
ISO 16949:1999 TS Systemy jakości - Dostawcy przemysłu samochodowego - szczególne wymagania dla stosowania normy ISO 9001: 1994 w przemyśle samochodowym	ISO 9126 cz1 - 4 Charakterystyki jakości oprogramowania

ISO/IEC 13335 TR .  
ISO/IEC 15846:1998 TR Information  
technology - Software life cycle  
processes - Configuration management  
ISO/IEC 16326 TR : Software  
Engineering - Guide for the application  
of ISO/IEC 12207 to project  
management.  
ISO/IEC 17799:2000 Praktyczne  
zasady zarządzania bezpieczeństwem  
informacji  
ISO/IEC 2: 1996 Standardization and  
related activities -- General vocabulary  
ISO/IEC 2382: Terminologia  
NATO AQAP 110  
NATO AQAP 120

NATO AQAP 150  
NATO AQAP 159  
NATO AQAP 169  
NATO AQAP 170  
NATO STANAG 4107 Standardization  
Agreement (luty 199)  
PN-71 T- 01016 Przetwarzanie danych  
i komputery- Podstawowe nazwy i  
określenia  
QS9000  
SQUARE Software Product  
Quality Requirement and Evaluation  
The TickIT Guide: A guide to Software  
Quality System Construction and  
Certification using ISO 9001:1994  
TL 9000:1999 QUEST

dr Andrzej Niemiec  
PRIM sp. z o.o.  
Ul Skłodowskiej 43/3  
50-369 Wrocław  
[ani@prim.com.pl](mailto:ani@prim.com.pl)  
[www.prim.com.pl](http://www.prim.com.pl)





# SYSTEM ZARZĄDZANIA JAKOŚCIĄ W PROJEKTACH PRODUKCJI OPROGRAMOWANIA W COMPUTERLAND S.A.

Lilianna WIERZCHOŃ

**Streszczenie:** W artykule przedstawiono System Zarządzania Jakością w projektach produkcji, kastomizacji, utrzymania i rozwoju oprogramowania aplikacyjnego w ComputerLand S.A.

ComputerLand świadczy zaawansowane usługi informatyczne dla klientów korporacyjnych i instytucjonalnych. Obsługuje pięć sektorów gospodarki: bankowo-finansowy, opieki zdrowotnej, przemysłowy, publiczny i telekomunikacyjny. Dzięki specjalizacji sektorowej posiada głęboką znajomość rynku, na którym działa klient. Nowoczesne rozwiązania informatyczne, oferowane przez firmę, generują zyski i przynoszą oszczędności. Wykształcona kadra i doświadczenie, zdobyte przy realizacji złożonych projektów, gwarantują najwyższy poziom usług.

## 1. Normalizacja systemów zarządzania jakością

Celem każdej firmy jest uzyskanie dobrej pozycji na rynku, a co za tym idzie zdobycie przewagi nad konkurencją. Wprowadzanie i stosowanie coraz nowszych narzędzi i technologii samo przez się nie zapewnia poprawy jakości wyrobów.

Tymczasem konsumenci oczekują produktów o coraz wyższej jakości. Również oprogramowanie, jak każdy wyrób rynkowy, podlega wnikliwej ocenie jakościowej klientów.

Dążenie do zapewnienia jakości wszelkiego rodzaju wyrobów i usług wymaga ujednoczenia terminologii w zakresie jakości i standaryzacji towarzyszących ich wytwarzaniu procesów. Standardy jakości dotyczą również oprogramowania.

Międzynarodowe organizacje, tj. **ISO** (the International Organization for Standarization), **IEC** (the International Electrotechnical Commission) oraz **IEEE** (Institute of Electrical and Electronic Engineers), przyjęły ustalenia w formie zalecanych norm jakościowych.

Modele zapewnienia jakości, przedstawione w trzech niżej wymienionych normach, reprezentują trzy różne zestawy wymagań dotyczących systemu jakości, odpowiednie do wykazania przez dostawcę jego zdolności oraz do oceny tej zdolności przez stronę zewnętrzną. Wymagania dotyczące systemu jakości zawarte w n/w normach nie zastępują wymagań technicznych wyrobu, a jedynie je uzupełniają. Mają one charakter ogólny i nie odnoszą się do określonej dziedziny czy branży gospodarki:

**ISO 9001** – stosowana, gdy dostawca ma zapewnić zgodność z ustalonymi wymaganiami podczas projektowania, prac rozwojowych, produkcji, instalowania i serwisu.

**ISO 9002** – stosowana, gdy dostawca ma zapewnić zgodność z ustalonymi wymaganiami podczas produkcji, instalowania i serwisu.

**ISO 9003** – stosowana, gdy dostawca ma zapewnić zgodność z ustalonymi wymaganiami jedynie podczas kontroli i badań końcowych.

## 1.1 Norma ISO 9001 – system zarządzania jakością

Norma **ISO 9001** określa zasady zarządzania, mające na celu zapewnienie zdolności do powtarzalnego dostarczenia klientowi wyrobu lub usługi na czas, przy spełnieniu określonych przez klienta wymogów. W 2000 roku norma ta została uaktualniona w ten sposób, aby przedsiębiorstwa mogły osiągać lepsze wyniki działalności skupiając uwagę na głównych procesach firmy.

Obecnie norma bierze pod uwagę wcześniejsze doświadczenia w stosowaniu norm systemów zarządzania, a także obejmuje nowe wymagania takie jak:

- ciągłe doskonalenie się przedsiębiorstwa,
- zwiększony nacisk na rolę najwyższego kierownictwa,
- uwzględnianie wymagań prawnych,
- ustalanie namacalnych celów,
- monitorowanie informacji dotyczących zadowolenia klienta,
- zwiększony nacisk na dostępność zasobów,
- określenie skuteczności szkoleń,
- uwzględnianie badań procesów oraz skuteczności działania,
- analizowanie danych systemu zarządzania.

Norma ISO 9001:2000 określa wymagania dotyczące systemu zarządzania jakością. Ma zastosowanie tam, gdzie wymagane jest od organizacji wykazanie zdolności do ciągłego dostarczania wyrobów spełniających wymagania klienta oraz spełniania wymagań prawnych. Norma ma na celu zapewnienie zadowolenia klienta poprzez skuteczne zastosowanie systemu jakości, w tym procesów pozwalających na ciągłe doskonalenie tego systemu i zapewnienie zgodności wyrobu z określonymi względem niego wymaganiami.

ISO 9001:2000 składa się z pięciu elementów, które są podstawą przy budowaniu każdego procesu:

### 1. System zarządzania jakością

element opisujący potrzebną dokumentację, dotyczące jej wymagania oraz sposób nadzorowania;

### 2. Odpowiedzialność kierownictwa

rola najwyższego kierownictwa przedsiębiorstwa, np. wymóg zapewnienia przez kierownictwo komunikowania oczekiwań klientów i przepisów prawnych na wszystkich szczeblach organizacyjnych, wymóg określenia polityki i celów firmy oraz zastosowania ich w całym przedsiębiorstwie, a także obowiązek prowadzenia okresowych przeglądów systemu zarządzania;



### 3. Zarządzanie zasobami

element wymagający zapewnienia przez organizację dostępu do zasobów ludzkich (jak i wszelkich innych) w celu właściwej realizacji procesów. Jednocześnie wymaga pełnego wykwalifikowania pracowników do wykonania powierzonych im zadań, zapewnienia zasobów fizycznych oraz odpowiedniego środowiska pracy;

### 4. Realizacja wyrobu

obejmuje wymagania dotyczące procesów niezbędnych do wytworzenia wyrobu lub dostarczenia usługi (według normy proces jest czynnością przetwarzania elementów wejściowych na wyjściowe);

### 5. Pomiary, analiza i doskonalenie

dotyczy badań, pozwalających na monitorowanie systemu i uzyskanie informacji o stopniu zadowolenia klienta, o skuteczności systemu, procesów i wyrobów, co pomoże określić środki konieczne do podjęcia w celu doskonalenia firmy.

## 1.2 ISO 9000-3 – wytyczne do stosowania normy ISO 9001 w produkcji oprogramowania

Norma ISO 9000-3 zawiera wytyczne do stosowania norm jakościowych ISO podczas tworzenia i obsługi oprogramowania. Norma ta porządkuje terminologię z zakresu produkcji oprogramowania.

**Oprogramowanie** definiuje jako „efekt pracy ludzkiego umysłu, obejmujący programy, procedury, zasady i wszelkie dokumenty związane, dotyczące obsługi systemu przetwarzania danych”. Pierwszy człon definicji podkreśla twórczy charakter tworzenia oprogramowania i uzasadnia zaliczenie programów komputerowych do utworów będących przedmiotem prawa autorskiego. Oprogramowanie, zgodnie z definicją, obejmuje całość dokumentacji tworzonej w kolejnych fazach cyklu życia.

**Wyrób programowy** jest to „kompletny zestaw programów komputerowych, procedur i dokumentów związanych oraz danych, dostarczany użytkownikowi”. Ta definicja każe traktować efekt pracy analityków, projektantów i programistów jako produkt rynkowy. Zasady gospodarki rynkowej obejmują z jednej strony ochronę praw producenta, a drugiej kładą nacisk na spełnienie oczekiwań nabywcy. Dokumentacja przeznaczona dla odbiorcy stanowi integralną część wyrobu.

Oprogramowanie coraz częściej stanowi kompleksowy produkt, złożony z wielu elementów, które dają się łączyć, integrować i odzyskiwać.

**Jednostka programowa** określona jest jako „dowolna część wyrobu programowego identyfikowalna podczas pośredniego lub końcowego etapu opracowywania”. Wyodrębnienie jednostki programowej wynika z dążenia do modularyzacji produktu, która ułatwia projektowanie i testowanie, a w przyszłości modyfikowanie, jak też odzyskiwanie na użytek innych aplikacji.

Wymaga się, aby każda wprowadzona jednostka projektu miała nazwę, przeznaczenie, funkcję oraz inne atrybuty. Węższe znaczenie ma pojęcie jednostki oprogramowania, którą jest oddzielnie kompilowany fragment kodu.

W miarę rozwoju kolejnych wersji wyrobu jednostka programowa staje się obiektem **konfiguracji**.

Zgodnie z zasadami inżynierii oprogramowania produkcja wyrobu programowego obejmuje czynności pogrupowane w fazy, które składają się na cykl rozwojowy. Norma ISO 9000-3 wprowadza termin **opracowywanie**, które stanowi „ogół działań prowadzących do stworzenia wyrobu programowego”. Wyodrębniony fragment prac nad wyrobem programowym nazywa się **etapem** lub **fazą**.

Przygotowanie kontraktu, formułowanie wymagań, opracowanie wyrobu programowego, a następnie jego wdrożenie i eksploatacja, zaś później jego pielęgnacja i utrzymanie oraz dalszy rozwój wymagają wykonania wielu czynności, które począwszy od lat 80 dwudziestego wieku, dzięki ukształtowaniu się nowej dziedziny inżynierii – inżynierii oprogramowania - stopniowo podlegają standaryzacji.

Wymienione działania tworzą wspólnie proces produkcji i rozwoju oprogramowania, który nadaje się do stosowania w odniesieniu do wszelkich przedsięwzięć programistycznych, niezależnie od wielkości końcowego produktu. Proces ten definiuje inżynierskie podejście do całości działań i dzieli się na wiele procesów składowych.

## 2. System Jakości ComputerLand S.A.

**Misja ComputerLand S.A.:** „Być niekwestionowanym strategicznym partnerem naszych Klientów dzięki wprowadzaniu przez najlepszych specjalistów niezawodnych systemów informacyjnych, przynoszących Klientom korzyści biznesowe i organizacyjne”.

**Polityka Jakości** określa jak w ComputerLand S.A. rozumiemy jakość i w jaki sposób zapewniamy jakość naszym Klientom...

„...Aby zostali z nami przez długie lata w tym celu:

- Staramy się znać i rozumieć potrzeby naszych Klientów. Dlatego tak ważne są dla nas rozmowy z naszymi Klientami, ich życzenia i opinie.
- Chcemy być częścią sukcesu naszych Klientów. Dlatego stale wzbogacamy naszą ofertę.
- Staramy się nie zawieść zaufania naszych Klientów. Dlatego wypełniamy zobowiązania, które podejmujemy.
- Chcemy być w każdej chwili gotowi skutecznie pomagać naszym Klientom. Dlatego stale doskonalimy siebie i organizację naszej firmy oraz dbamy o dobrą współpracę z naszymi Partnerami”.

Polityka Jakości opisuje podejście naszej organizacji do jakości, którą chcemy osiągnąć, doskonalić i utrzymywać stale na wysokim poziomie. W tym celu postanowiliśmy stworzyć, nierozzerwalnie związane z systemem zarządzania naszej firmy, system jakości.

System Jakości został opracowany i wdrożony w 1996 roku zgodnie z wymaganiami normy ISO 9001 (model dla projektowania, instalacji i serwisu) i obejmował swoim zakresem budowę i integrację sieci komputerowych w oparciu o urządzenia uznanych światowych producentów, z projektowaniem, instalowaniem, dostawą i serwisem włącznie.

W ciągu kolejnych lat działalność biznesowa ComputerLand S.A. ewoluowała w kierunku dostarczania naszym klientom kompleksowych rozwiązań informatycznych opartych na oprogramowaniu własnym oraz kastomizowanym oprogramowaniu uznanych światowych producentów oprogramowania aplikacyjnego. Dążąc do dostarczania naszym klientom oprogramowania o najwyższej jakości, zgodnie z normą ISO 9000-3 zawierającą wytyczne do stosowania normy ISO 9001 podczas opracowywania, dostarczania i obsługiwanie oprogramowania, opracowaliśmy i wdrożyliśmy uzupełnienie Systemu Jakości uzyskując w maju 2001 rozszerzenie dotychczasowego zakresu certyfikatu ISO 9001 w obszarze produkcji, kastomizacji, wdrażania i serwisu własnego oprogramowania aplikacyjnego oraz oprogramowania Oracle Applications.

System Jakości definiuje filozofię i cele firmy w Polityce Jakości, organizację opisaną w Księdze Jakości oraz metody działań mających wpływ na jakość oferowanych rozwiązań ujęte w formie polityk, wzorów, procedur, instrukcji i formularzy, opisane w Księdze Dokumentów Systemu Jakości. System Jakości jest wspierany przez rozwiązania w Lotus Notes.

System Jakości, ściśle zintegrowany z systemem finansowym i kontrollingowym jest narzędziem dla kierownictwa firmy i wszystkich pracowników, które właściwie wykorzystane pozwala na osiągnięcie wymaganej jakości i rentowności oraz obniżenie kosztów naprawiania popełnionych błędów. Należy jednak podkreślić, że sam system nie zapewnia nam jakości. Dlatego ważną rzeczą jest rozumienie celów jakościowych przez każdego pracownika firmy i ciągła praca nad udoskonalaniem siebie i metod swojej pracy.

Ze względu na ochronę know-how firmy, wszystkie dokumenty Systemu Jakości oznaczone są klauzulą „*Dokument zastrzeżony ComputerLand S.A.*”.

Jednostką certyfikującą ComputerLand jest światowy lider wśród firm certyfikujących **British Standards Institution**.

Rozszerzenie przyznanego ComputerLand S.A. certyfikatu ISO 9001 o zakres związany z produkcją oprogramowania zbiegło się w czasie z ogłoszeniem przez tygodnik COMPUTERWORLD rocznego raportu TOP 200 Polski Rynek Informatyczny i Telekomunikacyjny 2000 i przyznaniem ComputerLand S.A. pierwszego miejsca w rankingu największych producentów oprogramowania w roku 2000.

Co roku przeprowadzane są dwa audyty zewnętrzne – prowadzone przez firmę BSI, których celem jest stwierdzenie zgodności działań prowadzonych w ramach realizacji projektów z wymaganiami normy ISO 9001.





#### *Licence Scope*

*The procurement and integration of computer networks with associated products including sales, design, delivery, installation and service. In addition, the design, development, customisation, supply and maintenance of ComputerLand's own application software and Oracle Applications software.*

### **3. Rozwój i dokumentowanie kompetencji pracowników**

Kompetencje i umiejętności pracowników ComputerLand S.A. są regularnie rozwijane i doskonalone dzięki planowej realizacji programów szkoleniowych. Programy te przewidują dla poszczególnych grup stanowisk szereg regularnych szkoleń wewnętrznych i zewnętrznych.

Potrzeby i plany szkoleniowe są projektowane dla poszczególnych pracowników w bazie Lotus Notes ComputerLand Training, zgodnie z dokumentem Systemu Jakości „Polityka szkoleniowa”. Również w tej bazie odnotowywane są odbyte szkolenia, zdobyte umiejętności i kompetencje oraz zdobyte certyfikaty i dyplomy. Dzięki mechanizmom wbudowanym w bazę, w każdej chwili można ocenić poziom kompetencji poszczególnych osób, wyszukiwać osoby o zadanych kompetencjach itd., co jest bardzo pomocne przy budowaniu zespołów projektowych.

### **4. Metodyka zarządzania projektami**

Produkcja, wdrożenie i pielęgnacja oprogramowania to procesy, które są realizowane w ComputerLand S.A. w systematyczny i uporządkowany sposób. Stosowane w firmie modele cyklu życia projektu:

- wprowadzają podział na fazy życia oprogramowania,
- definiują, jakie dane będą podstawą działań projektowych danej fazy, jak będzie odbywać się ich przetwarzanie, jakie narzędzia lub metodyki powinny być zastosowane,

- określają czynności i zadania oraz ustalają kolejność ich realizacji oraz kto jest odpowiedzialny i kto realizuje poszczególne działania, jakie jest wsparcie organizacji (potrzebne zasoby i środki wytwórcze),
- określają, co będzie produktem każdej fazy i jakie będą kryteria jego odbioru i oceny,
- pozwalają uporządkować przebieg prac,
- ułatwiają planowanie zadań oraz monitorowanie przebiegu ich realizacji odwołując się do sprawdzonych schematów postępowania.

Od wybrania modelu cyklu życia oprogramowania zależy sposób realizacji projektu.

Szczegółowy opis cyklu życia oprogramowania uzupełniony o:

- specyficzne dla poszczególnych faz formularze projektowe,
- techniki analizy, projektowania i implementacji,
- metody weryfikacji i testowania
- procedury i formularze wspomagające zarządzanie projektem

tworzy w ComputerLand S.A. firmową metodykę zarządzania projektem.

Ze względu na różnorodność stosowanych w ComputerLand S.A.:

- modeli cyklu życia oprogramowania,
- metodyk projektowania systemów informatycznych,
- narzędzi wspierających procesy produkcyjne,
- języków i systemów programowania,
- platform bazodanowych i sprzętowo-narzędziowych

dokumenty Systemu Jakości przygotowane do wspierania i monitorowania procesów wytwórczych oprogramowania mają charakter polityk i wzorów, które należy dostosować do specyfiki każdego projektu i wymagań klienta.

W dziedzinie zarządzania projektami wspieranej przez dokumenty Systemu Jakości wyróżniamy dwa typy procesów: procesy kierownicze i procesy produkcyjne.

Procesy kierownicze są odpowiedzialne za organizowanie i monitorowanie prac w czasie całego cyklu życia projektu począwszy od zaplanowania przedsięwzięcia do fazy pielęgnacji.

Zaliczamy do nich przykładowo:

- planowanie projektu,
- szacowanie kosztów projektu,
- tworzenie harmonogramu prac projektowych,
- organizowanie zespołów projektowych,
- przygotowanie i przegląd umów,
- zarządzanie wymaganiami,
- zarządzanie jakością,
- zarządzanie czasem,
- zarządzanie budżetem,
- zarządzanie ryzykiem,
- zarządzanie zmianami,
- zarządzanie komunikacją,
- zarządzanie kontaktami z podwykonawcami i dostawcami.

Procesy produkcyjne związane, z tworzeniem (kustomizacja) oprogramowania, rozpoczynają się w fazie strategicznej projektu, gdzie wybiera się i planuje sposób realizacji procesu produkcyjnego, a kończą się w fazie pielęgnacji produktu.

Procesy produkcyjne realizowane są zgodnie z wybranym modelem cyklu życia oprogramowania.

Przykładowe procesy produkcyjne to:

- analizowanie wymagań użytkownika,
- określanie wymagań względem systemu,
- modelowanie systemu,
- projektowanie architektury systemu,
- projektowanie szczegółowe,
- projektowanie interfejsu użytkownika,
- kodowanie, integracja i testowanie oprogramowania,
- integracja i testowanie systemu,
- pielęgnacja systemu i oprogramowania.

Na zdefiniowane główne fazy produkcji nakładają się procesy związane z przygotowaniem i przeprowadzeniem wdrożeń:

- przygotowanie i przeprowadzenie szkoleń,
- opracowanie dokumentacji użytkownika,
- przygotowanie i przeprowadzenie konwersji danych z systemów klienta,
- przygotowanie i przeprowadzenie migracji danych z poprzedniej wersji oprogramowania,
- przygotowanie i instalacja interfejsów do systemów zewnętrznych,
- instalacja i uruchomienie systemu.

## **5. Planowanie realizacji projektu**

W ComputerLand S.A. wyróżniamy projekty:

- realizowane przez zespoły projektowe w oparciu o umowę z klientem,
- projekty wewnętrzne realizowane przez centra kompetencyjne produktowe, związane z wytwarzaniem i doskonaleniem oprogramowania aplikacyjnego.

Planowanie realizacji każdego projektu odbywa się wg procedury Systemu Jakości „Planowanie realizacji projektu”. Zawarte są w niej wytyczne do zaplanowania działań związanych z cyklem życia oprogramowania oraz wytyczne do opracowania następujących obowiązkowych dokumentów: „Budżet Projektu”, „Plan Projektu”, „Plan jakości”, „Plan zarządzania konfiguracją”, procedury zarządzania projektem.

### **5.1 Planowanie budżetu**

Budżet każdego projektu jest opracowywany zgodnie ze wzorem formatki budżetowej zaakceptowanym przez pion księgowo- finansowy.



Opracowując budżet realizacji danego projektu, kierownik projektu przy współpracy handlowca prowadzącego wprowadza:

- planowane w ramach umowy przychody (zgodnie z planowanym harmonogramem fakturowania),
- planowane koszty pionów handlowych (zgodnie z harmonogramem pojawiania się tych kosztów),
- planowane zlecenia wewnętrzne – wartości i harmonogram zamieszczane są zgodnie z rozpoczęciem prac w ramach realizacji poszczególnych zleceń przez poszczególne centra kompetencyjne i centra kompetencyjne technologiczne.

Wartości wprowadzanych zleceń wewnętrznych są, przed wprowadzeniem ich do budżetu, uzgadniane z szefami poszczególnych działów produkcyjnych.

Kierownik projektu jest również odpowiedzialny za wprowadzenie do formatki budżetu planowanych kosztów poszczególnych działów produkcyjnych w ramach zleceń wewnętrznych.

Koszty te wprowadza po uzyskaniu od szefów poszczególnych działów produkcyjnych oszacowania wartości odpowiednich dla rodzajów ponoszonych kosztów (towarów, usług zewnętrznych, usług wewnętrznych i innych).

Przy oszacowaniu kosztów usług wewnętrznych, wynikających z czasowego zaangażowania pracowników działu, do realizacji danej umowy szef działu określa ilość osobodni pracownika, z określonego poziomu kompetencji, koniecznych do realizacji tego zadania (lub sumę iloczynów osobodni i poziomów kompetencji).

Tak opracowany budżet realizacji projektu załączany jest do bazy „Przegląd Umowy”.

Jeżeli w ramach realizacji projektu zaistnieją zdarzenia znacząco wpływające na stan realizacji budżetu danego projektu, kierownik projektu przygotowuje w bazie „Przegląd Umowy” kolejną wersję budżetu, opisując w odpowiednim polu uzasadnienie tego kroku.

## 5.2 Przegląd umowy

Zgodnie z procedurą Systemu Jakości „Rozszerzony przegląd umowy”, każda umowa lub aneks do umowy, wraz z załącznikami, przed podpisaniem musi być zarejestrowana przez handlowca odpowiedzialnego za klienta na elektronicznym formularzu w bazie „Przegląd umowy” i podlega przeglądowi przez wszystkie osoby kluczowe potencjalnie odpowiedzialne za realizację umowy, którego celem jest:

- analiza treści umowy oraz zaprojektowanego budżetu projektu pod kątem rentowności,
- analiza treści umowy pod kątem zawarcia w niej wymaganych zapisów formalnych i merytorycznych, między innymi, takich jak:
  - określenie zakresu umowy,
  - precyzyjne zdefiniowanie i udokumentowanie wymagań,
  - sprawdzenie zabezpieczenia praw własności i ochrony know-how,

- rozwiązanie wszelkich różnic wymagań pomiędzy zapytaniem ofertowym a ofertą,
- pełne zdefiniowanie produktów i usług,
- ustalenie harmonogramu prac i dostaw oraz kryteriów odbioru,
- zarządzanie zmianami wymagań,
- ustalenie zasad serwisu,
- zarządzanie reklamacjami,
- zdefiniowanie obowiązków stron w realizacji projektu,
- ustalenie zasad współpracy,
- ocena ryzyka realizacji projektu,
- zbadanie zdolności do spełnienia wymagań i możliwości realizacyjnych ComputerLand S.A.,
- sprawdzenie odpowiedzialności ComputerLand S.A. w odniesieniu do prac wykonywanych przez podwykonawców,
- weryfikacja podwykonawców,
- sprawdzenie klienta pod względem możliwości spełnienia zobowiązań wynikających z umowy,

Po uzyskaniu akceptacji szefów wszystkich działów, które będą zaangażowane w realizację umowy, przeprowadzeniu analizy ryzyka realizacji danej umowy kierownik projektu ostatecznie akceptuje i zobowiązuje się do realizacji umowy zgodnie z przyjętymi założeniami.

Elektroniczny obieg formularza przeglądu umowy, dokonywanie przeglądu poprzez udzielanie odpowiedzi na zadane w formularzu pytania, automatyczne ustawianie statusu przeglądu pozwalają na sprawne zarządzanie umową oraz podpisywanie umów przez Zarząd ComputerLand S.A. z pełną odpowiedzialnością.

### 5.3 Planowanie procesów związanych z cyklem życia oprogramowania

Większość modeli cyklu życia oprogramowania można uznać za pewne wersje lub rozszerzenia modelu klasycznego; w szczególności fazy takie jak:

- określanie wymagań,
- analiza,
- projektowanie,
- implementacja,
- testowanie,
- wdrażanie,
- pielęgnacja

pojawiają się w większości realizowanych w ComputerLand S.A. przedsięwzięć, zatem we wzorcowych dokumentach Systemu Jakości związanych z procesem produkcji i wdrożenia oprogramowania często występują odwołania do klasycznego modelu cyklu życia.

Wytyczne dla kierownika projektu (kierownika produktu – oprogramowania aplikacyjnego), do planowania działań i produktów

wynikających z cyklu życia oprogramowania, znajdują się w dokumencie Systemu Jakości „Planowanie i realizacja projektów obejmujących produkcję, kastomizację i wdrożenie oprogramowania”.

### 5.3.1 Analiza (określanie wymagań i modelowanie)

W dokumencie Systemu Jakości „Planowanie i realizacja fazy analizy (określanie wymagań i modelowanie) dla projektów obejmujących produkcję (kastomizację) oprogramowania” znajdują się podstawowe informacje z inżynierii oprogramowania mające charakter wytycznych dla analityków do planowania i realizowania procesów określania wymagań i modelowania obejmujące następujące procesy:

- przygotowanie procesu pozyskiwania wymagań,
- przeprowadzanie wywiadów i pozyskiwanie wymagań,
- dokumentowanie wymagań,
- analiza i konsolidacja wymagań,
- opracowanie dokumentu „Specyfikacja wymagań względem systemu”,
- opracowanie dokumentu „Kryteria akceptacji systemu”,
- modelowanie przy podejściu strukturalnym/obiektywnym i opracowanie modelu logicznego systemu,

### 5.3.2 Projektowanie

W dokumencie Systemu Jakości „Planowanie i realizacja fazy projektowania dla projektów obejmujących produkcję (kastomizację) oprogramowania” znajdują się podstawowe informacje z inżynierii oprogramowania mające charakter wytycznych dla projektantów do planowania i realizowania procesu projektowania.

Główne procesy opisane w tym dokumencie to:

#### 1. Projektowanie ogólne (projektowanie architektury):

- określanie standardów projektowych,
- podział systemu na podsystemy,
- wybór sposobu realizacji elementów modelu,
- określanie sposobu realizacji sterowania,
- określanie sposobu realizacji magazynów danych,
- organizacja dostępu do zasobów globalnych,
- opracowanie dokumentu „Projekt ogólny (architektury) systemu”,

#### 2. Projektowanie szczegółowe:

- uszczegółowianie wyników analizy przy podejściu strukturalnym /obiektywnym,
- projektowanie interfejsu użytkownika,
- projektowanie zarządzania danymi (pliki, bazy danych),
- projektowanie zarządzania pamięcią operacyjną,
- projektowanie zarządzania czasem procesora,



- optymalizacja projektu,
- dostosowanie projektu do ograniczeń i możliwości środowiska implementacji,
- określanie fizycznej struktury systemu,
- opracowanie dokumentu „Projekt szczegółowy systemu”,
- opracowanie innych dokumentów projektowych,

### 5.3.3 Implementacja (Konstrukcja)

W dokumencie Systemu Jakości „Planowanie i realizacja fazy implementacji (konstrukcji) dla projektów obejmujących produkcję (kustomizację) oprogramowania” znajdują się podstawowe informacje z inżynierii oprogramowania mające charakter wytycznych dla programistów do planowania i realizowania procesu implementacji, obejmujące następujące procesy:

- tworzenie i komentowanie kodu,
- zwiększanie niezawodności na etapie implementacji przez zastosowanie technik unikania błędów i tolerowania błędów,
- optymalizacja kodu,
- weryfikowanie kodu,
- przeprowadzanie i dokumentowanie testów jednostkowych,
- integracja oraz przeprowadzanie i dokumentowanie testów integracyjnych,
- opracowanie dokumentu „Dokumentacja techniczna systemu”.

W każdym centrum kompetencyjnym obowiązują standardy tworzenia i komentowania kodu źródłowego, ściśle uzależnione od wykorzystywanych narzędzi i języków programowania, obejmujące:

1. szablony i wytyczne do dokumentowania prac implementacyjnych,
2. konwencje nazewnictwa elementów kodu (procedur, metod, funkcji, parametrów, zmiennych, tablic, pól, itp.),
3. standard stylu programowania (wcięcia, bloki, puste linie, spacje, używanie małych i wielkich liter, sposób redakcji komentarza kodu)
4. standard dokumentowania kodu:
  - standard opisu metryki funkcji,
  - standard dokumentowania działania fragmentów kodu i opisywania zmian w kodzie,
  - historia zmian – zazwyczaj na końcu kodu (data utworzenia/zmiany, wersja, identyfikator programisty, krótki opis zmian),
5. standardowy sposób realizacji konstrukcji pojawiających się w modelu (projekcie),
6. zalecane metody zwiększania niezawodności tworzonego oprogramowania (unikanie błędów i tolerowanie błędów),
7. zalecane techniki optymalizacji kodu (algorytmów i struktur danych: kolejność przeszukiwań i obliczeń, składowanie atrybutów wywiedzionych),

8. wytyczne i zalecenia do stosowania bibliotek - środowisko implementacji może zawierać szereg gotowych elementów (biblioteki standardowe lub biblioteki komercyjne), które mogą ułatwić implementację.

W centrach kompetencyjnych, które przygotowują kolejne wersje lub rozwijają produkty, do standardów kodowania mogą być dołączone elementy projektowe jako:

9. standardy implementacji parametrów i słowników aplikacyjnych,

10. standardy tworzenia interfejsu,

11. standardy budowy interfejsów do innych systemów.

#### **5.3.4 Testowanie**

Obowiązującym w ComputerLand S.A. dokumentem Systemu Jakości precyzującym podejście do testowania jest procedura „Testowanie”.

Dokument ten odwołuje się do wzorcowych dokumentów Systemu Jakości - szczegółowych formularzy, planów, procedur, instrukcji, które należy dostosować do specyfiki projektu i umowy z klientem.

Opisano w nich podejście do każdego z czterech poziomów testowania:

##### **1. Testy jednostkowe**

Najniższy poziom eksperymentowania z kodem wykonywalnym danego produktu (programu, systemu). Testy jednostkowe stosujemy do konkretnych fragmentów kodu, w celu zweryfikowania ich funkcjonalności lub wewnętrznej struktury w stosunku do projektu szczegółowego.

##### **2. Testy integracyjne**

Powyżej poziomu testów jednostkowych realizowane są testy integracyjne. Mają one na celu przetestowanie struktury systemu i funkcjonalności integrowanych fragmentów systemu w stosunku do projektu architektury. Istota testów integracyjnych polega na stopniowym łączeniu przetestowanych jednostek kodu w większą całość, a następnie testowaniu ich jako grupy jednostek. Proces łącznia i testowania jest powtarzany aż do powstania całego systemu.

##### **3. Testy systemowe**

Testy systemowe dostarczają niezbędnego materiału do podjęcia decyzji o tym, czy wszystkie składniki systemu są ze sobą połączone i współpracują tworząc system odpowiadający specyfikacji wymagań względem systemu i specyfikacji wymagań użytkowych. Na tym poziomie testowania nie ma znaczenia, w jaki sposób połączono ze sobą poszczególne składniki systemu.

##### **4. Testy akceptacyjne**

Testy akceptacyjne wspierają walidację systemu, czyli proces oceny produktu końcowego przez klienta, ze względu na sprawdzenie, czy produkt końcowy spełnia wymagania postawione mu w specyfikacji wymagań użytkownika na początku procesu wytwarzania.

Testy akceptacyjne mają na celu zademonstrowanie klientowi, że system (program, aplikacja) działa, jest kompletny i może być użyty w środowisku docelowym.

Wśród standardowych dokumentów z zakresu testowania, najbardziej istotne to:

- formularz „Skrypt testowy”, instrukcje: „Projektowanie skryptu testowego”, „Wykonanie skryptu testowego”,
- formularz „Scenariusz testowy”, instrukcje: „Projektowanie scenariusza testowego”, „Wykonanie scenariusza testowego”,
- formularz „Zgłoszenie problemu testowego” i instrukcja „Zgłaszanie problemów testowych”,
- wzór „Plan testów jednostkowych” oraz procedury: „Projektowanie testów jednostkowych”, „Przeprowadzenie testów jednostkowych” i formularz „Raport z testów jednostkowych”,
- wzór „Plan testów integracyjnych” oraz procedury: „Projektowanie testów integracyjnych”, „Przeprowadzenie testów integracyjnych” i formularz „Raport z testów integracyjnych”,
- wzór „Plan testów systemowych” oraz procedury: „Projektowanie testów systemowych”, „Przeprowadzenie testów systemowych” i formularz „Raport z testów systemowych”,
- wzór „Plan testów akceptacyjnych” oraz procedury: „Projektowanie testów akceptacyjnych”, „Przeprowadzenie testów akceptacyjnych” i formularz „Raport z testów akceptacyjnych”.

W każdym ze wspomnianych wyżej planów testów należy określić:

- cel testów,
- zakres testów,
- listę przypadków testowych,
- przyjęte strategie testowania,
- narzędzia i techniki testowe,
- cykle testów,
- klasyfikację błędów i problemów testowych,
- harmonogram przygotowania i przeprowadzenia testów,
- organizację zespołu testowego,
- organizację środowisk testowych,
- warunek rozpoczęcia testów,
- przebieg testów,
- poprawianie błędów i rozwiązywanie problemów testowych,
- kryteria zaliczenia testów,
- dokumentowanie testów,
- punkty kontrolne.

Na podstawie w/w dokumentów, osoby odpowiedzialne za przygotowanie i przeprowadzanie poszczególnych rodzajów testów opracowują plany testów i procedury testowania do zastosowania w danym projekcie.



## 5.4 Plan projektu

„Plan projektu” opracowywany jest przez kierownika projektu na podstawie dokumentu Systemu Jakości „Plan projektu”, przygotowanego dla modelu klasycznego cyklu życia projektu i mającego charakter wzoru, który należy dostosować do specyfiki projektu uwzględniając zapisy kontraktowe oraz przyjętą metodykę i model cyklu życia oprogramowania.

„Plan projektu” jest dokumentem konstytuującym:

- cele projektu,
- jego organizację,
- techniczne podejście do procesu wytwórczego,
- zastosowaną technologię,
- sposób zarządzania przedsięwzięciem,
- proces decyzyjny.

Dokument musi uzyskać akceptację kluczowych osób w projekcie ze strony ComputerLand S.A. i klienta (nawet wtedy, gdy jest to klient wewnętrzny). Powinien być zapoznany z nim każdy członek zespołu projektowego klienta i ComputerLand S.A.. „Plan projektu” jest uaktualniany w miarę rozwoju przedsięwzięcia w sposób wyprzedzający wykonanie czynności przypisanych kolejnym fazom produkcji oprogramowania w postaci dokumentu „Plan etapu”.

W „Planie projektu” ujęte są następujące zagadnienia:

### 1. Geneza projektu:

- informacje o umowie z klientem lub decyzji Zarządu ComputerLand S.A. będącej podstawą uruchomienia projektu,
- odwołanie do dokumentu z wymaganiami,
- ramowy harmonogram realizacji projektu,
- definicje pojęć specyficznych dla projektu,
- kontraktowe zobowiązania ComputerLand S.A.,
- kontraktowe zobowiązania klienta.

### 2. Cele projektu z punktu widzenia klienta.

3. Zakres i kontekst projektu - szczegółowe określenie zakresu przedsięwzięcia z opisaniem kontekstu systemu, czyli systemów zewnętrznych, z którymi współpracować będzie tworzony system.

4. Wyszczególnienie specyficznych ograniczeń – opis, czego projekt nie obejmuje (co nie wchodzi w zakres projektu).

5. Podejście techniczne - opis zastosowanego podejścia technicznego do tworzenia oprogramowania (np. budowa systemu dedykowanego, zastosowanie gotowych aplikacji (jakich) i ich integracja, dopasowanie gotowej aplikacji (jakiej) do potrzeb klienta), odniesienie do wybranej platformy sprzętowo-programowej:

- architektura systemu – wstępny opis architektury systemu z ujęciem kontaktu z otoczeniem zewnętrznym systemu, w dwóch aspektach: sprzętowo-komunikacyjnym i oprogramowania aplikacyjnego (podział na podsystemy i moduły),

- cykl życia projektu - opis zastosowanego modelu cyklu życia projektu i wynikającej z niego struktury procesu związanego z realizacją projektu (podział na etapy i fazy) z uwzględnieniem działań po stronie ComputerLand S.A. i klienta,
  - specyfikacja oraz opis produktów przejściowych i finalnych każdej fazy i etapu projektu,
  - opis podejścia do pielęgnowania wymagań,
  - zarządzanie konfiguracją projektu - opis podejścia do identyfikowania poszczególnych składowych oprogramowania i ich kolejnych wersji oraz sposobu implementacji, kontroli, zapisu i dokumentowania nanoszonych zmian – do rozwinięcia i uszczegółowienia w „Planie zarządzania konfiguracją projektu”,
  - narzędzia i środowiska projektowe – opis narzędzi, dedykowanych i standardowych oraz ich wersji, wspomagających proces produkcji oraz proces zarządzania projektem, charakterystyka środowisk projektowych: projektowo-programistycznego, szkoleniowego, testowego, demo, do konwersji, migracji, produkcyjnego.
6. Bazowy harmonogram projektu – sporządzony w oparciu o harmonogram ramowy realizacji systemu znajdujący się w umowie z klientem, po uwzględnieniu działań i procesów wynikających z przyjętej metodyki produkcji oprogramowania; bazowy harmonogram zawiera następujące elementy:
- podział projektu na poszczególne zadania w ramach etapów/faz z uwzględnieniem relacji między zadaniami,
  - wyszczególnienie zadań krytycznych,
  - terminy dostarczenia zasadniczych dla projektu produktów i usług,
  - wykaz punktów kontrolnych związanych z weryfikacją, walidacją i testowaniem oprogramowania,
  - wyszczególnienie kamieni milowych,
  - terminy czynności niezbędnych do przygotowania produktów i zrealizowania usług,
  - terminy podjęcia istotnych dla projektu decyzji,
  - odpowiedzialność i obciążenie poszczególnych zasobów.
7. Akceptacja i odbiory – opis kryteriów akceptacji produktów i procedur ich odbioru.
8. Organizacja projektu:
- opis struktury organizacyjnej projektu,
  - lista osób kluczowych w projekcie z określeniem ról i relacji między osobami kluczowymi po stronie klienta i ComputerLand S.A.,
  - opis kluczowych ról projektowych i przypisanych do nich zakresów odpowiedzialności,
  - lista i zakres odpowiedzialności podwykonawców,

- opis standardów pracy obowiązujących zespół projektowy w zakresie: analizy, projektowania, programowania, testowania, dokumentowania, szkoleń, wdrożeń.
9. Metody zarządzania projektem – opis dostosowanych do specyfiki projektu i umowy z klientem wzorcowych procedur i formularzy Systemu Jakości ComputerLand S.A. wspomagających realizację procesów kierowniczych w projekcie:
    - zarządzanie zakresem,
    - zarządzanie czasem,
    - zarządzanie ryzykiem,
    - zarządzanie budżetem,
    - zarządzanie zespołem projektowym,
    - zarządzanie kontraktem,
    - zarządzanie dostawcami i podwykonawcami,
    - zarządzanie problemami,
    - zarządzanie błędami i brakami,
    - zarządzanie korespondencją,
    - zarządzanie dokumentacją,
    - zarządzanie biblioteką i archiwum projektu,
    - dokonywanie systematycznych przeglądów projektu.
  10. Zarządzanie jakością:
    - opis dostosowanych do specyfiki projektu i umowy z klientem wzorcowych procedur i formularzy Systemu Jakości ComputerLand S.A. z zakresu zarządzania jakością,
    - opis zawartości „Planu jakości”,
    - opis podejścia do kontroli jakości produktów pośrednich i końcowych w cyklu życia projektu,
    - odwołanie do procedur Systemu Jakości ComputerLand S.A. związanych z organizacją i przeprowadzaniem audytów wewnętrznych projektu oraz działań związanych z weryfikacją, walidacją i testowaniem.
  11. Zarządzanie zmianami – opis dostosowanych do specyfiki projektu i umowy z klientem wzorcowych procedur i formularzy Systemu Jakości ComputerLand S.A. z zakresu zarządzania zmianami zapewniających, że wszystkie zmiany i modyfikacje projektu przed ich wdrożeniem będą określone, udokumentowane, przeglądane i zatwierdzone przez autoryzowany personel.
  12. Zasady raportowania – opis dostosowanych do specyfiki projektu i umowy z klientem wzorcowych procedur i formularzy Systemu Jakości ComputerLand S.A. z zakresu raportowania ustalających ścieżki, sposoby i okresy raportowania.
  13. Zasady komunikacji w projekcie – opis dostosowanych do specyfiki projektu i umowy z klientem wzorcowych procedur i formularzy Systemu Jakości ComputerLand S.A. z zakresu komunikacji w projekcie ustalających rodzaje spotkań i ich terminy.



14. Informacje podstawowe o projekcie – adres siedziby projektu i kontakty do członków zespołu projektowego.
15. Załączniki – harmonogram bazowy projektu i procedury z zakresu zarządzania projektem i realizacji procesów produkcyjnych.

## 5.5 Plan jakości

Dokumentem związanym z „Planem projektu” jest „Plan jakości”, który precyzuje aspekty zapewnienia jakości w projekcie (produktach i procesach) oraz określa szczegółowo kryteria i procedury odbioru produktów i usług.

„Plan jakości” opracowuje kierownik projektu we współpracy z kontrolerem jakości w oparciu o wzorcowy dokument Systemu Jakości ComputerLand S.A. „Plan jakości”, który należy dostosować do specyfiki projektu uwzględniając zapisy kontraktowe i przyjętą metodykę oraz model cyklu życia oprogramowania.

„Plan jakości” jest uaktualniany w miarę rozwoju przedsięwzięcia w sposób wyprzedzający wykonanie czynności przypisanych poszczególnym fazom produkcji oprogramowania w postaci dokumentu „Plan jakości etapu”.

W „Planie jakości” ujęte są następujące zagadnienia:

1. Zadania „Planu jakości” – opis zadań mających na celu zapewnienie jakości w procesie wytwórczym oprogramowania i usługach oraz w zarządzaniu projektem.
2. Cele jakościowe projektu – definicja mierzalnych celów jakościowych, wynikających ze strategii firmy klienta, szczegółów umowy z klientem i specyfikacji wymagań użytkowych.
3. Zapewnienie jakości w projekcie - opis zaplanowanych i systematycznie realizowanych działań, które są niezbędne do uzyskania i utrzymania odpowiedniego stopnia wiarygodności, że powstały w wyniku realizacji projektu system spełni założone cele jakościowe, a więc:
  - zdefiniowanie odpowiedzialności za jakość w projekcie,
  - zaplanowanie regularnych przeglądów realizacji działań opisanych w „Planie jakości”,
  - zaplanowanie audytów wewnętrznych,
  - ustalenie trybu zlecenia działań zapobiegawczo-korygujących,
  - ustalenie zasad dokumentowania i przechowywania zapisów z audytów i przeglądów jakości.
4. Kontrola jakości w projekcie – opis metod, technik i narzędzi kontrolnych stosowanych w projekcie:
  - badania, weryfikacja, zatwierdzanie - opis podejścia do przeprowadzania i dokumentowania testów, weryfikacji i walidacji produktów pośrednich i końcowych poszczególnych etapów/faz projektowych – do uszczegółowienia w „Planie weryfikacji, walidacji i testowania”,
  - kontrola i odbiór dostarczanych klientowi produktów i usług – opis dostosowanych do specyfiki projektu i umowy z klientem wzorcowych procedur, formularzy, raportów i protokołów Systemu Jakości

ComputerLand S.A. z zakresu kontroli i odbiorów oprogramowania, dokumentacji i usług,

- kontrola dostawców i podwykonawców – opis standardowych procedur Systemu Jakości ComputerLand S.A. w zakresie kontroli i weryfikacji podwykonawców i dostawców.
5. Ogólne kryteria jakości produktów i usług oraz metody ich weryfikacji, walidacji i odbioru – pełna specyfikacja produktów pośrednich i końcowych oraz usług z określeniem dla każdego z nich:
- ogólnych kryteriów jakości,
  - metod weryfikacji, walidacji i odbioru,
  - formy udokumentowania weryfikacji, walidacji i odbioru,
  - osób odpowiedzialnych, ze strony klienta i ComputerLand S.A., za wykonanie powyższych czynności.
6. Załączniki - „Plan weryfikacji, walidacji i testowania”, procedury, formularze, wzory raportów i protokołów z zakresu kontroli i odbiorów oprogramowania, dokumentacji i usług.

## **5.6 Plan weryfikacji, walidacji i testowania**

Zastosowanie uporządkowanej metodyki cyklu życia oprogramowania oraz odpowiednich narzędzi zarządzania projektem i umiejętne ich połączenie z wymaganiami klienta, jakim ma odpowiadać produkt docelowy tworzy warunki do oceny wytwarzanego lub kastomizowanego oprogramowania w trakcie całego jego cyklu życia.

Metody, techniki oraz procedury i narzędzia służące temu celowi sprecyzowane są w „Planie weryfikacji, walidacji i testowania” stanowiącym załącznik do „Planu jakości”.

„Plan weryfikacji, walidacji i testowania” opracowuje kierownik projektu, we współpracy z osobami kluczowymi w projekcie odpowiedzialnymi za poszczególne procesy produkcyjne, w oparciu o wzorcowy dokument Systemu Jakości ComputerLand S.A. „Plan weryfikacji, walidacji i testowania”, który należy dostosować do specyfiki projektu uwzględniając zapisy kontraktowe i przyjętą metodykę oraz model cyklu życia oprogramowania.

„Plan weryfikacji, walidacji i testowania” składa się z rozdziałów odpowiadających całemu cyklowi wytwórczemu oprogramowania. Ze względu na fakt, że prawie w każdym projekcie, poza produkcją lub dopasowaniem aplikacji do wymagań klienta, występują dodatkowo procesy związane z dokumentacją użytkową, szkoleniami, wdrożeniami w „Planie weryfikacji, walidacji i testowania” dodatkowo uwzględnia się weryfikacje związane z realizacją tych procesów.

Działania oceniające i kontrolne, jakie planuje się w trakcie cyklu życia projektu dotyczą:

- weryfikacji, czyli oceny produktu danej fazy projektowej w celu stwierdzenia, czy na zakończenie danej fazy spełnia on założenia przyjęte na jej początku,
- walidacji, czyli oceny produktu końcowego w celu sprawdzenia, czy spełnia on wymagania postawione mu w specyfikacji wymagań użytkownika na początku procesu produkcji,
- testowania, czyli badania jakości kodu oprogramowania, w celu sprawdzenia, przy pomocy specjalnie dobranych danych testowych i porównywania wyników uzyskanych z przewidywanymi, jak dobry jest produkt lub jego komponenty.

W dokumencie „Plan weryfikacji, walidacji i testowania”, dla każdego produktu pośredniego i końcowego:

- definiuje się kryteria, według których wykonywane jest jego sprawdzenie i ocena,
- określa się planowane techniki badań i weryfikacji (np. inspekcje, przeglądy wzajemne, recenzje ekspertów, symboliczna interpretacja kodu, testowanie funkcjonalne, testowanie strukturalne, pomiary osiągnięć),
- określa się plany z zakresu weryfikacji, walidacji i testowania, które są w trakcie trwania projektu rozwijane w postaci oddzielnych dokumentów oraz podlegają kolejnym i ostatecznym przeglądom, tak jak wszystkie inne produkty i dokumenty projektowe; plany te są tworzone na podstawie wzorcowych dokumentów Systemu Jakości ComputerLand S.A. po dostosowaniu do specyfiki projektu.

W „Planie weryfikacji, walidacji i testowania” ujęty jest również wykaz zaplanowanych działań weryfikacyjnych i punktów kontrolnych projektu ze wskazaniem osób odpowiedzialnych i punktów czasowych ich przeprowadzenia.

Zgodnie z obowiązującymi w ComputerLand S.A. procedurami, działania związane z weryfikacją, walidacją i testowaniem systemu są włączane do harmonogramu bazowego realizacji projektu.

„Plan weryfikacji, walidacji i testowania” ma charakter ewolucyjny, gdyż w każdej fazie cyklu wytwórczego podejmowane są kolejne decyzje i tworzą się warunki pozwalające planować weryfikację produktów następnej fazy do akceptacji systemu włącznie.

## 5.7 Plan zarządzania konfiguracją

Zarządzanie konfiguracją projektu obejmuje zasady i techniki zmierzające do identyfikacji, dokumentowania, śledzenia, oceny, sterowania i autoryzowania zmian we wszelkiej informacji projektowej, która ma być udostępniana różnym osobom związanym z projektem.

W celu sprawnego zarządzania konfiguracją kierownik każdego projektu obejmującego produkcję (kastomizację) oprogramowania w ComputerLand S.A. ma obowiązek powołać zespół ds. zarządzania konfiguracją projektu. Pierwszym



zadaniem tego zespołu jest opracowanie dokumentu „Plan zarządzania konfiguracją projektu”.

Wytoczne do opracowania tego dokumentu zawarte są we wzorcowym dokumencie Systemu Jakości ComputerLand S.A. „Plan zarządzania konfiguracją projektu”, w którym zaprojektowano mechanizmy pozwalające w każdej chwili otrzymać pełny obraz zmian w projekcie, dotrzeć do źródła ich powstania i prześledzić aktualne losy ich realizacji.

Zarządzanie konfiguracją obowiązuje w trakcie trwania projektu w stosunku składowych projektu począwszy od wskazanej wersji oraz w fazie utrzymania i serwisowania różnych wersji aplikacji.

W dokumencie „Plan zarządzania konfiguracją projektu” ujęte są następujące zagadnienia:

1. Określenie odpowiedzialności za zarządzanie konfiguracją – określenie składu osobowego i kompetencji zespołu ds. zarządzania konfiguracją.
2. Identyfikacja elementów projektu podlegających zarządzaniu konfiguracją - na podstawie prześledzenia cyklu życia projektu i zobowiązań kontraktowych zdefiniowanie listy elementów projektu, które będą podlegały zarządzaniu konfiguracją.

Dla każdego elementu konfiguracji określa się następujące dane:

- w jakim etapie (fazie, procesie projektowym) pojawi się dany element konfiguracji,
  - od kiedy będzie podlegał zarządzaniu konfiguracją wewnątrz projektu,
  - od kiedy będzie podlegał zarządzaniu konfiguracją w relacjach z klientem,
  - do kiedy będzie podlegał zarządzaniu konfiguracją,
  - kto będzie odpowiedzialny za zarządzanie konfiguracją danego elementu.
3. Oznaczenie i identyfikowalność konfiguracji – sposób oznaczania i utrzymywania wszystkich elementów konfiguracji projektu podczas wszystkich etapów projektu poczynając od określania wymagań poprzez analizę, projektowanie, implementację, testowanie, kopiowanie, dostarczanie, wdrożenie, pielęgnację i eksploatację uwzględniający następujące aspekty:
    - unikalność nazw,
    - wersjonowanie,
    - określanie statusu.
  4. Opracowanie struktury zawartości biblioteki i archiwum projektu w oparciu o standard repozytorium projektu – bazę wzorcową w Lotus Notes.
  5. Ustalenie zasad pracy zespołu projektowego nad elementami konfiguracji z uwzględnieniem:
    - kontroli i koordynacji równoczesnego uaktualniania poszczególnych elementów konfiguracji przez więcej niż jednego członka zespołu projektowego,
    - koordynacji uaktualniania różnych elementów konfiguracji w jednym lub kilku miejscach,

- identyfikacji i śledzenia wszystkich działań i zmian wynikających ze zmiany wymagań od momentu początkowego aż do wydania kolejnej wersji systemu.
6. Nadzór nad zmianami konfiguracji - autoryzacja zmian, jednoosobowa odpowiedzialność, specjalizacja, racjonalizacja.
  7. Informowanie o statusie konfiguracji – każdy dokument i produkt jest opisany metryką dokumentu z historią zmian i przeglądów (wg standardów przyjętych w Systemie Jakości ComputerLand S.A.), a każda nowa wersja (wydanie) oprogramowania opisana jest dokumentem „Metryka wersji systemu” jednoznacznie definiującym zawartość danej wersji (wytyczne do sporządzania takiego dokumentu zawarte są we wzorcowym dokumencie Systemu Jakości ComputerLand S.A. „Metryka wersji systemu (konfiguracji etapu/fazy projektu)”.
  8. Kontrola systemu zarządzania konfiguracją – zaplanowanie regularnych przeglądów systemu zarządzania konfiguracją, sposobu kontroli i dokumentowania jej wyników.
  9. Narzędzia, techniki, metodologie – lista elementów konfiguracji projektu i używanych w projekcie narzędzi, technik, metodologii wspomagających system kontroli wersji (do zarządzania wersjami kodu programu, przechowywania i śledzenia różnych dokumentów, zarówno tekstowych jak i binarnych).
- „Plan zarządzania konfiguracją projektu” jest uaktualniany w miarę rozwoju przedsięwzięcia w sposób wyprzedzający wykonanie czynności przypisanych poszczególnym fazom produkcji oprogramowania.

## 6. Realizacja projektu

Projekt jest realizowany zgodnie z procedurą Systemu Jakości „Zarządzanie realizacją projektu”, która obliguje kierownika projektu, zespół projektowy ze strony ComputerLand S.A. i klienta do stosowania przyjętej metodyki cyklu życia oprogramowania oraz zapisów zawartych w dokumentach konstytuujących projekt.

### 6.1 Zarządzanie realizacją projektu

Kierownik projektu zarządza realizacją projektu zgodnie z procedurami zdefiniowanymi i zatwierdzonymi na etapie planowania realizacji projektu, zawartymi w:

- „Planie Projektu”,
- „Planie jakości”,
- „Planie zarządzania konfiguracją projektu”,
- „Planie weryfikacji, walidacji i testowania”,
- Budżecie projektu.

W obszarach projektu nie pokrytych dedykowanymi dla projektu procedurami pracowników ComputerLand S.A. obowiązują standardowe procedury i dokumenty Systemu Jakości ComputerLand S.A.

Dzięki systemowi dekretowania i zatwierdzania dokumentów finansowych projektu tylko przez kierownika projektu oraz upoważnione przez niego osoby i powiązaniu budżetu projektu z systemem finansowym firmy oraz systemem rejestracji czasu pracy, wykonanie budżetu projektu jest na bieżąco monitorowane na zgodność z budżetem zaplanowanym przez aplikacje wspomagające controlling..

Zapisy z zarządzania projektem (harmonogramy, notatki, raporty, zgłoszenia problemów, wnioski o zmianę, zgłoszenia zagrożeń, korespondencja, itp.) przechowywane są w bibliotece projektu.

## **6.2 Realizacja procesów związanych z cyklem życia oprogramowania**

Procesy produkcyjne są realizowane zgodnie z przyjętą metodyką, podejściem technicznym i cyklem życia oprogramowania opisanym w „Planie projektu” oraz przyjętym harmonogramem.

Powstają produkty przejściowe i finalne poszczególnych faz procesu wytwórczego. Wytyczne do realizacji poszczególnych faz projektowych zawarte są w dokumentach Systemu Jakości ComputerLand S.A.:

- „Planowanie i realizacja projektów obejmujących produkcję, kastomizację i wdrożenie oprogramowania”
- „Planowanie i realizacja fazy analizy (określanie wymagań i modelowanie) dla projektów obejmujących produkcję (kastomizację) oprogramowania”
- „Planowanie i realizacja fazy projektowania dla projektów obejmujących produkcję (kastomizację) oprogramowania”
- „Planowanie i realizacja fazy implementacji (konstrukcji) dla projektów obejmujących produkcję (kastomizację) oprogramowania”
- „Testowanie”.

Wszelkie zmiany są zarządzane zgodnie z procedurą zarządzania zmianami obowiązującą w projekcie.

## **6.3 Weryfikacja, walidacja, testowanie**

Wyprodukowane w trakcie realizacji projektu produkty przejściowe i finalne poszczególnych faz procesu wytwórczego podlegają procesom weryfikacji zgodnie z działaniami zaplanowanymi w „Planie weryfikacji, walidacji i testowania”. Opracowane narzędzia i techniki oraz zapisy z przeglądów, inspekcji, testów przechowywane są w bibliotece projektu i repozytorium testowym.

Repozytorium testowe jest najczęściej bazą Lotus Notes z wbudowanym elektronicznym przepływem prac i dokumentów.

Każdy produkt, przed przekazaniem do klienta, podlega odbiorowi wewnętrznemu. Walidacja systemu jest wspierana przez testy akceptacyjne zgodnie z uzgodnionymi z klientem procedurami akceptacji i odbiorów.



## 7. Pielęgnacja i serwis oprogramowania aplikacyjnego

Po zakończeniu wdrożenia wyprodukowanego oprogramowania aplikacyjnego, na które zazwyczaj składają się następujące procesy realizowane przez zespoły wdrożeniowe:

- przygotowanie i przeprowadzenie szkoleń,
- akceptacja systemu (zazwyczaj w instalacji testowej u klienta),
- instalacja systemu w miejscu docelowym,
- przeprowadzenie konwersji danych z systemów lokalnych,
- przeprowadzenie migracji danych z poprzedniej wersji oprogramowania,
- instalacja interfejsów do/z systemów zewnętrznych,
- uruchomienie systemu,
- powielenie instalacji i dystrybucja oprogramowania

oraz podpisaniu stosownych protokołów odbioru system przechodzi u klienta w fazę eksploatacji, w ComputerLand S.A. w fazę pielęgnacji (maintenance).

Każde oprogramowanie aplikacyjne, jest pielęgnowane i serwisowane zgodnie z zapisami w umowach serwisowych z klientami oraz planami produkcji i rozwoju oprogramowania aplikacyjnego przez dane centrum kompetencyjne.

Wersjonowanie produktów programowych składających się na utrzymywane oprogramowanie aplikacyjne jest realizowane zgodnie z planami zarządzania konfiguracją oprogramowania.

Proces pielęgnacji obejmuje działania związane z utrzymywaniem produktu w długim czasie, do których zaliczamy:

- poprawianie błędów i usterek,
- wprowadzanie ulepszeń i modyfikacji,
- rozwój funkcjonalności,
- dostosowanie do zmian prawnych,
- przenoszenie do innego środowiska sprzętowo-programowego,
- usuwanie awarii.

Do wspomagania prac serwisowych wykorzystywane są standardowe dla ComputerLand S.A. procedury zgłaszania i obsługi zgłoszeń serwisowych wbudowane w aplikacje help-desk'owe z rozwijanymi bazami wiedzy i wbudowanymi mechanizmami kontroli terminowości wykonania działań serwisowych. Zgłoszenia serwisowe, które nie mogą być rozwiązane pierwszej linii help-desku są przesyłane do produktowych centrów kompetencyjnych. Po pełnym rozpoznaniu problemu opisanego w zgłoszeniu serwisowym przez centrum kompetencyjne przesyłana jest do działu serwisu szczegółowa diagnoza zgłoszenia oraz propozycja rozwiązania problemu. Jeżeli okaże się, że konieczne jest wykonanie zmian w oprogramowaniu aplikacyjnym, uruchamiany jest proces produkcji nowej wersji (wydania) oprogramowania obejmujący cały cykl produkcyjny, zgodnie z planem produkcji i rozwoju oprogramowania.

## 8. Kontrola projektów

Realizowane w ComputerLand S.A. projekty, zgodnie z zapisami w dokumentach Systemu Jakości, podlegają systematycznym kontrolom i ocenom.

### 8.1 Raportowanie z projektu

Każdy kierownik projektu, poza ścieżkami raportowania ustalonymi z klientem, ma obowiązek miesięcznego raportowania do dyrektora projektu oraz dyrektora centrum kompetencyjnego realizującego projekt.

Formularz „Raport stanu projektu” jest dokumentem Systemu Jakości i składa się z następujących elementów:

- kontrola terminowości i realizacji prac – w stosunku do punktów kontrolnych i kamieni milowych w harmonogramie bazowym projektu,
- kontrola budżetu projektu – przychody rzeczywiste w stosunku do przychodów planowanych, koszty rzeczywiste w stosunku do kosztów planowanych,
- raport z realizacji prac z wyszczególnieniem zadań zrealizowanych, zadań nie zrealizowanych z podaniem przyczyny i wykazu podjętych akcji,
- plan na najbliższe 2 miesiące,
- rejestr problemów,
- rejestr zagrożeń.

Na podstawie raportu, adresaci raportu mogą śledzić realizację każdego projektu i reagować na swoim poziomie kompetencji odpowiednio wcześniej lub eskalować podjęcie koniecznych decyzji na poziom dyrektora generalnego właściwego sektora lub Zarządu ComputerLand S.A. W sytuacjach trudnych mogą wystąpić o przeprowadzenie audytu wewnętrznego i wskazanie działań naprawczych.

### 8.2 Audyty wewnętrzne

Każdy projekt realizowany w oparciu o umowę z klientem, dla którego ryzyko w bazie „Przegląd umowy” oszacowano jako duże lub ekstremalne podlega systematycznym audytom wewnętrznym. Audyty są przeprowadzane przez certyfikowanych audytorów systemu ISO 9001, nie związanych z projektem, którzy posiadają merytoryczne przygotowanie i doświadczenie w zakresie inżynierii oprogramowania i wewnętrznych dokumentów Systemu Jakości ComputerLand S.A.

Audyty innych projektów są realizowane wybiórczo lub na zlecenie kierownictwa ComputerLand S.A.

Systematycznie są również przeprowadzane audyty w poszczególnych centrach kompetencyjnych zajmujących się produkcją i wdrażaniem oraz serwisem i pielęgnacją oprogramowania. Raporty z audytów są przechowywane w bazie „Raporty z audytów”.

W wyniku audytu zlecane są działania zapobiegawcze i korygujące ze wskazaniem terminu realizacji i osób odpowiedzialnych za ich implementację. Realizacja zaleceń jest kontrolowana systematycznie.

## 9. Analiza funkcjonowania Systemu Jakości ComputerLand S.A. w zakresie produkcji oprogramowania

Droga do rozszerzenia certyfikatu ISO 9001 o produkcję i kastomizację oprogramowania nie była łatwa:

- własnymi siłami ComputerLand S.A opracowano wzorcowe dokumenty Systemu Jakości z zakresu produkcji i kastomizacji oprogramowania oraz zarządzania projektami, pokrywające wszystkie procesy od przygotowania oferty, przez przegląd umowy, procesy produkcyjne i zarządcze aż do pielęgnacji i utrzymania,
- zaprojektowano i wykonano w technologii Lotus Notes szereg narzędzi wspierających wdrożenie Systemu Jakości w centrach kompetencyjnych,
- opracowano i przeprowadzono serię szkoleń dla pracowników centrów kompetencyjnych i kierowników projektów zarówno metodą tradycyjną jak i intranetową, przy czym szkolenia w trybie „distance-learning” były zakończone egzaminem,
- w każdym centrum kompetencyjnym przeprowadzono serię audytów wewnętrznych, mających na celu doprowadzenie do osiągnięcia zgodności z procedurami Systemu Jakości za pomocą działań korygujących i zapobiegawczych oraz weryfikacji opisywanych i realizowanych procesów,
- w każdym centrum kompetencyjnym, dla każdego produkowanego (kastomizowanego) tam oprogramowania aplikacyjnego powstały:
  - plan (projektu) produkcji i rozwoju,
  - plan jakości,
  - plan weryfikacji walidacji i testowania,
  - plan zarządzania konfiguracją,
  - procedury, formularze, instrukcje opisujące realizowane procesy.
- opracowane dokumenty zostały wdrożone, a właściwa realizacja procesów i dokumentowanie w narzędziach oraz repozytoriach projektowych zostały potwierdzone w audytach wewnętrznych,
- każdy projekt, realizowany przez zespoły projektowe w oparciu o umowę z klientem, był kontrolowany na zgodność z obowiązującymi standardami jakości.

Poprzedzający certyfikację na zgodność z normą ISO 9001, 3 dniowy próbny audyt w zakresie produkcji oprogramowania miał miejsce w lutym 2001 w centrach kompetencyjnych Sektora Telekomunikacyjnego w Gdańsku oraz w Warszawie w centrach kompetencyjnych Sektora Bankowego i centrach kompetencyjnych zajmujących się kastomizacją i wdrażaniem Oracle Applications.



W cztery miesiące później, w tych samych lokalizacjach odbył się audyt certyfikujący.

Na początku 2002 roku audytorzy zewnętrzni stwierdzili prawidłowe funkcjonowanie systemu w trakcie 3 dniowego audytu sprawdzającego w centrach kompetencyjnych Sektora Bankowego we Wrocławiu oraz w centrach kompetencyjnych technologicznych w Warszawie.

W trakcie każdego z audytów zewnętrznych, obok nowego zakresu certyfikatu ISO 9001 audytorzy badali również funkcjonowanie i przestrzeganie procedur Systemu Jakości w dotychczas obowiązującym zakresie certyfikatu ISO 9001.

W celu utrzymania zakresu certyfikatu ISO 9001 w zakresie produkcji i kastomizacji oprogramowania, audytorzy wewnętrzni przeprowadzają regularne audyty wewnętrzne, przy czym każde centrum kompetencyjne jest audytowane 3-4 razy w roku.

Aktualnie tym zakresem certyfikatu ISO 9001 objętych jest około 1000 pracowników w 25 centrach kompetencyjnych. Dwa razy do roku, każdy pracownik firmy ma obowiązek zdać egzamin na szkoleniu internetowym z systemu jakości. Przynajmniej raz w roku przeprowadzana jest ankieta u klientów ComputerLand S.A.. Wyniki są pieczołowicie zbierane i analizowane. Osiągane rezultaty świadczą o wzroście notowań ComputerLand S.A. w stosunku do oczekiwań klientów i zeszłorocznych wyników.

Monitorowaniu podlegają zbierane w trakcie procesu produkcji i utrzymania oprogramowania miary dotyczące w różnych aspektach: wymagań, zmian, problemów testowych, błędów i braków, reklamacji, zgłoszeń problemów, zgłoszeń serwisowych, harmonogramu, budżetu, itd.

Wdrożenie i utrzymanie Systemu Jakości nie byłoby możliwe bez wyraźnego wsparcia Zarządu ComputerLand S.A., kierowników wszystkich szczebli i ogromnego zaangażowania i przekonania pracowników.

Kolejny audyt zewnętrzny będzie miał miejsce tuż po wakacjach i prawdopodobnie obejmie włącznie z początkiem roku 2002 w struktury ComputerLand S.A. centra kompetencyjne Sektora Zdrowie w Gliwicach oraz centra kompetencyjne nowych technologii w Warszawie. W czasie tego audytu zostaną prawdopodobnie również zaudytowane centra kompetencyjne Sektora Publicznego w Gliwicach, Krakowie i Warszawie.

## **9.1 Silne strony – z czego jesteśmy zadowoleni**

Do silnych stron wdrożonego w ComputerLand S.A. rozwiązania można zaliczyć:

- połączenie (zintegrowanie) Systemu Jakości z narzędziami finansowymi - wprowadzenie mechanizmów łączących budżet projektu z systemem finansowym, systemem zleceń wewnętrznych i systemem rejestracji czasu pracy, co umożliwi bieżące śledzenie kosztów i marży na każdym projekcie oraz w działach handlowych i produkcyjnych,
- zmniejszenie kosztów wytwarzania i utrzymania oprogramowania,

- poprawę jakości wytwarzanego oprogramowania aplikacyjnego charakteryzującą się np. obserwowanym spadkiem ilości zgłoszeń serwisowych,
- wzrost zadowolenia klientów z jakości dostarczanego oprogramowania,
- zmniejszenie ilości ujawnianych błędów oraz skrócenie cykli wytwórczych oprogramowania dzięki wprowadzeniu systematycznych weryfikacji i wielopoziomowych testów,
- zwiększenie % projektów kończonych zaplanowanym czasie i budżecie,
- budowanie oprogramowania w bliskiej współpracy z klientem,
- możliwość strojenia i ulepszania procesów produkcyjnych dzięki zbieraniu pomiarów i analizie statystycznej wyników,
- uczenie się organizacji,
- wymienialność osób w zespołach,
- poprawę procesu zarządzania projektami,
- powołanie osób odpowiedzialnych za jakość w każdym centrum kompetencyjnym,
- uzyskanie przewagi rynkowej nad konkurencją - mało dużych polskich firm informatycznych ma certyfikat ISO 9001 w zakresie produkcji oprogramowania,
- możliwość uczestniczenia w przetargach, w których jest wymagane posiadanie certyfikatu ISO 9001 w zakresie produkcji oprogramowania.

## 9.2 Słabe strony – nad czym musimy jeszcze popracować

Do słabych stron wdrożonego rozwiązania można zaliczyć:

- brak jednolitych narzędzi korporacyjnych do produkcji oprogramowania i zarządzania wiedzą (centra kompetencyjne wykorzystują różne technologie),
- za dużo dokumentów jakości ma charakter lokalny – problemy z utrzymaniem standardu jakości i przeprowadzaniem kontroli,
- brak jednolitych kryteriów doskonalenia jakości (opomiarowanie procesów produkcyjnych).

## 9.3 Dalsze plany – cele, które chcemy osiągnąć

Wśród planów, których realizacja ma pozbawić System Jakości ComputerLand S.A. słabych stron i wzmocnić jego silne strony oraz zapewnić firmie większą przewagę konkurencyjną na rynku produkcji oprogramowania i integracji rozwiązań należy wymienić:

- recertyfikację z normy ISO 9001:1994 na normę ISO 9001:2000 do końca 2003 roku,
- wprowadzenie profesjonalnych korporacyjnych narzędzi do produkcji oprogramowania, które zapewnią spójność i jednolitość realizowanych procesów produkcyjnych,

- uzyskanie pełnej standaryzacji dokumentów jakości przez odejście od lokalnych dokumentów Systemu Jakości,
- dokładne i standardowe opomiarowanie procesów produkcyjnych,
- dokładne i standardowe opomiarowanie procesów zarządczych,
- osiągnięcie 4 poziomu dojrzałości (Managed - Zarządzany) wg modelu CMM (Capability Maturity Model – model dojrzałości zdolności produkcyjnych oprogramowania), opracowanego przez SEI (Software Engineering Institute – instytut inżynierii oprogramowania) w Pittsburgu (USA) do końca 2004 roku.

## 10. Literatura

1. Polski Komitet Normalizacyjny. PN-ISO 9001:1996 – Model zapewnienia jakości w projektowaniu, pracach rozwojowych, produkcji, instalowaniu i serwisie. Wydawnictwa Normalizacyjne ALFA – WERO Sp. z o.o., Warszawa, 1996
2. Polski Komitet Normalizacyjny. PN-ISO 9000-3:1994 – Wytyczne do stosowania normy ISO 9001 podczas opracowywania, dostarczania i obsługiwanania oprogramowania. Wydawnictwa Normalizacyjne ALFA – WERO Sp. z o.o., Warszawa, 1995
3. Polski Komitet Normalizacyjny. PN-EN ISO 9000:2001 – Systemy Zarządzania jakością. Podstawy i terminologia. Wydawnictwa Normalizacyjne ALFA – WERO Sp. z o.o., Warszawa, 2001
4. J. Górski i inni. Inżynieria oprogramowania w projekcie informatycznym. MIKOM, Warszawa, marzec 1999
5. B. Begier. Inżynieria oprogramowania – problematyka jakości. Wydawnictwo Politechniki Poznańskiej, Poznań 1999
6. A. Jaskiewicz. Inżynieria oprogramowania. HELION, Gliwice, 1997

mgr inż. Lilianna Wierzchoń

ComputerLand S.A.

02-486 Warszawa, Al. Jerozolimskie 180, tel. +48 22 571 11 32

e-mail: lwierzchon@computerland.pl



## ROZDZIAŁ 3

# SZTUKA PROJEKTOWANIA SYSTEMÓW



# STRATEGIA ZAPEWNIENIA JAKOŚCI OPROGRAMOWANIA

Anna BOBKOWSKA

**Streszczenie:** Zapewnienie jakości jest bardzo istotnym aspektem projektu informatycznego. Niestety, jest ono obszarem trudnym i pociągającym za sobą dodatkowe nakłady. Istnieje potrzeba wprowadzania systemów jakości dopasowanych do specyfiki konkretnych firm informatycznych. W referacie zaproponowano strategię zapewniania jakości oprogramowania, bazującą na działaniach w następujących czterech obszarach:

- Określanie definicji jakości oraz sposobu weryfikacji jej osiągnięcia,
- Dobór technik dotyczących poprawy produktu,
- Poprawa procesu wytwarzania oprogramowania,
- Podnoszenie kultury jakości organizacji.

Proponowane podejście bazuje na zasadach TQM (ang. Total Quality Management) oraz idei eksperymentalnego doboru użytecznych technik.

## Wstęp

Prace związane z jakością oprogramowania mają już swoją historię. Wyróżnione zostały okresy sprawdzania jakości (weryfikacja produktów), sterowania jakością (sterowanie procesem wytwarzania w celu otrzymania produktu o wymaganej jakości) i zapewnienia jakości (zaangażowanie kierownictwa, system sterowania i poprawy jakości). Obecnie mówi się o zarządzaniu jakością, które jest ukierunkowane na zaspokojenie oczekiwań użytkownika i efektywność ekonomiczną przy zaangażowaniu wszystkich pracowników. Zaproponowano wiele technik wspomagających otrzymanie produktu o wymaganej jakości, metryk dla różnych charakterystyk jakości oraz metod, których zadaniem było wyeliminowanie problemów pojawiających się w projektach. Wiele z nich wdrożono już w firmach i zebrano cenne doświadczenia. Powstały także systemy jakości (np. ISO [14,16], TQM [10,18,19]), obejmujące swoim zakresem strukturę organizacyjną, podział odpowiedzialności, procedury, procesy, i zasoby umożliwiające wdrożenie i realizację zarządzania jakością, oraz modele dojrzałości oprogramowania (CMM [8], CMMI [9], SPICE [15]). Mimo tak wielu prac, problem jakości oprogramowania nadal nie jest rozwiązany.

Wśród przyczyn takiego stanu rzeczy można wyróżnić:

- Rozwój technologii informatycznych (technologie obiektowe, internetowe), co jest związane z zapotrzebowaniem na nowe techniki w zakresie zapewnienia jakości lub koniecznością dopasowania istniejących technik do tych technologii;
- Wzrost wymagań względem systemów oraz związany z tym wzrost złożoności oprogramowania;
- Potrzeba dopasowania systemów jakości i modeli dojrzałości do potrzeb konkretnych zespołów i organizacji;



- Problemy związane z wdrożeniem systemów jakości i wykonywaniem zadań pro-jakościowych.

W raportach ze stosowania technik związanych z zapewnieniem jakości znajduje się wiele danych świadczących o tym, że bardzo ważna jest strategia jakości dopasowana do konkretnej organizacji oraz zrozumienie i akceptacja jej przez wszystkich zaangażowanych. Negatywnym przykładem może być firma, w której wdrożono bardzo skomplikowany system pomiarów procesu wytwarzania oprogramowania, a później okazało się, że większość gromadzonych danych nie była przez nikogo wykorzystywana. Natomiast pozytywnym przykładem może być dość powszechna praktyka, żartobliwie nazwana "poziomem 6 CMM", oznaczająca wybór ze wszystkich poziomów CMM, tych zaleceń, które są przydatne w konkretnej organizacji.

Ciekawe są także raporty dotyczące ciągłego dążenia do jakości. Pokazują one, że wcale nie jest to prosty proces, a wyeliminowanie większych problemów często prowadzi do ujawnienia się mniejszych. Przykładem może być brak oczekiwanej jakości oprogramowania pomimo identyfikacji problemów oraz iteracyjnego wprowadzania zaawansowanych metod pro-jakościowych, z zarządzaniem ryzykiem włącznie. Osoby odpowiedzialne za jakość, charakteryzujące się wielką determinacją w dążeniu do celu, próbują obecnie do swoich firm wprowadzać zarządzanie etyką i odpowiednią kulturę jakości. Ta tendencja podkreśla rolę człowieka w procesie zapewnienia jakości. Odpowiednia inteligencja, poziom wykszolenia, motywacja i zaangażowanie wykonawców oprogramowania pełnią decydującą rolę w dążeniu do sukcesu. Jednakże ważne jest także organizacyjne, techniczne i kulturowe wsparcie dla działań poszczególnych jednostek, gdyż pozwala ono na pełne wykorzystanie potencjału zaangażowanych w projekt informatyków.

W referacie zaproponowano strategię zapewnienia jakości oprogramowania opartą na filozofii TQM (ang. Total Quality Management) oraz idei eksperymentalnego dopasowania struktur, metod i technik do potrzeb konkretnych firm, czy też zespołów projektowych. W rozdziale 2 wyjaśniono ideę strategii oraz zaprezentowano cztery obszary działań, w następnych czterech rozdziałach - omówiono te obszary, natomiast w ostatnim rozdziale przedstawiono podsumowanie oraz perspektywy dalszych prac w tym zakresie.

## 1. Podstawy strategii zapewnienia jakości

TQM (ang. Total Quality Management, tłum. Kompleksowe zarządzanie jakością) jest filozofią zarządzania ukierunkowaną na ciągłe spełnianie oczekiwań klientów przy najniższych kosztach i rzeczywistym zaangażowaniu wszystkich pracowników w działania pro-jakościowe na swoim stanowisku pracy. TQM zakłada:

- Zaangażowanie wszystkich pracowników, więc również kierownictwa – menadżerowie są odpowiedzialni za określanie zakresu i celów jakości, polityki jakości, planów jakości oraz przegląd i przeformułowywanie celów.
- Koncentracja na klientach i pracownikach - celem jest satysfakcja klienta i rozwój możliwości pracowników.

- Koncentracja na faktach – system ciągłych pomiarów, gromadzenia danych i prezentowania faktów w celu śledzenia postępów.
- Ciągłe doskonalenie – rozwój przedsiębiorstwa poprzez ciągłą poprawę i doskonalenie systemu jakości oraz kwalifikacji pracowników.

Jest wiele zaleceń, m.in. zalecany jest pomiar sukcesu, praca zespołowa i efektywna komunikacja, pozytywna postawa pracowników i motywacje, dobra współpraca z klientem. Konsekwencjami są zadowoleni klienci i pracownicy, wypracowanie efektywnych procedur jakościowych, poprawa komunikacji pomiędzy działami i pomiędzy pracownikami oraz przyjęcie odpowiedzialności pracowników za swoją pracę. Podstawą "koncentracji na pracownikach" jest przekonanie, że na zwiększenie efektywności pracy ma wpływ zadowolenie pracowników ze swojej pracy oraz realny wpływ na poprawę procesy wytwarzania i procedur jakości. Natomiast koncentracja na faktach i ciągłe doskonalenie otwiera drogę dla metod eksperymentalnych [1], które mogą być stosowane nie tylko w badaniach naukowych, ale także do poprawy sytuacji w konkretnej firmie.

Skuteczna i efektywna strategia zapewniania jakości oprogramowania powinna umożliwiać samo-doskonalenie przy zastosowaniu metod eksperymentalnych oraz powinna uwzględniać działania w następujących czterech obszarach:

- Określanie definicji jakości oraz sposobu weryfikacji jej osiągnięcia,
- Dobór technik dotyczących poprawy produktu,
- Organizacja procesu wytwarzania i działania wspomagające,
- Podnoszenie kultury jakości organizacji.

Określanie definicji jakości polega na skonkretyzowaniu, co zespół projektowy będzie rozumiał pod pojęciem jakości, a sposób weryfikacji osiągnięcia jakości sprowadza się do określenia miar, ocen i procedur mających na celu sprawdzenie, czy rzeczywiście jakość jest osiągana. Dobór technik dotyczących poprawy produktu polega na wyborze odpowiednich technik wytwarzania, dokumentacji i weryfikacji produktu oraz półproduktów za pomocą inspekcji i testów. Organizacja procesu wytwarzania uwzględnia określenie procesu wytwarzania, plany i procedury dokumentacyjne, komunikacyjne oraz związane z zarządzaniem problemami, zmianami itp. Kultura jakości organizacji dotyczy wspólnych przekonań służących realizacji celów organizacji na poziomie wartości i postaw. Uwzględnia ona takie czynniki jak system motywacji, satysfakcja klienta i pracowników, zasady współpracy. Jest szereg danych świadczących o tym, że organizacje o wyższej kulturze utrzymują wysoką efektywność, produktywność i innowacyjność w dłuższym okresie czasu.

## **2. Definicja jakości oraz weryfikacji jej osiągnięcia**

Określanie definicji jakości aplikacji oraz sposobu weryfikacji jej osiągnięcia jest podstawą działań nad poprawą jakości produktu. Pojęcie jakości



jest abstrakcyjne, zmienne i subiektywne, więc dopiero określenie jego ujednoczonego rozumienia w projekcie umożliwia podjęcie prac nad jej zapewnieniem. To skonkretyzowanie rozumienia jakości wcale nie jest łatwe, dlatego często jest ono zanedbywane i pozostawiane pod hasłami „zapewnienia satysfakcji użytkownika i efektywności ekonomicznej”. Pomijając tutaj problem właściwych relacji pomiędzy jakością oprogramowania i efektywnością ekonomiczną, warto zauważyć fakt, że o ile efektywność ekonomiczną można zmierzyć w powszechnie znanych jednostkach (koszty w poszczególnych walutach, nakład w osobomiesiącach, czas trwania projektu w tygodniach lub miesiącach), to dla ogólnie rozumianej jakości oprogramowania takie naturalne jednostki nie istnieją. Konieczne jest więc określenie poszczególnych atrybutów i charakterystyk jakości. Pomocą w tym celu mogą być standardy jakości produktu [17], ale ponieważ one zawierają nadmiarowy zbiór atrybutów i charakterystyk, w każdym rzeczywistym projekcie jest potrzeba dostosowania tych atrybutów do specyfiki wytwarzanej aplikacji, oczekiwań klientów oraz priorytetów firmy. Atrybutami jakości, od których można rozpocząć analizę oczekiwań jakościowych może być następujący zestaw [7]:

- *Wiarygodność* (ang. *Dependability*) - stopień zaufania do systemu;
  - *Satysfakcja* (ang. *Satisfaction*) - zespół cech określających stopień zadowolenia użytkownika;
  - *Funkcjonalność* (ang. *Functionality*) - stopień dopasowania produktu do potrzeb funkcjonalnych;
  - *Elastyczność* (ang. *Flexibility*) - umiejętność dopasowania się systemu do zachodzących zmian;
  - *Wydajność* (ang. *Performance*) - zbiór cech związanych z osiąganiami systemu.
- Następnie należy poddać analizie charakterystyki poszczególnych atrybutów, które w przypadku „satysfakcji” mogą mieć postać:
- *Łatwość użycia* (ang. *Ease of use*) - łatwość obsługi produktu;
  - *Zrozumiałość* (ang. *Understandability*) - łatwość zrozumienia produktu wynikająca z zastosowanych konwencji jego opisu;
  - *Łatwość nauki* (ang. *Learnability*) - nakład pracy wymagany do zdobycia umiejętności posługiwania się produktem;
  - *Produktywność* (ang. *Productivity*) - stopień wspomaganie zadań użytkownika przez system;
  - *Akceptacja* (ang. *Acceptance*) - stopień akceptacji innych wymagań użytkownika.

Tak zdefiniowana „satysfakcja”, chociaż w swojej nazwie nie różni się od „satysfakcji” z hasła, to jednak ma zupełnie inne znaczenie, które pozbawione jest dużej części abstrakcji i niejasności. Aby jeszcze bardziej urealnić „satysfakcję” należy określić sposób weryfikacji poszczególnych charakterystyk tego atrybutu. Weryfikacja polega na podaniu szeregu pomiarów i ocen subiektywnych, które mają być wykonane w celu stwierdzenia, czy wykonany produkt osiągnął



wymaganą jakość. Nie można tutaj pominąć roli doboru odpowiednich udziałowców oceny i technik ich wspomagających. Okazuje się, że osoby w zależności od doświadczeń profesjonalnych i edukacyjnych oraz osobowości są w różnym stopniu wyczulone na poszczególne charakterystyki jakości i naturalnym wykorzystaniem tego potencjału jest zaangażowanie ich do oceny różnych aspektów jakości.

### 3. Dobór technik poprawy produktu

Funkcjonalność jest uniwersalnym atrybutem jakościowym aplikacji. Ważne jest spełnienie celów systemu i dopasowanie jego usług do potrzeb i oczekiwań użytkowników. W tym celu stosowane mogą być rozmaite sposoby pozyskiwania wymagań użytkownika. Oprócz tradycyjnej analizy systemu i wywiadów z przyszłymi użytkownikami, można także wykorzystać [21]:

- Wywiady z różnymi udziałowcami projektu, pozyskiwanie wiedzy dziedzinowej z literatury, dokumentów, od ekspertów dziedzinowych;
- Ankietowanie grup użytkowników i metodę QFD (ang. Quality Function deployment) dla produktów masowych;
- Wnioski z zastosowania istniejących aplikacji o podobnym zastosowaniu;
- Prototypowanie, które służy nie tylko zrozumieniu systemu, ale także umożliwia łatwą komunikację i weryfikację przez użytkownika;
- Metody modelowania procesów biznesowych i w tym kontekście można przeprowadzić analizę wymagań;
- A także inne zalecenia inżynierii wymagań, takie jak dokładne określenie celów, wizji i zakresu systemu, kategoryzacja i priorytetyzowanie wymagań oraz zarządzanie zmianami.

Dobrze wykonana dokumentacja techniczna również przyczynia się do jakości aplikacji. Zastosowanie metod analizy i projektowania obiektowego umożliwia rozpoznanie cech rzeczywistego obiektu przed jego zbudowaniem, zrozumienie problemu oraz redukcję złożoności poprzez wyróżnienie cech istotnych dla rozwiązywanego problemu. Technologia obiektowa jest zbliżona do naturalnych procesów poznawczych człowieka i charakteryzuje się jednorodnością koncepcyjną i notacyjną we wszystkich etapach wytwarzania. Ponadto ułatwia modularyzację i podział prac na zespoły oraz umożliwia powtórne wykorzystanie komponentów. Dobrze wykonana dokumentacja może być poddana inspekcji, a także wykorzystana do weryfikacji systemu podczas testów.

Inspekcje mają na celu [11,12]:

- wskazanie błędów i sposobów ich poprawy w produkcie lub potwierdzenie, że zmiany nie są potrzebne;
- poprawę jakości produktu poprzez wzrost prognozowalności jakości i usprawnienie zarządzania produktem;
- polepszenie komunikacji w zespole realizującym dany projekt;
- zapewnienie śledzenia postępu prac w sensie jakościowym, a nie tylko

ilościowym oraz wiarygodne informowanie kierownictwa o jakości prac technicznych;

- zwiększenie jakości i obniżenie kosztów wytwarzania, a także doskonalenie procesu wytwarzania oprogramowania i procesu inspekcji oraz rozwój kultury jakości, co jest osiągnięte w firmie w długoterminowym okresie.

Poszczególne rodzaje inspekcji różnią się stopniem formalizmu, liczbą zaangażowanych osób oraz strategią stosowaną podczas analizy dokumentu [12]. Przeglądy polegają na przesłedzeniu treści dokumentu prezentowanego przez autora zgodnie z jego logicznym uporządkowaniem oraz zgłoszeniu uwag przez uczestników przeglądu. Inspekcje Fagana są bardziej sformalizowane, a zakres analizy jest wyznaczany przez listy kontrolne lub reguły. Recenzje polegają na indywidualnej kontroli z przedstawieniem wyników na piśmie, a audyt prowadzony jest pod kątem spełnienia określonego standardu. Wśród korzyści związanych z zastosowaniem inspekcji można wyróżnić: wzrost produktywności i redukcję czasu i kosztów wytwarzania, redukcję kosztów utrzymania, poprawę jakości procesu i produktów, eliminację opóźnień czasowych w dostarczaniu produktów, systematyczne usuwanie defektów, wyłonienie krytycznych obszarów wytwarzania oprogramowania oraz wypracowanie bardziej efektywnych standardów, a także podnoszenie jakości i kwalifikacji wykonawców oprogramowania poprzez udział w szkoleniach, spotkaniach kontrolnych, itp. Wyniki odpowiednio przeprowadzonej inspekcji można poddać przetworzeniu w celu uzyskania prognozowanej jakości oprogramowania [2].

Testy [23] są najbardziej tradycyjną i w zasadzie niezastąpioną metodą weryfikacji jakości oprogramowania. Bardzo ważne jest opracowanie strategii testowania, która powinna uwzględniać: modele i scenariusze testowe, zakres, metody i procedury testowania, pomiary testów, narzędzia wspomagające proces testowania oraz przydział odpowiedzialności za zadania testowania i poprawę produktu.

#### **4. Organizacja procesu wytwarzania**

Sposób wykonywania prac związanych z wytwarzaniem oprogramowania ma wpływ na jakość produktów końcowych. Takie przekonanie jest podstawą metod poprawy procesu wytwarzania i modeli dojrzałości procesu wytwarzania. Chociaż znane są przykłady wspaniałych procesów w ramach których powstają słabej jakości produkty, to jednak trudno się nie zgodzić ze stwierdzeniem, że im bardziej dojrzały jest proces wytwarzania, tym mniejsze ryzyko wytworzenia produktu o słabej jakości, a dla inwestora - mniejsze ryzyko niepowodzenia projektu przy zleceniu go w danej firmie.

Dla przykładu poniżej omówię podstawowe założenia i wybrane praktyki modelu SPICE (ang. Software Process Improvement and Capability Determination) [15]. SPICE jest standardem ISO/IEC mającym na celu określenie dojrzałości procesu wytwarzania w danej organizacji, poprawę procesu oraz

umożliwienie potwierdzenia określonego poziomu przez audyt. Jest uznawany za model bardziej nowoczesny i elastyczny w porównaniu z CMM. Podstawą do wykonania powyższych celów są podstawowe praktyki, które są określone w pięciu kategoriach:

- klient-dostawca – procesy, w które zaangażowany jest klient,
- inżynieria – procesy specyfikacji, implementacji lub utrzymania systemu,
- projekt – procesy związane z powstaniem projektu, jego koordynacją i zarządzaniem,
- wsparcie – procesy wspomagające wykonanie innych procesów,
- organizacja – procesy związane z celami biznesowymi i strukturą ich realizacji.

Wyróżnionych zostało sześć poziomów dojrzałości procesu:

*Poziom 0:* Brak – nie ma spełnienia podstawowych praktyk dotyczących procesu i brak określonych produktów.

*Poziom 1:* Nieformalny – podstawowe praktyki systematycznego wytwarzania są wykonywane, ale nie są rygorystycznie planowane i śledzone. Są zidentyfikowane produkty prac dla procesu.

*Poziom 2:* Planowany i śledzony – podstawowe praktyki są planowane i śledzone, a wydajność jest weryfikowana zgodnie z określonymi procedurami.

*Poziom 3:* Dobrze zdefiniowany – proces jest planowany i zarządzany przy użyciu standardowego procesu organizacji.

*Poziom 4:* Sterowany ilościowo – związany z ilościowym rozumieniem procesu i zwiększonymi możliwościami przewidywania. Efektywność jest obiektywnie zarządzana, a jakość produktów jest określana ilościowo.

*Poziom 5:* Stale poprawiany – stała poprawa procesu pod względem celów biznesowych, która jest możliwa dzięki ilościowemu sprzężeniu zwrotnemu ze zdefiniowanym procesem i pilotowym przebiegom innowacyjnych pomysłów i technologii.

Wśród praktyk ogólnych na poziomach 2-4 znajdują się między innymi: przypisywanie zasobów i odpowiedzialności za zadania, dokumentowanie procesu wytwarzania, dostarczanie odpowiednich narzędzi do wykonania zadań, zapewnienie szkoleń, planowanie procesu, używanie standardów i procedur, zarządzanie konfiguracją, sprawdzanie zgodności procesu z odpowiednimi planami i procedurami, audyt produktów, śledzenie procesu z wykorzystaniem pomiarów, wykonywanie akcji naprawczych i zarządzanie procesem w oparciu o dane.

## 5. Kultura jakości organizacji

Zainteresowanie kulturą organizacji ma związek z zauważeniem różnic w efektywności pracy zespołów pracujących w różnych krajach w takich samych warunkach zewnętrznych oraz w tym samym kraju w firmach o odmiennej specyfice. Stwierdzono, że odpowiedzialna za tę zmienność jest kultura społeczności lub organizacji. Wśród socjologów nie istnieje jedna powszechnie uznawana definicja kultury, ale mimo to uznaje się, że to właśnie kultura ma



wpływ na styl pracy i współpracy w firmie oraz, że organizacje o wyższej kulturze jakości utrzymują wysoką efektywność, produktywność i innowacyjność w dłuższym okresie czasu. Poniżej przedstawię kilka definicji kultury oraz wskażę na badania, które potwierdzają powyższe stwierdzenia.

Hofstede [13] postrzega kulturę jako zaprogramowanie umysłu w postaci wzorców myślenia, odczuwania i zachowania, które zostały nabyte w środowisku z okresu dzieciństwa. Stanowi ona środkowy poziom w trój-poziomowym modelu ludzkiego zaprogramowania umysłu zawierającego następujące elementy: natura - kultura - osobowość. Zewnętrznymi objawami kultury są praktyki: *symbole* - słowa, gesty, obrazy lub przedmioty, które są rozpoznawalne i mają znaczenie dla członków danej kultury, *bohaterowie* - to postacie współczesne lub historyczne, realne lub fikcyjne, które utożsamiają cechy szczególnie cenione w danej kulturze, oraz *rytuały* - zbiorowe działania, które są postrzegane jako społecznie niezbędne. Trzon kultury stanowią *wartości*, definiowane jako skłonność do wykonywania określonych wyborów. Turner [22] postrzega kulturę jako systemy symboli. Do systemów kulturowych zalicza język, technologię, wartości, przekonania, normy i zasoby wiedzy. Natomiast Schein [9] wyróżnia w kulturze trzy warstwy: najbardziej widoczna warstwa obejmuje technologię, sztukę i wzorce zachowań; druga warstwa - to wartości widoczne dla ludzi z większą świadomością społeczną; praktycznie niewidoczna trzecia warstwa, to wspólne założenia, zależności od środowiska i działań ludzkich oraz podejście do rzeczywistości i prawdy.

Kultura jakości to nie tylko zasady i struktury, ale także wspólne wartości, podejście do wytwarzania oprogramowania, jakości, współpracowników, celów itp. Eksperymentalne badania wpływu kultury jakości i etyki na efektywność i rezultaty pracy prowadził Bottorff [6]. Udostępnił w internecie ankietę w której każda organizacja mogła zdiagnozować swój poziom kultury. Następnie badał korelacje pomiędzy tym poziomem, a innymi aspektami działalności tych organizacji. Wykazał on szereg korzyści związanych z zarządzaniem etyką. Wśród nich wymienia: poprawę w pracy zespołu, wzrost zaufania, lepszą komunikację i współpracę, usuwanie źródeł błędów, poprawę morale pracowników oraz ich motywacji i zadowolenia, a także lepszą produktywność i wydajność. Natomiast autorka prowadziła poszukiwanie uniwersalnych wartości dla różnych kultur [5]. W wyniku otrzymała cechy, które chociaż są różnie nazywane (co jest częściowo spowodowane wpływem języka), to jednak można je podzielić na 3 grupy odpowiadające satysfakcji uczestników procesu, efektywności oraz jakości współpracy (ang. social aspect). Dodatkowe obserwacje potwierdzają, że łatwość komunikacji prowadzi do szybszego rozwiązywania pojawiających się problemów, a dobre samopoczucie i klimat pracy w zespole przyczynia się do lepszej jakości wyników.

## 6. Podsumowanie

W referacie została zaproponowana strategia zapewnienia jakości, którą można skonfigurować w zależności od specyfiki konkretnej firmy. Strategia określa cztery kluczowe obszary prac:

- Określanie definicji jakości oraz sposobu weryfikacji jej osiągnięcia, który to obszar zawiera metody dekompozycji pojęcia jakości na atrybuty i charakterystyki, a później określanie sposobów ich weryfikacji z udziałem odpowiednich udziałowców;
- Dobór technik dotyczących poprawy produktu, który zawiera techniki pozyskiwania wymagań, metody analizy i projektowania obiektowego, inspekcje i testy;
- Organizacja procesu wytwarzania i działania wspomagające, w którym można wykorzystać zalecenia z modeli dojrzałości oprogramowania (dla przykładu podany został schemat modelu SPICE);
- Podnoszenie kultury jakości organizacji, zgodnie z przekonaniem, że organizacje charakteryzujące się wyższą kulturą jakości oznaczają się także większą efektywnością i wykonują produkty o wyższej jakości.

Uzupełniającą częścią metody jest styl postępowania zgodny z filozofią TQM (zaangażowanie wszystkich pracowników w działania pro-jakościowe, satysfakcja klientów i pracowników, koncentracja na faktach oraz ciągłe doskonalenie) oraz metody eksperymentalne, które dzięki połączeniu metod ilościowych i jakościowych [3] mogą pomóc w identyfikacji problemów oraz poszukiwaniu rozwiązań. Czasami potrzebna jest integracja kilku technik pro-jakościowych [4,20]. W ramach dalszych prac planowane są eksperymenty stwierdzające, w jakich warunkach poszczególne techniki jakościowe są przydatne oraz prace nad kulturą jakości.

## Literatura

1. Basili V., Empirical Software Engineering, Journal of Empirical Software Engineering, (1) 1996.
2. Bobkowska A., Prognozowanie jakości oprogramowania na podstawie modeli UML., rozprawa doktorska, 2001.
3. Bobkowska A., Quantitative and qualitative methods in process improvement and product quality assessment., Proceedings of ESCOM 2001.
4. Bobkowska A., Cysewski G., Szejko St., Integration of Quality Assurance Techniques, Proceedings of ACS'2000 (Advanced Computer Systems).
5. Bobkowska A., Software Quality Management in Context of the Ethics of Values, Proceedings of ETHICOMP 2001.
6. Bottorff, D. L., Ethics and Culture Management, Internet <http://www.ethicsquality.com/>

7. Begger D., Bobkowska A., Cysewski G., Krawczyk H., Ludwiszewski B., Proficz J.: Atrybuty i charakterystyki jakości oprogramowania., Raport WETI Nr 84/98 grantu KBN 8T11C 043 12.
8. Capability Maturity Model for Software V1.1, Technical report CMU/SEI-93-TR-24, 1993.
9. Capability Maturity Model Integration (CMMI), V1.1, Technical report CMU/SEI-2002-TR-012, 2002.
10. Dahlgaard J.J., Kristensen K., Kanji G. K., Podstawy zarządzania jakością, PWN 2000.
11. Hofstede G., Kultury i organizacje, Polskie Wydawnictwo Ekonomiczne, 2000.
12. Gilb T., Graham D., Software Inspections, Addison Wesley, 1996.
13. Górski J. (red.), Inżynieria oprogramowania, MIKOM, 2000.
14. ISO, Odpis normy ISO 9000-3, internet: [www.prim.com.pl/normy/9000\\_3.htm](http://www.prim.com.pl/normy/9000_3.htm)
15. ISO/IEC, SPICE Software Process Assessment, working draft v1.0.
16. ISO, Wyciąg z normy międzynarodowej ISO 9001, internet: [www.prim.com.pl/normy/](http://www.prim.com.pl/normy/)
17. ISO/IEC 9126: Information Technology. Software Product Evaluation. Quality Characteristics and the guidelines for their use.
18. Iwasiewicz A., Zarządzanie jakością, PWN, 1999.
19. Kolman R. (red), Zarządzanie jakością., Wydawnictwo Politechniki Gdańskiej, 1996.
20. Quality assessment, internet: [http://wwwsel.iit.nrc.ca/6ICSQ/conference/Papers.html#Quality Assessment](http://wwwsel.iit.nrc.ca/6ICSQ/conference/Papers.html#Quality%20Assessment)
21. Svendsen F.N., European Trends in Software Quality: Some Challenges in The Next Years, Materiały konferencyjne KKIO 1999.
22. Turner J. H., Socjologia. Koncepcje i ich zastosowanie., Zysk i S-ka, 1994
23. Witryna poświęcona testowaniu oprogramowania, zapewnieniu jakości oraz zagadnieniom użyteczności, Internet: [www.bbj.com.pl](http://www.bbj.com.pl)

Dr inż. Anna Bobkowska  
 Katedra Zastosowań Informatyki  
 Politechnika Gdańska  
 Ul. Narutowicza 11/12  
 80-952 Gdańsk  
 e-mail: [annab@eti.pg.gda.pl](mailto:annab@eti.pg.gda.pl)



# ASPEKTY PROJEKTOWANIA STRON INTERNETOWYCH DLA DZIAŁALNOŚCI GOSPODARCZEJ

Witold CHMIELARZ

**Streszczenie:** Zasadniczym celem niniejszego referatu było przedstawienie i ustosunkowanie się do różnych aspektów projektowania stron internetowych dla zastosowań gospodarczych. W kolejnych częściach pracy – zgodnych z cyklem życia systemu informatycznego tej klasy - ujęto zasady rządzące ustalaniu strategii projektowania strony, analizy jej treści, tworzenia projektu graficznego i jego implementacji, ustalania strategii marketingowej i wreszcie eksploatacji. Najwięcej uwagi poświęcono tym cechom, które odróżniają projekt witryny od projektu tradycyjnego systemu informatycznego: analizie i umiejscowieniu treści na stronie i projektowi graficznemu. W końcowych partiach materiału zwrócono uwagę na kryteria akceptowalności tworzenia witryny oraz zasadnicze problemy występujące w tworzeniu strony i metody ich przewycięzania.

## Wstęp

Pomimo, jak się wydaje - przejściowego kryzysu lat 2000-2001, trwa szybki rozwój aplikacji internetowych służących obsłudze coraz większej ilości dziedzin gospodarczych: handlu, bankowości, giełdy, ubezpieczeń zintegrowanych coraz częściej z systemami informatycznymi przedsiębiorstw. Podstawowym narzędziem komunikacji z użytkownikiem w tych aplikacjach pozostaje szeroko pojęta witryna internetowa. Proces tworzenia witryny internetowej jest wbrew pozorom zjawiskiem bardzo złożonym, które nie może zachodzić w sposób przypadkowy. Przekonanie, że aplikacje gospodarcze w Internecie może tworzyć każdy uległo deprecjacji po roku 2001, gdy jako jedno z najistotniejszych źródeł kryzysu wskazywano brak profesjonalizmu w organizacji elektronicznego biznesu, którego nieodłączną składową jest budowa narzędzi komunikacji z użytkownikiem. Dlatego celem tego referatu jest próba specyfikacji czynników, które mogą wpłynąć na podniesienie jakości procesu kreacji strony internetowej, tworzonej dla zastosowań gospodarczych. Analiza literatury wskazuje, że tego rodzaju badania były, jak dotąd, prowadzone przeważnie dość jednostronnie, z punktu widzenia:

- czysto programistycznego – poradniki tworzenia stron internetowych<sup>1</sup>,
- technicznego – oprogramowania używanego w tym procesie oraz definiowania płaszczyzny jego działania<sup>2</sup>,
- sieciowego – dominującą stała się problematyka tworzenia sieci i ich współdziałania z Internetem<sup>3</sup>,

<sup>1</sup> patrz: Taylor D.: *HTML 4. Tworzenie stron www*, Wydawnictwo RM, Warszawa, 1998

<sup>2</sup> patrz: Siegel D.: *Tworzenie stron www*, Optimus Pascal, Multimedia, Warszawa, 2000

<sup>3</sup> por.: McAteer E., Shaw R.: *The Design of Multimedia Learning Programs*, University of Glasgow, Glasgow, 1995

- architektonicznego – wyróżniająca podstawowe kategorie strony, ich powiązania ze sobą i wpływ na użytkownika<sup>4</sup>,
- prezentacyjnego – zajmującego się wizualizacją treści na stronie i metodami przybliżenia ich użytkownikowi<sup>5</sup>.

Ciekawe i całościowe podejście do problemu tworzenia witryn internetowych prezentowane jest przez Autorów podchodzących do tego problemu z punktu widzenia cyklu życia<sup>6, 7</sup>. Według tego podejścia aby osiągnąć sukces w procesie tworzenia witryny trzeba przejść przez wszystkie typowe fazy cyklu życia systemu informatycznego. W każdej z nich można wyróżnić dopiero specyficzne cechy projektowania stron internetowych.

Zgodnie z tym podejściem wszystkie informacje zawarte w projekcie serwisu internetowego muszą być zaprezentowane w odpowiedni sposób i dotrzeć do ludzi o konkretnych zainteresowaniach. Proces przygotowania projektu oraz późniejsze jego wdrożenie powinny przejść przez co najmniej pięć faz:

- określenie strategii projektu witryny i analiza zagadnienia - zdefiniowanie celu, zakresu i ograniczeń tworzenia witryny, rozpoznanie jej elementów oraz zaplanowanie pracy nad witryną,
- projektowanie logicznej struktury informacji i ich graficznej prezentacji, oraz sprawdzenie ich współdziałania,
- implementacja i wdrożenie witryny,
- ustalenie strategii marketingowej witryny,
- eksploatacja, kontrola rozwoju, przekształcenia i utrzymanie witryny.

## 1. Określenie strategii projektu witryny i analiza zagadnienia

W pierwszej fazie należy przede wszystkim określić cel, jakiemu stworzenie strony ma służyć. Do najczęstszych powodów utworzenia strony internetowej należą<sup>8</sup>:

- możliwość prowadzenia sprzedaży,
- swobodny dostęp do dowolnych informacji,
- moda na reklamę w Internecie.

<sup>4</sup> por.: Turban E.: *Determinants of Successful Website Design: Relative importance and Recommendations for Effectiveness*, <http://www.eturban.csulb.edu>

<sup>5</sup> patrz: Turban E., Lee J., King D., Chung H.M.: *Electronic Commerce. A Managerial Perspective*, Pearson Education, London, Hongkong, NY, 2000

<sup>6</sup> Ziemia E.: *Wskazówki do metodyki projektowania i wdrażania witryn internetowych w: Systemy Wspomagania Organizacji SWO'2001* pod red. naukową J. Gołuchowskiego, H.Sroki, Wydawnictwo AE w Katowicach, Katowice, 2001

<sup>7</sup> Mazurek K.: *Rozwiązywanie problemów technicznych konstruowania i wdrażania „Portalu Regionalnego Mazowsze”*, WSISiZ, Warszawa, 2001

<sup>8</sup> Ellsworth J. H., Matthew V.: *The Internet Business Book*, John Wiley & Sons, New York, 1998

Zdarza się, że ma to być wyłącznie wizytówka firmy, która zawiera jedynie podstawowe dane, takie jak adresy, telefony. Jej zadaniem jest więc przekazanie informacji o istnieniu danej firmy i jej lokalizacji. Witryna może też mieć charakter dynamiczny. Wtedy jest stale aktualizowana. Najczęściej jest wpisana w kampanię reklamową firmy, stając się jednym z kanałów budowania świadomości marki lub produktu. Gdy cel, któremu ma służyć strona WWW jest już sprecyzowany, warto się zastanowić ile na jej stworzenie przeznaczyć środków. Ma to decydujące znaczenie w takich kwestiach, jak wielkość strony, wybór dostawcy internetowego, czy też zakres wykorzystania sieci. Dobór rodzaju reklamy zależy od stopnia ważności zakupu dla klienta oraz stopnia łatwości sprzedaży danego produktu przez Internet. Do wyboru są przeważnie trzy podstawowe formy<sup>9</sup>:

- budowa witryny profesjonalnej – o szerokim zakresie informacji i rozbudowanych użytecznych lub rozrywkowych elementach przyciągających klienta,
- mikrowitryny – atrakcyjnie wykonane strony, informujące o produktach i markach firmy, umieszczane na innych witrynach, tematycznie związane z reklamowanymi produktami,
- bannery – małe elementy graficzne, reklamujące konkretny produkt.

W sytuacji, gdy produkt może być w łatwy sposób sprzedawany i dostarczony przez Internet (lub przesyłką kurierską) – muzyka, oprogramowanie, gry, bilety – to opłaca się zainwestować w witrynę rozbudowaną, wspierającą cały ciąg sprzedaży produktu. Dla towarów, które muszą być dostarczane z magazynów, a ponadto np. zajmują wiele miejsca a ich sprzedaż jest okazjonalna (meble, sprzęt AGD, farby, ubrania itp.) powinno się stosować mikrowitryny. W sytuacji, gdy produkty należą do grupy powszechnego użytku, ale muszą być szybko dostarczone (żywność i napoje) - witryny internetowej nie opłaca się w ogóle utrzymywać.

Dostępne zasoby finansowe decydują też o tym, kto taką stronę nam stworzy. Może to być wyspecjalizowana firma zatrudniająca informatyków, grafików, specjalistów od marketingu, czyli ludzi, którym można zaufać zarówno w kwestiach technicznych, jak i kreatywnych. Czasem mogą to być operatorzy Internetu, którzy posiadają własne studio graficzne lub edycyjne, a czasem agencje reklamowe, które coraz częściej rozszerzają działalność na nowe medium, jakim jest sieć. Nie poleca się korzystania z osób i/lub firm, dla których działalność tego typu ma charakter uboczny.

Przed rozpoczęciem prac warto też przyjrzeć się stronom naszej konkurencji, starając się spojrzeć na nią okiem potencjalnego nabywcy. To pozwala na uświadomienie sobie powszechnych błędów oraz generowanie świetnych rozwiązań. Jak już zauważono wcześniej, istnieją trzy główne czynniki przyciągające klientów do strony WWW:

- dostęp do aktualizowanych, bieżących wiadomości,

---

<sup>9</sup> Dyba A.: *Jak skutecznie zaplanować strategię internetową*, <http://idm.supermedia.pl/iso/luneta/>



- możliwość rozrywki i kontaktu z innymi użytkownikami Internetu,
- dostęp do sklepu internetowego.

Uświadomienie sobie potrzeb klientów najlepiej służy ustalaniu zawartości merytorycznej stron. Według firmy Cavern<sup>10</sup> pomocne tu może być odpowiedzenie sobie na wstępie na trzy podstawowe pytania:

- jaki jest cel utrzymania strony w Internecie?
- do kogo ma dotrzeć reklama w Internecie?
- na co położyć nacisk przy kontaktach z klientami?

Trzy powyższe punkty stanowią klucz do sukcesu lub porażki firmy startującej w Internecie. Każda kombinacja odpowiedzi wymaga bowiem innego podejścia do konstrukcji strony WWW i metody tej promocji.

Tak więc, w pierwszej, wstępnej fazie powstawiania witryny, powinniśmy zdefiniować cele oraz zamierzenia jakie mają być realizowane przez nasze strony WWW oraz zaplanować ilość informacji jakie będziemy prezentować na stronach. Również w tej fazie powinien powstać budżet przedsięwzięcia, rozwiązanie sprawy interaktywności w prezentacji informacji i przygotowanie zaplecza technicznego, które zapewnią zadowolenie użytkowników korzystających z naszych stron.

Budżet przedsięwzięcia zależy od środków finansowych, jakie firma zamierza przeznaczyć na stworzenie i późniejsze utrzymanie i prowadzenie witryny decydując o takich kwestiach, jak:

- wielkość strony (liczba podstron),
- wybór dostawcy internetowego,
- tworzenie strony własnymi siłami, czy skorzystanie z coraz bardziej popularnego outsourcingu,
- zakres wykorzystanie Internetu do prowadzenia działalności i inne.

Jeżeli planowane jest nawiązanie współpracy z firmą zewnętrzną, trzeba jak najszybciej zebrać własnych specjalistów i zatrudnić ich przy współtworzeniu projektu.

W fazie strategicznej trzeba przygotować i zaplanować następujące zagadnienia:

- w jakim celu powstaje witryna, co ma przynieść firmie oraz swoim użytkownikom.
- dla jakiego odbiorcy przygotowywana jest witryna, i jakie informacje będą dla niego interesujące?
- jak będzie wyglądał zespół tworzący witrynę?
- kto będzie zarządzał przebiegiem powstawania witryny?
- kto będzie odpowiadał za przygotowanie treści merytorycznej prezentowanych na stronach, kto będzie je weryfikował?
- kto będzie łącznikiem z firmami zewnętrznymi, usługowymi?

<sup>10</sup> <http://www.cavern.pl/poradnik.htm>

- kto będzie pracował na stanowisku Web-mastera w ujęciu długoterminowym?
- oraz ustalenie wymagań technologicznych:
- jakie przeglądarki i systemy operacyjne będą wspierane przez twoje strony?
  - jakiej przepustowości łączami mogą dysponować odwiedzający nasze strony? (Brać pod uwagę trzeba zarówno wewnętrzną sieć firmy, jak i odwiedziny naszych stron z zewnątrz).
  - budowa stron HTML i użycie technologii Dynamicznego HTML-a (DHTML):  
JavaScript, VBScript (Visual Basic Script), Java applets, Cascade Style Sheets, dodatki do przeglądarek (Plug-ins, ActiveX controls), elementy stron przeznaczone dla danego typu przeglądarki (różnice w interpretacji znaczników HTML przez przeglądarki), dodatkowe elementy bezpieczeństwa odwiedzających oraz naszego serwisu.
  - w jaki sposób możliwy będzie kontakt z personelem obsługującym witrynę (Technical Support) – poprzez email, czy przez publiczne forum dyskusyjne,
  - decyzja o używaniu bazy danych - przygotowanie odpowiedniej infrastruktury serwerowej i oprogramowania. Przyjęcie odpowiedniej dla danej witryny obsługi klientów – czy wymagane będzie logowanie do systemu, czy będzie wyszukiwanie w systemie.
  - czy będą udostępniane pliki audio i video, czy będzie potrzebny dodatkowy serwer dla mediów strumieniowych.
  - wybór serwera WWW, (musi być zagwarantowane 24 godzinne wsparcie techniczne dla urządzeń zakupionych do obsługi serwisu), wybór systemu operacyjnego jaki będzie obsługiwał serwer WWW, jakiego typu łącze internetowe będzie stosowane i jakiego Usługodawcę Internetowego wybrać. Jak będzie prowadzona statystyka odwiedzin i analiza logów. Wsparcie dla CGI (Common Gateway Interface) oraz oprogramowanie i koordynacja pracy nad bazą danych. Rozplanowanie wykonania kopii zapasowych.
  - wybór odpowiedniej nazwy domeny, łatwo kojarzącą się z treścią lub firmą.

Zaplanowanie budżetu przedsięwzięcia to niełatwa sprawa. Pod uwagę trzeba wziąć przede wszystkim przewidywane profity oraz pensje pracowników firmy, planowane inwestycje technologiczne, szkolenia pracowników, pomoc techniczną, Utrzymanie i rozwój bazy danych, rozszerzanie i aktualizację informacji na stronach.

## 2. Projektowanie logicznej struktury informacji i ich graficznej prezentacji

Projektowanie struktury logicznej serwisu ma za zadanie zebranie i pogrupowanie wszystkich informacji jakie mają być w nim prezentowane. Pozwoli to stworzyć strukturę organizacyjną witryny i stworzenie prototypu nawigacji. Powstać także powinny propozycje graficznych struktur obowiązującego schematu nawigacji.

Dobrze zaprojektowana struktura daje na koniec gotowe spisy treści, szczegółowy opis zawartości stron, rozmieszczenie ilustracji względem tekstu pisanego, włączając w to miniatury do powiększenia. Dobrze widoczne staną się potrzebne elementy programistyczne, które trzeba będzie dopasować do przewidzianej struktury technicznej i bazodanowej. Wiadomo również będzie jakie przeglądarki i jakie wymagania będą postawione przed Serwerem WWW, jakiego typu łącza będą potrzebne i jakie przeglądarki internetowe będziemy uważać na najważniejsze dla naszego projektu.

Ważne jest również zaplanowanie wykonania i wdrożenia elementów systemu nawigacji.

Jedną z podstawowych zasad tworzenia strony WWW, to jej logiczny układ. Komunikacja na stronie powinna być jasna. Klient musi wiedzieć, w którym miejscu witryny się znajduje i jak w łatwy sposób powrócić/dotrzeć do określonej partii materiału lub informacji. W takim przypadku wskazane jest, aby odnośniki nawigacyjne były widoczne i łatwo identyfikowalne – w tym celu warto podstawowe linki umieścić w górnym prawym rogu, gdyż tam najczęściej pada nasz wzrok, a połączenia umożliwiające przejście do innych podstron – dobrze wyróżnione. Ważne jest, aby dzięki zastosowanej grafice strona była estetyczna, ale i funkcjonalna. Klient, który korzysta z Internetu potrzebuje informacji szybkiej i jeżeli “ozdobna” strona ładuje się powyżej kilkudziesięciu sekund, poszuka interesujących go danych pod innym adresem.

Warto przy tworzeniu strony pamiętać o specyfice medium, jakim jest Internet. Tutaj klient nie spotyka się ze sprzedawcą twarzą w twarz – trudno więc przekonać go o wiarygodności firmy i pozyskać jego zaufanie. Dlatego też witryna powinna zawierać wszelkie dane identyfikujące firmę – adresy – też e-mailowy, numery telefonów i faksów, nazwiska właścicieli, referencje. Informacje te – w kontekście dużej ilości podobnych ofert – mogą być czynnikiem decydującym o ostatecznym wyborze tej, a nie innej oferty<sup>11</sup>.

W sklepie wirtualnym klient nie może towaru dotknąć, powąchać, spróbować. Dlatego ważne jest umieszczenie jak najpełniejszej i profesjonalnej informacji o produkcie. Z drugiej strony nie powinna ona klienta przytłaczać. Rozwiązaniem jest logiczny układ witryny z możliwością przechodzenia od informacji ogólnych do najbardziej szczegółowych. Bardzo ważne jest, aby cena była podana przy towarze. Powinna to być cena końcowa, czyli taka jaką ma zapłacić klient (powinna uwzględniać koszty przesyłki, VAT, akcyzę, itp.). Cena w sieci powinna być niższa niż w sklepach tradycyjnych, aczkolwiek dla klienta często ważniejsze są inne aspekty zakupów on-line – przede wszystkim możliwość ich dokonania w domu, 24 godziny na dobę.

Klient musi łatwo zidentyfikować stronę i szybko do niej dotrzeć – dlatego też ważny jest jej adres. Powinien być krótki, prosty i łatwo kojarzyć się z firmą. Z punktu widzenia spodziewanych korzyści firma powinna zarejestrować swoją witrynę pod adresem domeny, czyli np.: [www.firma.pl](http://www.firma.pl). Jest to rozwiązanie najdroższe, ale gwarantuje największy stopień zapamiętania przez klientów. Inne

---

<sup>11</sup> por: Vassos T.: *Strategie marketingowe w Internecie*, Studio Emka, Warszawa, 1999



rodzaje adresów to: adres funkcjonalny (strona egzystuje pod adresem serwera WWW usługodawcy – niskie koszty, ale można domniemywać, że firma nie przywiązuje zbytnej wagi do działalności w sieci lub nie stać jej na profesjonalny adres) lub domena międzynarodowa (bez wskazania kraju pochodzenia firmy) z reguły stosowana przez międzynarodowe koncerny posiadające przedstawicielstwa na całym świecie.

Równie ważne jak utworzenie witryny jest jej utrzymanie i prowadzenie. Jest to narzędzie dynamiczne i należy pamiętać, żeby wszystkie jego elementy były na bieżąco uaktualniane. Dotyczy to przede wszystkim oferowanego asortymentu i cen.

Jeśli już firma stronę posiada, nie może zapomnieć o jej promocji. Ważne jest aby o niej informować za pośrednictwem mediów tradycyjnych (adres strony przy reklamie telewizyjnej, radiowej, prasowej, na wizytówkach pracowników). Warto też stronę zarejestrować w największych serwerach katalogujących zasoby internetowe, jak: Onet, Wirtualna Polska, czy Yahoo! Inną metodą promocji jest umieszczenie na najpopularniejszych stronach WWW bannerów – czyli graficznych odnośników do naszej strony. Inną – skuteczną i tanią formą promocją strony – jest wysyłanie przesyłek na konta poczty elektronicznej. Dostawcy usług elektronicznych często oferują usługę polegającą na rozsyłaniu – na wyraźne zlecenie klienta – listów do wszystkich użytkowników prowadzonych przez nich kont. Przekaz taki może zawierać tekst reklamowy, ale również odnośnik do strony WWW, na której zainteresowany ofertą może uzyskać więcej informacji. Dopiero tak rozreklamowana witryna może stanowić medium promocji firmy oraz kolejny kanał sprzedaży jej produktów.

Reasumując, najczęstsze rekomendacje dotyczące tworzenia stron internetowych dadzą się sprowadzić do następujących pięciu grup czynników<sup>12</sup>:

### ***Szybkość ładowania strony***

- zamieszczanie na stronach internetowych rozsądnych rozmiarów i proporcji: grafiki, tabel i tekstów, w celu minimalizacji czasu ich ściągania na ekran użytkowników.
- limitowanie animacji i środków multimedialnych. Rekomenduje się nie przeladowywanie stron dodatkowymi trikami dynamicznymi, a jeżeli już to np. wyłączanie animacji po 10 sekundach i pozwolenie użytkownikowi na spokojne przewijanie strony<sup>13</sup>.
- powszechne używanie zamiast dużych rysunków i zdjęć grafiki o rozmiarach paznokcia,
- dostarczanie użytkownikowi możliwości wczytywania tylko tekstu,
- monitorowanie niezawodności własnego serwera,

<sup>12</sup> Chmielarz W.: Handel elektroniczny nie tylko w gospodarce wirtualnej, Wydawnictwa Naukowe Wydziału Zarządzania UW, Warszawa, 2001.

<sup>13</sup> Haine D.: *Five most Serious Web Design Errors*, E-Business Magazine, marzec, 1998, <http://hpcc920.external.hp.com/Ebusiness/webdesign.html>

- technika renderingu – polegająca na ładowaniu najpierw tekstu – klient czyta tekst w czasie, gdy jako uzupełnienie, ładują się elementy graficzne.

### ***Zawartość merytoryczna***

- używanie jasno sformułowanego i zwięzłego tekstu o poprawnej gramatyce i pisowni, koncentrującego się na podaniu: nazwy firmy, logo, celu i zakresu działania, nagłówka informacji o produktach i usługach, ogłoszenia, listę najlepiej sprzedających się w ostatnim czasie produktów, nowości, pozdrowienia od szefa firmy, referencje lub linki do zadowolonych klientów, ogłoszenia o zatrudnieniu, ciekawostki – gry, rebusy, kreskówki itp., data ostatniej aktualizacji.
- umieszczenie, najlepiej na każdej stronie kontaktu (np. e-maila) do osoby ds. współpracy w firmie (opiekuna strony). Niektórzy teoretycy<sup>14</sup> proponują możliwość konsultacji w postaci dyskusji on-line, chat itp.
- stosowanie stonowanych kolorów i wzorców tła – bardzo jasnego tła i ciemnych liter, najlepiej białego tła i czarnych liter (bardzo dobrze drukują się na czarno-białej drukarce), lub bardzo ciemnego tła i jasnych liter,
- dostarczanie bezpłatnych informacji i usług – zakłada się, że na poważnej stronie internetowej tylko 10% zawartości powinno być umieszczane po to by przynosić zysk.
- dokonywanie regularnych zmian na stronie – ponieważ klienci chcą oglądać świeży materiał, a częste odświeżanie zapewnia aktualność informacji na stronie,
- ograniczanie czynności rejestracyjnych dla potencjalnych klientów do niezbędnych informacji – unikanie pytań o np. liczbę dzieci, warunki zdrowotne itp.,
- umieszczanie dowcipów – tylko w sytuacji gdy będą okresowo odświeżane,
- unikanie komunikatów – strona w budowie, przebudowie itp.

### ***Łatwość nawigacji na stronie***

- używanie dobrze opisanych i zaetykietowanych odwołań, odpowiadających na dwa pytania: *co uzyskam klikając w odwołanie?* i *dlaczego chciałbym się tam dostać?*,
- unikanie używania ramek – ponieważ np. dzielą ekran na okienka wymagające poziome lub pionowe poruszanie się po stronie, brzydko wyglądają, często nie chcą się drukować czy prawidłowo wyświetlać na ekranie, niektóre przeglądarki nie chcą pokazywać ramek,
- utrzymywanie spójności systemu nawigacji – czyli najlepiej jednego, ale dobrego systemu,
- dostarczanie klientowi efektywnych przeszukiwarek – najlepiej wyrafinowanych i łatwych w użyciu

---

<sup>14</sup> Maloff J.: *Growing Business at Net Speed*, Internet World, lipiec 1997, str. 68-70.

- używanie wyróżniających się gorących klawiszy – kleksy, efekty trójwymiarowe, efekty tekstury itp.
- używanie raczej długich stron, niż krótkich z licznymi odwołaniami do podstron,
- prezentacja przewodników i map strony w przypadku budowania skomplikowanych stron WWW,
- unikanie linków otwierających nowe okienka.

### ***Bezpieczeństwo***

- zastosowanie na stronie systemów z protokołami bezpieczeństwa, co najmniej na poziomie technologii SSL lub SET. Wymienia się<sup>15</sup> osiem głównych bezpiecznych systemów operacji na kartach: CyberCash, CyberSource, IC Verify, Open Market, Verifone, VeriSign, IBM Internet Division i ClearCommerce,
- staranie się o zachowanie praw autorskich wszystkiego co jest nowe, oryginalne i publikowane na stronie,
- generowanie strony potwierdzającej ewentualny wybór konsumenta tak, aby klient był pewien dokonanego wyboru.

### ***Aspekty marketingowe – nastawienie na klienta***

- wypromowanie swojej strony: ułożenie listy słów kluczowych, tak by wyszukiwarki łatwo je odnajdywały, zaaranżowanie wymiany linków z innymi stronami biznesowymi, włączenie reklamy strony do wszystkich tradycyjnych mediów reklamowych firmy, zorganizowanie konferencji prasowej, rozesłanie wiadomości do klientów, zorganizowanie bannerów reklamowych, reklamowanie strony w radio, telewizji, nie stosowanie spamu, itd.
- odkrywanie potrzeb konsumenta – używanie ciasteczek (cookie), nadawanie tymczasowych identyfikatorów, losowo generowanych numerów kart, itp.,
- dostarczenie użytkownikowi zajęcia dodatkowego w postaci gier, zagadek itp.,
- dostarczenie tak wielu sposobów płatności, jak to jest możliwe,
- nazywanie domeny tak by była łatwa do zapamiętania,
- używanie automatycznego generowania odpowiedzi,
- sprzedawanie nadmiarowej informacji na zewnątrz poprzez mikropłatności,
- unikanie stosowania widocznych liczników odwiedzin,
- umieszczanie wewnętrznych odwołań przed wysyłaniem klientów na inne strony,
- stosowanie powszechnie zachęt do dokonania operacji na stronie np. kuponów, zniżek, upominków reklamowych itp.,
- opisywanie strony w języku angielskim, bądź w kilku wersjach językowych.

---

<sup>15</sup> Sweet L.: *Ringing the E-Commerce Cash Register*, Internet Computing, marzec, 1998, str. 57-68.



Obecnie, konstruowanie witryn internetowych uważa się za przedsięwzięcie wymagające kombinowanej znajomości psychofizjologii, sztuki malarstwa i technologii informacyjnych. Wynikają z nich szczegółowe wskazówki dotyczące projektowania grafiki i tekstu na ekranie strony<sup>16</sup>:

- przeglądamy ekran od góry do dołu i od lewej do prawej, dobrze jest układać elementy na ekranie w taki sposób, aby oczy odbiorcy poruszały się w pożądanym kierunku,
- preferencje postrzegania są następujące:
  - większe obrazy przed mniejszymi,
  - obiekty poruszające się przed statycznymi,
  - niesymetryczne przed symetrycznym,
  - kolorów nasyconych przed nienasyconymi,
  - kolorów żywych przed matowymi,
  - ilustracji czy rysunków kolorowych przed czarno-białymi.
- ekran dzielony jest na obszary funkcjonalne, mogące się powiększać lub kurczyć, ale utrzymujące swoje relatywne położenie,
- jeśli wyświetla się informacje pojawiające się na wielu stronach powinny mieć standardowy kształt, rozmiar, kolor i położenie, co jest ważne dla płynności pracy,
- trzeba pamiętać by nie przeładowywać ekranu, idealny jest ekran na którym ważne elementy są wyeksponowane, a pozostałe pozwalające odbiorcy na dostęp do dalszych funkcji można używać niemal instynktownie,
- badania nad przystosowaniem oka, czasem czytania, zrozumieniem i czytelnością sugerują, że dla czytania ciągłego najlepiej używać czcionki slab-serif lub sans-serif,
- tekst podstawowy powinien być odróżniony krojem, stylem, rozmiarem i kolorem od innych tekstów, należy unikać zbyt wielkiej różnorodności,
- duże ilości tekstu są trudne do czytania i zrozumienia, dzieli się więc tekst na kolumny lub redaguje w taki sposób by czytelnik mógł objąć kilka wyrazów za jednym razem,
- istnieje optymalna wielkość czcionki i odstępów pomiędzy liniami, które ułatwiają czytanie,
- zaleca się dosuwanie tekstu do lewej strony,
- należy łączyć wielkie i małe litery, stosowanie samych wielkich liter, zwiększa czas czytania o ponad 10%,
- kolory tekstu i tła powinny być dobierane bardzo ostrożnie, za najlepsze uważa się kombinacje kolorów:
  - czarny na białym
  - żółty na szarym,
  - biały na granatowym,
  - ciemnoniebieski na jasnoszarym,
  - czarny na jasnoszarym.

---

<sup>16</sup> Por.: Klus J.: *Ewolucja reklamy w Internecie*, WZ UW, Warszawa, 2001,

- należy ograniczyć liczbę kolorów na stronie do co najwyżej czterech,
- informacja przedstawiana graficznie jest zauważana i pamiętana szybciej niż tekstowa,
- ruch przyciąga uwagę, ale inne ważne informacje mogą być przez to zignorowane,
- sekwencje wideo wymagają dobrego sprzętu i stałego łącza, mogą więc być przeznaczane tylko dla wybranej grupy użytkowników,
- około 80% informacji absorbujemy przez medium wzroku, pozostałe ze smaku, dotyku, zapachu i słuchu. Z tego wynika, że dźwięk póki co jest tylko dodatkiem do informacji wizualnych, używany dla zwrócenia uwagi, uatrakcyjnienia strony lub wzmocnienia informacji,
- na stronach nie należy stosować kolorów w nadmiarze, najlepiej stosować je tylko w pewnych odcieniach, po to by zachowały jasność i przejrzystość,
- z powodu różnic dyfrakcji między długością fal różnych kolorów a fizyczną strukturą oka, występują różnice które należy w trakcie projektowania strony wziąć pod uwagę:
  - jeśli grupa równoważnych kolorów będzie położona w tej samej odległości, czerwony będzie postrzegany bliżej niż zielony, zielony bliżej niż niebieski,
  - kolory podstawowe: czerwony, zielony, niebieski nasycone bardziej skupiają wzrok niż nienasycone,
  - odcienie niebieskiego są trudne do rozróżnienia,
  - kolory nasycone w połączeniu z jasnymi bardziej przyciągają uwagę, niż ciemne nienasycone,
- podczas używania koloru do zaakcentowania pewnych partii lub ukazania zależności pomiędzy informacjami, zaleca się używanie (w naszej kulturze) takich sposobów kolorowania jak: żółty dla ostrzeżenia, czerwony dla zatrzymania, zielony do startowania lub wykonywania manewru oraz konsekwentnie je używać w całej witrynie,
- dobrze jest stosować sprzężenie zwrotne pomiędzy: twórcami witryny (negocjacje i konsultacje wewnętrzne w grupie projektowej lub całej firmie), jak również twórcami witryny a klientami (zewnętrzne np. w postaci przedstawienia do wypełnienia użytkownikowi skali oceny strony) żeby nie popaść w samozadowolenie i modyfikować projekt w kierunku najbardziej odpowiadającemu klientowi.

Jak można zauważyć część tych porad szczegółowych była już w jakiś sposób uwzględniana w zaleceniach ogólnych tworzenia witryn.

Wszelkie witryny internetowe odbiegają od tradycyjnych publikacji papierowych. Strony publikowane w Internecie mogą być traktowane oddzielnie, co umożliwia swobodne reguły linkowania stron. Jest to powodem, dla którego wymogi formalne stawiane przed oddzielnymi plikami HTML tworzącymi witrynę są zawsze wyższe. Z tego wynikają następujące reguły:

- każda strona musi więc zawierać duży i łatwo widoczny tytuł strony (również na pasku tytułu przeglądarki). Często, na stronach o dużej ilości

grafiki jest to pierwsza rzecz jaką użytkownik widzi przez dłuższy czas. Tytuł także jest dodawany jako opis zakładki w przeglądarce, przy czym powinien być na tyle czytelny by łatwo było zidentyfikować tekst strony,

- dokument musi zawierać wyraźnie autora danego opracowania,
- na stronie powinna być też umieszczona data, albo nawet czas publikacji tekstu (teksty czasami są rzadko aktualizowane). Użytkownicy szukają zawsze najbardziej aktualnych wersji opracowań, a aktualizacje mogą nie zmieniać wyglądu strony na tyle, by na pierwszy rzut oka można określić, że strona uległa zmianie,
- istotne jest też z jakiego źródła pochodzi dokument, co często wiąże się z osobą, która go przygotowała. Najprostszym sposobem aby to osiągnąć jest dodawanie nazwy i adresu URL witryny na stronie,
- link powrotny do strony początkowej witryny powinien być dobrze widoczny.

Podczas tworzenia interfejsu użytkownika na stronie należy założyć możliwość niemal intuicyjnego poruszania się po przedstawionych danych. Bardzo źle jest przyjmowane zmuszanie użytkownika nawigacji, która jest wynikiem wybujałej fantazji projektanta, a ponadto np. powoduje długotrwałe wczytywanie strony. Trzeba zaraz na początku zdefiniować jaką strukturę dokumentów proponujemy użytkownikowi i konsekwentnie ją stosować. W każdej strukturze powinno być kilka kluczowych punktów nawigacji, w których użytkownik będzie informowany o dalszych możliwościach nawigacji.

Użytkownicy zawsze chcą osiągnąć informacje w jak najkrótszym czasie. Uważa się, że najłatwiejsze do zapamiętania i najszybsze do przejścia jest menu składające się z pięciu do siedmiu pozycji. Lepiej jest konstruować krótkie, konkretne menu rozłożone na paru (dwóch, trzech) stronach, niż rozległe menu uproszczone, ale zawarte na kilkunastu (10-15) podstronach. Najlepiej jest umieszczać na każdej stronie sprowadzonego do granic możliwych dla porównywalności układu poszczególnych elementów (przynajmniej nagłówka i stopki zawierających najważniejsze elementy nawigacji). Przyzwyczajai to użytkownika do rozkładu informacji na stronie i w ten sposób będzie on zawsze wiedział gdzie znaleźć pomoc i inne informacje.

Strukturę informacji na stronie można podzielić na cztery najważniejsze układy:

- sekwencyjny,
- macierzowy,
- hierarchiczny,
- sieciowy.

Struktura sekwencyjna jest bardzo prosta, o podstawowej złożoności i skierowana głównie do ludzi uczących się ze stron internetowych. Z tego powodu nadaje się do tworzenia samouczków, czy też poradników. Użytkownik z założenia powinien przejść i zobaczyć cały zestaw stron połączonych ze sobą.



Struktura macierzowa wymaga od widza pewnej znajomości tematu i swobody poruszania się w danej dziedzinie. Nadaje się do prezentowania informacji sformułowanej jako np. linie czasowe pewnych wydarzeń, niezależnych od siebie, ale jednocześnie powiązanych datami i osobami. W ten sposób można dobrze prezentować dane typu encyklopedycznego. W celu ułatwieni nawigacji użytkownikowi mapy serwisu z listą kluczowych linków na stronie.

Struktura hierarchiczna najlepiej nadaje się do budowania stron z dużą ilością złożonych informacji. Większość dzisiejszych witryn koncentruje się wokół strony domowej (główniej), a ich struktura przypomina bardzo struktury organizacyjne, czy funkcjonalne różnych instytucji. Z tego powodu jest strukturą najbardziej odpowiadającą przeciętnemu odwiedzającemu. Wymaga jednak bardzo dokładnego i przestrzegania podziału informacji.

Struktura sieciowa (hipertekstowa), budowana jest na zasadzie przewidywania przez projektantów i symulowania zachowań i toku myślenia odwiedzających. Struktura ta jest najbardziej otwartą ze wszystkich opisanych. Podstawową zasadą działania tych stron jest jak największa ilość połączeń wewnętrznych i zewnętrznych. Co prawda taka organizacja najlepiej uwypukla wszystkie korzyści jakie dają nam łącza hipertekstowe, ale powoduje również największe problemy z odnalezieniem właściwych informacji przez użytkowników. Przeznaczona jest dla stron, posiadających wiele łączy do innych oraz tych użytkowników, którzy są bardzo doświadczeni w poruszaniu się w Internecie i zarazem doskonale znają tematykę witryny, szukając tylko specjalistyczne informacje.

Najważniejszą częścią każdej witryny, najczęściej odwiedzaną ze stron jest strona domowa (główna). Jest ona widoczna zaraz po wczytaniu strony i dlatego w musimy dobrze rozplanować jej wykorzystanie. Przede wszystkim powinna znajdować się na niej dokładna i czytelna informacja o naszej firmie i naszych propozycjach dla klienta. A początku rozwoju Internetu najczęściej spotkać można było strony domowe funkcjonujące jako zaproszenie do przeglądania zawartości witryny z dużą ilością linków. Rozwój przeglądarek spowodował, że linki tekstowe zastępowane były kompleksowymi rozwiązaniami graficznymi, oferującymi bardziej optymalne wykorzystanie przestrzeni. W ostatnim czasie coraz popularniejsze stają się strony domowe, zawierające nie tylko nawigację i zbiór linków, ale także najważniejsze i najświeższe informacje z danej dziedziny. Częste aktualizacje zapewniają witrynie częste powroty czytelników. Bardzo doceniane jest też umieszczanie na stronach krótkich ciekawostek, kalendarium, czy imienin, byle tylko nie czynić tego w nadmiarze. Niekiedy, by ułatwić czytelnikowi znalezienie informacji, na stronie domowej dokonuje się podziału materiału na grupy przeznaczone dla konkretnych, a różnych użytkowników. Takie rozwiązanie stosować muszą np. strony uniwersyteckie, które od razu na pierwszej stronie mają podział informacji na przeznaczone dla już studiujących, dla przyszłych studentów oraz pracowników uczelni. W tym przypadku zbyt ogólne tytuły oraz wiadomości dla studentów nie sprzyjałoby dokonywaniu właściwych wyborów. Z kolei

bezsensowne krążenie po serwisie mogłoby spowodować zniecierpliwienie studentów oraz wywołać efekt sztucznego, a dużego natężenia ruchu.

Ta część firm, którym strona służy właściwie tylko za wizytówkę i prezentacja informacji schodzi na drugi bądź dalszy plan, tworzą strony początkowe zawierające tylko i wyłącznie elementy graficzne, mające na celu przyciągnięcie wzroku i prowokację użytkownika do wyszukania linku do dalszych informacji. Dla przeciętnego użytkownika stanowi to w dłuższym okresie czasu przeszkodę w dotarciu do informacji. Dzieje się tak zwłaszcza, gdy dysponujący tylko słabym połączeniem modemowym czytelnik musi czekać na załadowanie całej strony wypełnionej grafiką.

Informacja pokazywana na stronach powinna być łatwa w odbiorze i czytelna. W odróżnieniu od tradycyjnych, papierowych dokumentów drukowanych nie ma ograniczeń długości stron internetowych. Informacje prezentowane na nich mogłyby teoretycznie ciągnąć się w nieskończoność, co nie byłoby jednak zbyt dobrze odbierane przez czytelników i kompletnie nieczytelne.

Wraz z pierwszym spojrzeniem klienta na stronę determinowana jest jego świadomość występowania elementów strony i sposobu nawigacji. Pierwsze co rozpoznaje to układ strony: bloki na niej występujące, tytuł, nagłówek i stopka, a następnie dopiero tekst zawierający interesującą go informację. Dlatego bardzo istotna jest kolorystyka i gra światłocienia na stronie. Dla zastosowań komercyjnych ważne jest również odpowiednie umieszczenie informacji o firmie. Najlepiej by było gdyby logo firmy umieszczono u góry strony wraz z jej nazwą i tytułem. Ponadto w projekcie liczy się także zrównoważenie układu grafiki i tekstu. Wydaje się, że internauta musi mieć przede wszystkim jasność co o tego do jakiego tekstu odnosi się dana ilustracja lub jaki fragment tekstu ma zastępować.

Tytuły i zdania, czy akapity powinny raczej być ustawiane do lewego marginesu. Należy unikać pisania tytułów kapitalikami – ich rozpoznanie wymaga dużo więcej uwagi niż tekstu rozpoczynanego z wielkiej litery, ale za to większego.

Proces formatowania tekstu jest żmudny i pracochłonny ze względu na różnice występujące w interpretacji podczas przeglądania go w różnych przeglądarkach. Z tego powodu w poradnikach budowy stron zaleca się definiowanie wielkości liter w pikselach a nie w punktach, gdyż każda przeglądarka i system operacyjny trochę inaczej interpretuje punkty. Najlepsze rezultaty uzyskuje się przeglądając stronę w różnych przeglądarkach i najlepiej pod różnymi systemami.

W czasie konstrukcji stron internetowych problematyczny bywa również dobór formatu plików graficznych. W Internecie najpopularniejszymi standardami zapisu grafiki są formaty GIF89a (Graphics Interchange Format) oraz JPEG (Joint Photographic Experts Group).

Obrazki w formacie JPEG nadają się świetnie na zapisywanie dużych, wielokolorowych zdjęć i ilustracji. Format GIF z uwagi na bezstratną kompresję i wierne odwzorowanie kolorów nadają się świetnie do tworzenia interaktywnych menu na stronach.

W analizowanej fazie projekt witryny zaczyna nabierać już konkretnego kształtu i wyglądu. Wszystkie elementy graficzne powinny być już gotowe;



przygotowane ilustracje, gotowa wspólna nawigacja, gotowe formatki/szablony stron, poprawione i sczytane teksty; działające skrypty Javascript i Java applets, stworzona struktura bazy danych, oraz przykładowe strony interaktywne dla użytkowników, gotowy i przetestowany mechanizm wyszukiwania.

Kwestią najważniejszą dla każdego użytkownika jest nawigacja na stronie. Jak już wcześniej wspomniano zależy ona od przyjętej architektury projektu witryny i powinna być różna dla różnych użytkowników.

W zależności od użytkownika witryna musi przede wszystkim pomagać mu w znajdowaniu informacji. Część witryn np. budowie sekwencyjnej powinna mieć jasno i wyraźnie zaznaczone jak można przejść do następnej i poprzedniej części strony. Mogą do tego posłużyć prozaiczne przyciski graficzne umieszczone w stopkach i nagłówkach stron. Nie można zapominać o łatwym powrocie do strony głównej, czy spisu treści. Czyni to witrynę czytelną i łatwą w odbiorze.

Głównym problemem w systemie macierzowym jest odpowiednie rozłożenie prezentowanych treści. Strony w tym układzie powinny posiadać czytelne informacje o tym, w którym miejscu użytkownik w danej chwili się znajduje. W łatwy sposób powinny umożliwiać przenoszenie go do podobnych informacji, ewentualnie na podobnych stronach. Na takich witrynach potrzebny jest dobra struktura strony i zaplanowane w niej miejsce na menu podstawowe i linki dodatkowe. Ze względu na dotychczasowe przyzwyczajenia użytkowników najlepiej stosować tu model klasycznego rozkładu strony. W modelu tym po lewej stronie umieszczone, znajdują się propozycje zobaczenia dodatkowych informacji związanych z informacjami z właśnie oglądanej strony. Pomaga w tym również zamieszczenie kompletnej mapy serwisu, z odnośnikami do wszystkich ważniejszych części publikacji.

Nie wydaje się aby trudnym zadaniem było prawidłowe zaprojektowanie interfejsów w lubianej przez użytkowników strukturze hierarchicznej. Należy dbać tylko o logikę rozwijania struktury, ograniczenie ilości zagłębień oraz umieszczenie w widocznym miejscu przycisków powrotu do poprzedniej i głównej strony. Jest on uważany za najbardziej rozpowszechniony na stronach internetowych ponieważ pozawala na łagodne przejście między tematami i połączenie wszystkiego w jedną logiczną całość poprzez stronę główną lub stworzony spis treści.

Najtrudniejszym do zbudowania jest projekt nawigacji dla struktury sieciowej (hipertekstowej). W zasadzie struktura ta charakteryzuje się dużą dowolnością połączeń między dokumentami. Tylko więc od projektantów zależy sposób poruszania się po nieograniczonej ilości połączeń między stronami. Decydującą sprawą jest tutaj przewidzenie toku myślenia przyszłego użytkownika i pobierania przez niego informacji ze stron oraz takie eksponowanie linków, żeby nie poczuł się zagubiony na stronie w takcie nawigacji.

Dla wszystkich rodzajów witryn pomocnym elementem w nawigacji mogą być dobrze skomponowane, a wspólne dla wszystkich stron, nagłówki i stopki. Powinny one zawierać podział na takie kategorie jakie znalazły się w ostatnim punkcie nawigacyjnym, albo też zachowany schemat organizacji połączeń ze strony głównej.



Teraz, gdy mamy już gotowe: projekt graficzny, zamieszczone wszystkie informacje, załączone ilustracje i zakończone programowanie elementów interakcyjnych, przychodzi czas na wstępne testy naszej strony. Najlepszymi do tego celu osobami są użytkownicy z zewnątrz, które wcześniej nie miały styczności z materiałami dotyczącymi naszej witryny. Powinni oni zwracać uwagę na wszystko – począwszy od błędów językowych i literówek na błędnym prezentowaniu informacji, linkowaniu stron i działaniu wyszukiwania kończąc. Jest ostatnia faza przed promowaniem stron.

### 3. Implementacja i wdrożenie witryny

Dotarcie do konkretnej informacji i znalezienie konkretnej strony stało się głównym problemem w zdecentralizowanej sieci. Duża ilość komputerów pomnożyła ilość danych. Brak komputera centralnego oznacza, że nie ma żadnego katalogu głównego, do którego można by się odwołać poszukując informacji – są one rozrzucone po całej sieci Internetu. Powstała więc konieczność wprowadzenia powszechnego mechanizmu jednoznacznie identyfikującego informacje w przestrzeni danych. Tym mechanizmem jest URL (*Uniform Resource Locator*) – unikatowy deskryptor, który może identyfikować dowolny dokument (zwykły lub hipertekstowy), grafikę, menu, artykuł w Usenecie, komputer, a nawet archiwum plików w dowolnym miejscu Internetu.

Do poszukiwania stron WWW używany jest specjalny sposób komunikacji tzw. HTTP (*HyperText Transfer Protocol*), czyli protokół transferu hipertekstu. Jest to metoda dokonywania transferu dokumentów WWW.

Deskryptor URL oraz usługa HTTP ułatwiają docieranie do poszukiwanych informacji. Okazuje się jednak, że to nie wystarczy. Liczba dokumentów WWW w Internecie jest ogromna – rozrzucone są one na serwerach całego świata. Aby uniknąć powielania się nazw dokumentów WWW w Sieci wprowadzono DNS – system nazw domenowych. Adres do strony WWW lub sklepu internetowego powinien być jak najkrótszy – łatwy do zapamiętania. Krótki adres zajmuje mało miejsca na (i tak małych) bannerach. Reklama internetowa często wspierana jest przez inne media np. outdoor, prasę – tu krótki format adresu ma jeszcze większe znaczenie.

Projektowane dokumenty prezentują się w inny sposób w różnych przeglądarkach internetowych. Wynika to przede wszystkim z odmiennego interpretowania kodu HTML. Strona zaprojektowana pod przeglądarkę Netscape Navigator może być inaczej odczytana przez Internet Explorera i odwrotnie. Na rynku przeglądarek internetowych najmocniejszą pozycję ma wspomniany już Internet Explorer, drugie miejsce zajmuje Netscape Navigator. Tak więc zanim utworzoną stronę WWW umieści się w Sieci, trzeba ją przetestować w co najmniej dwóch wyżej wymienionych przeglądarkach internetowych.

Strony WWW są tworzone w języku HTML - hipertekstowym języku adjustacji. Jest on zbiorem kodów osadzonych w tekście służących za elementy

formatujące lub stanowiące informację łączącą. HTML oparty jest na języku SGML (*Standard Generalizer Markup Language*).

Do utworzenia dokumentu HTML wystarczy zwykły edytor tekstów. Może to być, przynajmniej teoretycznie, najprostszy edytor dla DOS, mający funkcję zapisania dokumentu w postaci tekstowej, choćby Norton Editor, PC Write. W praktyce najczęściej używa się Notatnika lub programu Word Pad.

Na ogół w dokumentach HTML – tak jak w każdym dokumencie tekstowym – występuje podział tekstu na: nagłówki, tekst główny, odnośniki itd.

Odrębną kwestią jest wprowadzanie na stronę elementów graficznych. Jak powiedziano, najpopularniejsze formaty graficzne wykorzystywane na stronach WWW to gif i jpg. Format gif zajmuje mniej pamięci, ale ma niższą jakość. Pliki graficzne jpg są wyższej jakości, zajmują za to więcej pamięci. Ważne jest rozsądne rozmieszczanie grafiki i dobór formatów – tak aby strona WWW ładowała się szybko i wyglądała dobrze. Pliki graficzne na stronie umieszczać można zarówno jako tło, jak i jako ilustracje. Na stronach WWW można umieszczać również wewnętrzne pliki video – zajmują one jednak dużo pamięci. Ostatnim elementem multimedialnym na stronie jest dźwięk. Zanim wprowadzono format MP3 i technologię Real Audio dźwięku w Internecie prawie w ogóle nie było. Wynika to z faktu, że pliki dźwiękowe zajmują dużo pamięci, a co za tym idzie – ściąganie ich zajmuje dużo czasu. Wymienia się następujące narzędzia tworzenia stron www:

### *Java Script*

Java Script jest najpowszechniejszym rozszerzeniem kodu HTML. Java Script to prosty język programowania – język skryptowy – służący do definiowania zachowania się elementów na stronie www.

### *Visual Basic Script*

VBS Script dla Internet Explorera jest rozwiązaniem alternatywnym do języka Java Script. Visual Basic Script nawiązuje do znanego niemalże wszystkim prostego języka Basic. Aplety VBS Script umieszcza się w podobny sposób jak aplety Java Script. Największą wadą VBS Scriptu jest to, że obsługiwany jest jedynie przez Internet Explorera. Z tego powodu nie jest on zbyt często stosowany. Większość twórców stron WWW dba o to, aby wyglądały one dobrze w jak największej ilości przeglądarek internetowych – nie tylko w jednej.

### *Java*

Wbrew pozorom Java nie ma nic wspólnego z Java Script. Są to dwa różne języki programowania. Java to skomplikowany język obiektowy, oparty na języku C++, który sam jest zmodyfikowaną wersją języka C. Język C został stworzony z myślą o pisaniu programów obsługi urządzeń UNIX-a, ma więc wiele wspólnego z najbardziej elementarnym z języków: assemblerem. Do osadzania apletów Javy w dokumentach HTML służy komenda <OBJECT>. Zaletą Javy jest to, że obsługiwany jest zarówno przez Internet Explorera i Netscape Navigatora.

## **ActiveX**

Tak jak VBS Script jest alternatywą Microsoftu dla Java Scriptu, tak ActiveX ma być alternatywą dla Javy. Możliwości języka Activex są podobne do Javy – z jedną różnicą – nie obsługuje go Netscape Navigator<sup>17</sup>.

## **Macromedia Flash**

Program Flash4 firmy Macromedia obecnie staje się coraz bardziej popularny. Atrakcyjność programu Flash polega na tym, że umożliwia on płynne połączenia animacji (opartej na grafice wektorowej) oraz dźwięku – z wykorzystaniem formatu MP3. Macromedia Flash ma duże możliwości optymalizacji i kompresji, pozwala na umieszczanie pluginów Flasha na stronach WWW, a także tworzenie całych stron WWW opartych na technologii Flash.

## **Edytory stron WWW**

Dokument HTML można zredagować w najprostszym edytorze tekstów, wymagałoby to jednak ręcznego wprowadzania wszystkich kodów, co jest dość uciążliwe i naraża na ryzyko popełniania błędów składniowych. Można więc skorzystać z bogatej oferty wyspecjalizowanych edytorów HTML.

Edytory HTML można podzielić umownie na trzy klasy:

- edytory tekstowe,
- edytory graficzne,
- edytory „klockowe”.

Przykładem tego ostatniego jest komercyjny program Home Page Author, wchodzący w skład pakietu Internet Mania, firmy Corel, który pozwala wstawiać do dokumentu bloki informacji. Ostateczny efekt widzimy w przeglądarce, nie mamy natomiast bezpośrednio do czynienia z samym dokumentem HTML. Jest to wygodny program lecz nie nadaje się do tworzenia skomplikowanych stron WWW.

## **4. Ustalenie strategii marketingowej witryny**

Witrynę internetowa można promować w bardzo różny sposób zarówno tradycyjny jak i internetowy. W sposób tradycyjny - począwszy od drukowanych reklam w czasopiśmie przeznaczonych do konkretnej grupy odbiorców, gazetach codziennych, listach i kartkach pocztowych informujących o powstaniu nowej firmy (a więc i strony) a na ulotkach kończąc. Często też możemy spotkać reklamy telewizyjne i radiowe witryn internetowych lub informację, że film czy słuchowisko mamy udostępnione dzięki sponsoringowi sklepu i skojarzonej z nim witryny internetowej. Firmy drukują swój adres internetowy na materiałach firmowych, kartkach pocztowych, długopisach. Są to małe rzeczy, ale mogą być używane przez dłuższy czas i informować przypadkowych odbiorców o adresie

---

<sup>17</sup> Taylor D.: "Tworzenie stron WWW"; Wydawnictwo RM, 1998.



strony w Internecie. Czasami takie reklamówki rozdaje się na różnego rodzaju targach, gdzie przychodzą docelowi klienci. Obowiązkowe jest także wydrukowanie odpowiedniego URL na wizytówkach, którymi posługują się pracownicy firmy. Bardzo często zdarza się, że sponsorowanie różnych wydarzeń przynosi duży rozgłos, warto więc namówić organizatorów do umieszczenia informacji o witrynie na ulotkach reklamowych i plakatach. Nieliczni decydują się na promocję poprzez rozdawanie ulotek na ulicach, w szkołach, czy bibliotekach. Czasami takie gazetowe czy telewizyjne dodatki są bardziej efektywne i tańsze niż reklama on-line. Można też spróbować zgłosić stronę do licznych konkursów internetowych o ile strona jest wystarczająco ciekawa, użyteczna, profesjonalna i zachęcająca.

Promocja w Internecie może dotyczyć:

- rejestracji w wyszukiwarkach internetowych – po odpowiednim przygotowaniu strony powinno się ją zarejestrować w kilku największych i najważniejszych, wynikających ze specyfiki rynku, indeksujących wyszukiwarkach. Wyszukiwarki są to potężne bazy danych zawierające adresy i informacje na temat stron www, z których później pośrednio korzysta użytkownik. Dane te są zbierane automatycznie, dzięki działaniu specjalnych programów przeszukujących sieć. Przeglądaj one kolejne strony webowskie podążając za występującymi na nich odnośnikami hipertekstowymi oraz zapamiętują słowa, jakie zostały w nich użyte. Najlepiej byłoby, gdyby nasza strona pojawiła się jako jedna z pierwszych dziesięciu do trzydziestu „trafień” w wyszukiwarce,
- reklama serwisu w miejscach odwiedzanych przez potencjalnych klientów firmy - firmy zazwyczaj należą do różnych towarzyszeń branżowych mających listy swoich członków na własnych stronach internetowych. Jeśli umieści się odnośnik do swojej firmy na takiej stronie to rośnie liczba osób, które mogą dotrzeć do ogłaszanego adresu. Nawet jeśli pobierają za to opłaty to taka reklama może być dla naszej firmy korzystna.
- opracowanie bezpłatnego serwisu – co pomaga w przyciągnięciu ludzi. Stworzenie go kosztuje czasami wiele czasu i energii ale może przyciągnąć wielu klientów chcących otrzymać coś „za darmo”, problem polega jedynie na wymyśleniu dobrej zachęty, będącej jednocześnie zachętą odpowiadającą prawdzie,
- wzajemne odnośniki – występują na stronach firm mających wspólne umowy o reklamie. Znalezienie komplementarnych stron można osiągnąć za pomocą wyszukiwarki, a następnie poprosić wybrane strony o umieszczenie tam linków w zamian za umieszczenie ich linków na swoich stronach. Czasami trwa to trochę czasu i wymaga negocjacji w kwestii miejsca, wielkości itp.,
- wiadomości z branży – w Internecie występują strony, które publikują wiadomości na temat różnych branż. Na ogół nie ma problemu z poproszeniem o zamieszczenie tam artykułu o nowym, zwłaszcza bezpłatnym serwisie. Można też o tym poinformować dziennikarzy zajmujących się nowymi mediami.

- odnośniki o bezpłatnych serwisach – w Internecie występują też strony będące spisami różnych darmowych serwisów – można poprosić o umieszczenie tam linków do swojej strony,
- kolekcjonowanie adresów e-mailowych odwiedzających naszą stronę klientów – w ramach haseł związanych z CRM modna stała się personalizacja serwisów. Firmy zbierają więc adresy e-mailowe użytkowników odwiedzających stronę, jednocześnie prosząc o nadsyłanie uaktualnień. Jeżeli użytkownik wyraża na to zgodę nie może być to uznane za zwykły spamming,
- użytkownicy bezpłatnej poczty często spotykają się z reklamami rozsyłanymi do skrzynek poczty elektronicznej bez ich zgody (spam) – nie jest to mile widziana reklama, ponieważ w odróżnieniu od ulotek roznoszonych do tradycyjnych skrzynek pocztowych użytkownik musi zapłacić za zapoznanie się z reklamą (np. impulsy telefoniczne przy odbieraniu poczty przez modem).
- opublikowanie newslettera – publikacji stworzonej przez firmę lub indywidualne jednostki i wysłana na żądanie. Można je rozprowadzać używając programu pocztowego, stworzonej aplikacji pobierającej adresy bezpośrednio z bazy danych i wysyłającej e-maile z użyciem serwera e-mailowego lub innych programów dostępnych na rynku.
- podpis w e-mailach – w trakcie wysyłania podpisu do innej firmy, zamieszcza się w nim podpis (signature). Jest to plik tekstowy automatycznie dołączany automatycznie do każdego wysyłanego listu, czy wiadomości. Podpis ten jest tworzony na życzenie użytkownika i zawiera najczęściej jego imię, nazwisko i dane adresowe np. nazwę firmy, zakres działalności, adres e-mailowy i URL. Pozwala na szybkie nawiązanie kontaktu z autorem korespondencji i łatwe znalezienie serwisu jego firmy.
- listy dyskusyjne i news – promocja na nich jest dodatkowym sposobem dostarczenia informacji o organizacji i dotarcia do potencjalnych użytkowników. Ponieważ jest to miejsce przeznaczone do dyskusji promocje można tu „przemycać” tylko między jej wierszami. Inaczej może zostać odebrane to jako próba spamu i firma straci reputację. Można też ewentualnie utworzyć własną grupę dyskusyjną np. o problemach z produktami naszej firmy.
- wykupienie reklam bannerowych – jest to jedna z najskuteczniejszych form reklamy, wymaga jednak pieniędzy (około 20 tys. zł/miesiąc) i odpowiedniego przygotowania, sprowadzającego się do precyzyjnego rynku docelowego oraz właściwe określenie serwisów odwiedzanych przez potencjalnych klientów firmy. Rozliczenie odbywa się w oparciu o liczbę odsłon reklamy lub w oparciu o czas jej emisji.
- program rotacji bannerów – każdy uczestnik tej wymiany może prezentować swój banner na stronach innych uczestników w zamian za prezentowanie ich reklam swoim serwisie. Zaletą tego rozwiązania jest , że praktycznie jest to darmowa reklama, wadą że trzeba udostępnić miejsce reklamowe na własnej

stronie nie mając kontroli nad potencjalnymi reklamodawcami, którymi mogą być np. konkurenci.

- wykupienie reklamy w newsletterach – najlepiej wykupywać tam małe reklamy od 4 do 12 linii. Informują one i motywują użytkowników do kliknięcia na dany URL, przez co mogą przyciągnąć do firmy wielu nowych klientów,
- prośba o zapamiętanie strony – jeśli użytkownik zapamięta stronę to istnieje duża szansa, że zajrzy na nią znowu, jeśli nie to zapewne skorzysta z wyszukiwarki i może trafić do konkurencji. Pojawiły się firmy, które oferują robienie zakładek (zapamiętywania) on-line. Polega to na zarejestrowaniu się w serwisie, a później w przypadku znalezienia na stronie przycisku służącego robieniu zakładek – użyciu go. Powoduje to zapamiętanie informacji na serwerze firmy utrzymującej zakładki. Wygoda dla użytkownika polega na trzymaniu zakładek w jednym miejscu i szybkim dostępie do ich katalogu,
- prośba rekomendacji znajomym – wykorzystuje się tu psychologiczną zależność, że jeśli znajomy coś rekomenduje, to może warto poświęcić na to przynajmniej kilka minut.

## 5. Eksploatacja, kontrola rozwoju, przekształcenia i utrzymanie witryny

Głównym elementem pozwalającym na dalszy rozwój stron jest analiza logów<sup>18</sup> serwera WWW. Serwer po otrzymaniu żądania wysłania pliku powinien zapisywać spod jakiego adresu IP nastąpiło żądanie, pełną ścieżkę dostępu do pliku z jego nazwą, kod błędu lub poprawnego wysłania pliku, ew. komunikatu informacyjne na temat wykonywanych programów. Używając „cookies” (internetowych ciasteczek) możemy nie tylko sprawdzić ile razy dany użytkownik oglądał stronę ale również odczytać, które strony najczęściej oglądał i jak wyglądały jego podróże po naszej witrynie. Są to bardzo ważne informacje, które posłużą mogą do odpowiedniego formowania treści i zmiany hierarchii informacji, tak by użytkownik mógł odleźć to czego szuka przy jak najmniejszej ilości kroków. Wszystko to służy do sprawdzenia:

- aktualności informacji na stronie,
- „używalności” informacji,
- aktualności celu i funkcji witryny.

Jeżeli strona przestaje być postrzegana przez klientów, to znak, że powinien rozpocząć się ostatni etap cyklu życia witryny – wycofywanie jej z eksploatacji.

---

<sup>18</sup> Log – od ang. logging to dzienniki tworzone przez oprogramowanie serwerów w czasie ich pracy



## 6. Kryteria akceptowalności tworzenia witryny internetowej

Zasadnicze kryteria, jakie należy rozpatrzyć w procesie tworzenia projektu witryny internetowej dadzą się sprowadzić do następujących trzech grup wskaźników:

- ekonomicznych,
- organizacyjnych,
- informatycznych.

Do najważniejszych wskaźników ekonomicznych, które należało by wziąć pod uwagę przy projektowaniu systemu należą<sup>19</sup>:

- koszt analizy strategii i projektowania witryny,
- koszty współpracy ze specjalistami zewnętrznymi,
- koszt zakupu sprzętu oraz instalacji oprogramowania,
- koszt szkolenia i wdrożenia systemu,
- koszt wypromowania portalu,
- koszt bieżącej eksploatacji serwisu oraz jego konfiguracji, modyfikacji, rozbudowy,
- przewidywane efekty wdrożenia,
- czas realizacji, wpływający znacząco na koszty.

Wskaźniki organizacyjne powinny związane są ze zmniejszeniem ryzyka niepowodzenia wdrożenia i utrzymania portalu oraz szeroko rozumianymi jego możliwościami rozwojowymi.

Wdrożenie systemu jest procesem długotrwałym i wymagającym ogromnego nakładu pracy, dlatego należy wymagać aby:

- system – o ile jest taka możliwość i konieczność - był wdrażany modułowo,
- zostały określone środki, jakie muszą być do tego zaangażowane,
- zespół przeprowadzający przedsięwzięcie powinien mieć ugruntowaną i stabilną pozycję na rynku a ponadto: kompetencje i zaangażowanie,
- system powinien się charakteryzować się elastycznością uniwersalnością i niezawodnością,
- prowadzić rozsądną politykę marketingową i inwestycyjną. Zawsze należy brać pod uwagę wysokie ryzyko przeinwestowania.

Kryteria informatyczne dotyczą z jednej strony technologii tworzenia witryny, jego architektury, platformy sprzętowej i systemowej oraz współpracy z innymi systemami, a z drugiej strony bezpieczeństwa przeprowadzanych w nim operacji.

Wśród najważniejszych kryteriów technologicznych można wyróżnić:

- określenie platformy działania systemu i platformy baz danych,

---

<sup>19</sup> por.: Safaryn A.: *Portale w elektronicznym biznesie na przykładzie sektora finansów i bankowości*, WSISiZ, Warszawa, 2001,

- specyfikacja narzędzi tworzenia i utrzymania aplikacji usługowych,
- stworzenie kompletnych rozwiązań sieciowych,
- zdefiniowania narzędzi WebMaster'a,
- postawienia (lub nie) FireWall'a,
- zakup i instalację serwerów.

Oferując użytkownikom informacje i usługi trzeba zadbać o rozwiązania pozwalające na korzystanie z portalu w sposób bezpieczny. Wśród najważniejszych kryteriów zapewnienia bezpieczeństwa oferuje się:

- uwierzytelnienie - polegające na potwierdzeniu tożsamości użytkownika, a zwłaszcza weryfikacji tożsamości przesyłki,
- poufność – ochrona informacji przed jej ujawnieniem nieuprawnionemu odbiorcy,
- przezroczystość – powodująca, że zabezpieczenia będą dla użytkowników końcowych niewidoczne,
- integralność – ochronie przed modyfikowaniem wiadomości przez osoby nieupoważnione.

## 7. Problemy związane z projektem witryny i sposoby ich przewyżczenia

Tworzenie projektu strony internetowej i jego realizacja przysparza wiele problemów. Niezbędne jest znaczące zaangażowanie pracujących przy nim osób oraz szeroko rozumiane poparcie kierownictwa. Pomimo ogólnie znanych zdroworoządkowych zasad tworzenia witryn, istniejących poradników i podręczników oraz coraz większej ilości specjalistów z tego zakresu nadal w trakcie projektowania stron www zdarzają się liczne problemy wymagające jak najszybszego rozwiązania.

W poniższej tabeli podano wybrane z nich, wraz z metodami zaradczymi.

Tablica1. Wybrane problemy w projektowaniu witryny oraz metody zapobiegania problemom

<b>Charakterystyka problemu</b>	<b>Sposób zapobiegania</b>
Projekt graficzny portalu powinien być przejrzysty, jego struktura jasna i zrozumiała, podstawowe funkcje jasno i jednoznacznie wykonywalne. Występuje problem z uzgodnieniem postaci, akceptowalnej przez wszystkich członków zespołu projektowego	Zastosowanie lekkiej grafiki i pastelowych kolorów, oddzielenie miejsc na reklamę, tak aby bannery reklamowe nie myliły się z odsyłaczami do usług dostępnych w witrynie. Konsultacja z grafikiem. Stworzenie pilota, do konsultacji z członkami zespołu projektowego.
Brak projektu struktury, organizacji, interfejsów witryny.	Należy zaplanować strukturę w taki sposób, który będzie najbardziej odpowiedni dla określonego typu klienta.
Stworzenie organizacji baz danych	Zaimplementowanie baz danych. W tym celu należy stworzyć projekty poszczególnych baz danych (specyfikacja źródeł danych, rekordów,

Charakterystyka problemu	Sposób zapobiegania
	relacji kryteriów wyszukiwania, raportów, sposobów aktualizowania oraz udostępniania danych). Wspólna koncepcja utrzymania i administrowania bazami danych – wyznaczenie administratorów.
Organizacja usług on-line	Lista proponowanych usług. Oferty poszczególnych usług (cenniki, regulaminy, formularze zamówień, specyfikacja odbiorców i dostawców). Sposoby udostępnienie poszczególnych usług (interfejsy, uwierzytelnienie klientów, umowy i zasady rozliczeń z dostawcami usług). Wspólna koncepcja utrzymania i administrowania usługami – pomoc.
Konfiguracja i utrzymanie platformy sprzętowej i systemowej	Wyznaczenie osób z odpowiednimi kwalifikacjami, przeznaczenie funduszu na zakup sprzętu i oprogramowania – przetestowanie.
Brak kontaktów z przedstawicielami banków lub instytucji finansowych upoważnionymi do współpracy przy tworzeniu witryny.	Na potrzeby współpracy z bankami należy stworzyć wewnętrzną listę kontaktów, wg której będą rozsyłane propozycje, zapytania, formularze, zgłoszenia i inne dokumenty dotyczące współpracy przy tworzeniu i redagowaniu witryny.
Nawiązanie współpracy z innymi instytucjami zewnętrznymi	Stworzyć zespół osób, które opracują program motywujący instytucje, przewidywane do współpracy w celu rozwinięcia usług on-line w na witrynie oraz aktualizacji baz danych
Jeżeli witryna ma zawierać serwis informacyjny, to: brak redakcji informacyjnej	Należy podpisać umowę o współpracy na zasadach partnerskich z istniejącą już redakcją prasową oraz zakupić serwisy informacyjnych np.: PAP, Reuters, innych. Operacja wymaga zebrania i porównania ofert cenowych oraz opracowania zasad udostępniania serwisów.
Brak narzędzia do publikacji i archiwizacji informacji	Niezbędne jest narzędzie, umożliwiające automatyczną publikację i archiwizację informacji – wymaga to zakupu odpowiednich narzędzi programistycznych, wykonania aplikacji i przeznaczenia odpowiednich zasobów dyskowych na serwerze

Źródło: opracowanie własne na podstawie Safaryn A.: *Portale w elektronicznym biznesie na przykładzie sektora finansów i bankowości*, WSISiZ, Warszawa, 2001,



## 8. Podsumowanie

Jak już wspomniano na samym wstępie proces projektowania witryny internetowej nie jest – wbrew powszechnemu przekonaniu – sprawą łatwą i prostą. Wymaga umiejętności i wiadomości z wielu, czasami bardzo odległych dziedzin. Niniejszy artykuł stanowi próbę skatalogowania podstawowych aspektów projektowania witryn internetowych, podjęta z różnych punktów widzenia. Specyfikację aspektów projektowania stron przedstawiono w oparciu o fazy cyklu życia witryny internetowej. Starano się uwzględnić w niej wszystkie problemy związane z tworzeniem witryn w każdej fazie cyklu, wielowątkowość porównań nie uchroniła jednak prezentowanego przeglądu od pewnych powtórzeń. Tym niemniej może on stanowić materiał wyjściowy do dalszych uogólnień i badań zmierzających do wypracowania bardziej szczegółowej procedury i określenia optymalnej metodyki projektowania stron internetowych przeznaczonych do obsługi sytuacji biznesowych.

## Literatura

1. Chmielarz W.: *Handel elektroniczny nie tylko w gospodarce wirtualnej*, Wydawnictwa Naukowe Wydziału Zarządzania UW, Warszawa, 2001.
2. Dyba A.: *Jak skutecznie zaplanować strategię internetową*, <http://idm.supermedia.pl/iso/luneta/>
3. Ellsworth J. H., Matthew V.: *The Internet Business Book*, John Wiley & Sons, New York, 1998
4. Haine D.: *Five most Serious Web Design Errors*, E-Business Magazine, marzec, 1998, <http://hpcc920.external.hp.com/Ebusiness/webdesign.html>
5. <http://www.cavern.pl/poradnik.htm>
6. Klus J.: *Ewolucja reklamy w Internecie*, WZ UW, Warszawa, 2001,
7. Maloff J.: *Growing Business at Net Speed*, Internet World, lipiec 1997, str. 68-70.
8. Mazurek K.: *Rozwiązywanie problemów technicznych konstruowania i wdrażania „Portalu Regionalnego Mazowsze”*, WSISiZ, Warszawa, 2001,
9. McAteer E., Shaw R.: *The Design of Multimedia Learning Programs*, University of Glasgow, Glasgow, 1995
10. Safaryn A.: *Portale w elektronicznym biznesie na przykładzie sektora finansów i bankowości*, WSISiZ, Warszawa, 2001,
11. Sweet L.: *Ringing the E-Commerce Cash Register*, Internet Computing, marzec, 1998, str. 57-68.
12. Taylor D.: *“Tworzenie stron WWW”*; Wydawnictwo RM, 1998.
13. Turban E., Lee J., King D., Chung H.M.: *Electronic Commerce. A Managerial Perspective*, Pearson Education, London, Hongkong, NY, 2000
14. Turban E.: *Determinants of Successful Website Design: Relative importance and Recommendations for Effectiveness*, <http://www.eturban.csulb.edu>

15. Vassos T.: *Strategie marketingowe w Internecie*, Studio Emka, Warszawa, 1999
16. Ziemia E.: *Wskazówki do metodyki projektowania i wdrażania witryn internetowych w: Systemy Wspomagania Organizacji SWO'2001* pod red. naukową J. Gołuchowskiego, H.Sroki, Wydawnictwo AE w Katowicach, Katowice, 2001

Dr hab. Witold CHMIELARZ, prof. UW  
Wydział Zarządzania Uniwersytetu Warszawskiego  
ul. Szturmowa 3; 02-678 Warszawa  
tel.: (22) 55-34-002  
vitec@post.pl

# METODA *COSMIC-FFP* – UNIWERSALNY SPOSÓB POMIARU I ESTYMACJI ROZMIARU FUNKCJONALNEGO SI

BEATA CZARNACKA-CHROBOT

*“Problem z podejściem opartym na punktach funkcyjnych jest taki, że ocenia ono ograniczoną grupę rodzajów aplikacji: zwykle duże, bazujące na aplikacjach systemy, realizowane w organizacjach takich jak banki, towarzystwa budowlane oraz instytucje handlu detalicznego, nie radząc sobie z systemami hybrydowymi (...) ze znacznym komponentem komunikacyjnym.”*

D. C. Ince,

*History and industrial application, in Fenton, N.E., Software Metrics: A Rigorous Approach, Chapman & Hall, 1991.*

**Streszczenie:** Skuteczne zarządzanie projektowaniem i utrzymaniem systemów informatycznych (SI) nie jest możliwe bez pomiaru i estymacji rozmiaru tworzonego lub gotowego systemu. Jak bowiem słusznie zauważył Tom DeMarco: *“nie można prawidłowo zarządzać czymś, czego się nie mierzy”* (por. [5]). Wyznaczenie wielkości systemu stanowi warunek konieczny do właściwego określenia jego wartości, produktywności, czasu koniecznego do jego powstania oraz nakładów niezbędnych do realizacji SI. Prawidłowy pomiar lub szacowanie rozmiaru systemu informatycznego jest zaś możliwe jedynie przy użyciu odpowiednich metod, które do tego celu służą. Niniejszy artykuł ma na celu ogólne przybliżenie zasad właśnie jednej z funkcjonalnych metod estymacji i analizy wielkości SI. Historia metody *COSMIC-FFP* rozpoczęła się pod koniec lat 90., kiedy to - po niemal dwudziestu latach wykorzystywania analizy opartej na zwykłych punktach funkcyjnych, stworzonej przez A. Albrechata i modyfikowanej przez instytucję *IFPUG* - okazało się, iż metoda ta ma ograniczone zastosowanie. Wtedy właśnie, w celu stworzenia sposobu postępowania, który charakteryzowałby się większą uniwersalnością, powołano organizację *COSMIC* (ang. *Common Software Measurement International Consortium*), która w połowie 2001 roku wydała zbiór zasad i norm określających wykorzystanie metody opartej na tzw. **pełnych punktach funkcyjnych** (ang. *Full Function Point - FFP*) w wersji 2.1. W artykule przedstawiono jedynie podstawowe idee metody *COSMIC-FFP*, w tym genezę tego podejścia, jego przeznaczenie oraz wykorzystywany w nim ogólny model pomiaru rozmiaru funkcjonalnego SI<sup>1</sup>.

---

<sup>1</sup> Opis modelu pomiaru i estymacji bazuje w dużej części na informacjach zawartych w [COSM01].



## 1. Metody funkcjonalne pomiaru i estymacji rozmiaru SI

Ogólnie rzecz ujmując, do pomiaru i szacowania projektów informatycznych, jak też do wyznaczenia wielkości gotowego systemu informatycznego, wykorzystuje się dwie grupy metod, które opierają się na zupełnie odmiennych zasadach i uwzględniają zupełnie inne elementy. Pierwszą grupę stanowią metody oparte na jednostkach programowych (liczba linii kodu źródłowego i liczba poleceń), drugą zaś metody oparte na jednostkach umownych, określane mianem metod funkcjonalnych (por. [3]). Znaczenie jednostek programowych obecnie zdecydowanie zmalało, między innymi dlatego, iż nie pokazują one funkcjonalności, jaką system dostarcza, czy powinien dostarczać, użytkownikowi. A owa funkcjonalność stanowi parametr kluczowy dla rzeczywistego wykorzystania zrealizowanych systemów informatycznych, co z punktu widzenia użytkownika ma znaczenie decydujące o wartości SI. Miary funkcjonalne są niezbędne do określenia poziomu wydajności produktów lub usług właśnie z perspektywy użytkownika - dlatego powinny być niezależne od technicznych parametrów projektowania i implementacji. Wtedy mogą być wykorzystywane do porównania produktywności różnych technik i technologii.

Zauważając konieczność pomiaru i estymacji wielkości systemu z punktu widzenia jego funkcjonalności dla użytkownika, już pod koniec lat 80. Allan Albrecht (por. [1]) stworzył **metodę punktów funkcyjnych** (ang. *Function Points Method*) jako alternatywą do tych, dla których podstawę stanowią jednostki programowe. Była to pierwsza wersja metody dokonującej pomiaru funkcjonalności systemu, którą wyraża się w punktach funkcyjnych (PF), stanowiących jednostki umowne rozmiaru funkcjonalnego SI. Metoda ta przez ponad dwadzieścia lat rozwoju ulegała licznym modyfikacjom, uściśleniom i udoskonaleniom, wprowadzanym przede wszystkim przez międzynarodową organizację *IFPUG* (ang. *International Function Point Users Group*). W dniu dzisiejszym obowiązują jej standardy zgodne z wersją *IFPUG 4.1* (por. [4]).

Rozwiązanie oparte na zwykłych punktach funkcyjnych stanowi efektywną miarę rozmiaru funkcjonalnego systemów wspomagających zarządzanie, czyli – ogólnie rzecz ujmując - aplikacji powstających w celu wspomagania operacji informacyjnych, gospodarczych oraz administracyjnych. Ocenia się, iż takie systemy stanowią od 60 – 80% ogólnej liczby SI. Użytkownikiem tego rodzaju systemów, a więc i odbiorcą ich funkcjonalności, jest zwykle *człowiek*. Istnieją jednak również inne kategorie SI, których wielkość i funkcjonalność powinna także podlegać pomiarowi i estymacji. Kategorie te różnią się znacznie od systemów informatycznych zarządzania, przede wszystkim z punktu widzenia *odbiorcy funkcjonalności*, dlatego, jak się wydaje, nie mogą być mierzone w sposób identyczny jak aplikacje wspomagające zarządzanie.

Wniosek ten potwierdza praktyka. Mimo liczących zmian, jakie wprowadzają kolejne wersje metody punktów funkcyjnych, w tym wersja ostatnia z 1999 r., ciągle bowiem uważa się, iż jej bieżące standardy *nie są uniwersalne*. Ściślej mówiąc, jest to metoda, która prawidłowo wykorzystywana znakomicie nadaje się do pomiaru i szacowania wielkości systemów informatycznych wspomagających zarządzanie, jednak, *nie uwzględniając w sposób wystarczający wewnętrzną złożoność przetwarzania w SI*, co uważa się za jej podstawową wadę,

nie może być stosowana dla wszystkich rodzajów owych systemów. W odniesieniu do systemów informatycznych, dla których zasadnicze znaczenie mają algorytmy przetwarzania (np. systemy czasu rzeczywistego), nie daje ona prawidłowych rezultatów.

Ze względu na niemożność odzwierciedlenia przez analizę PF wysokiej złożoności wewnętrznej systemu informatycznego już na przełomie lat 80. i 90. pojawiło się wiele jej wariantów. Miały one przede wszystkim na celu rozszerzenie obszaru zastosowania umownych jednostek pomiaru rozmiaru funkcjonalnego na wszystkie kategorie systemów informatycznych. Ocenia się, iż obecnie istnieje co najmniej 20 takich wariantów, a do najpopularniejszych należą (por. [3]):

- ▶ metoda punktów charakterystycznych (1987 r.),
- ▶ bardzo popularna w Wielkiej Brytanii metoda *Mark II* (1989 r.),
- ▶ metoda *3D Function Points* (1994 r.), oraz
- ▶ metoda pełnych punktów funkcyjnych.

## 2. Geneza i przeznaczenie metody *COSMIC-FFP*

Koncepcja **metody pełnych punktów funkcyjnych** (ang. *Full Function Point - FFP*) powstała już w 1997 roku na *University of Quebec* w Montrealu (*UQAM*) we współpracy z firmą *Software Engineering Laboratory in Applied Metrics (SELAM)*. Jej celem było rozszerzenie analizy PF na te rodzaje oprogramowania, których głównym odbiorcą funkcjonalności, a więc użytkownikiem, *nie jest człowiek*. W związku z powyższym, punktem wyjścia dla stworzenia nowego rozwiązania stało się rozszerzenie definicji użytkownika wykorzystywanej przez *IFPUG*: "**Użytkownikiem mogą być nie tylko osoby, ale też oprogramowanie i urządzenia mechaniczne wchodzące w interakcje z mierzonym oprogramowaniem**" [7, s. 15]. Dlatego koncepcja *FFP* z 1997 roku, nazwana później wersją 1 metody *FFP*, skupia się przede wszystkim na systemach czasu rzeczywistego, systemach operacyjnych, infrastrukturalnych aplikacjach wspomagających oraz oprogramowaniu wbudowanym i sterującym. Oczywiście przedmiotem jej badań są nadal systemy wspomagające zarządzanie.

Obecnie rozwojem metody *FFP* zajmuje się organizacja *COSMIC* (ang. *Common Software Measurement International Consortium*), powołana do życia pod koniec lat 90., stanowiąca międzynarodową grupę ekspertów zajmujących się pomiarem rozmiaru funkcjonalnego SI. Celem jej powstania było właśnie rozwinięcie i wprowadzenie na rynek kolejnej generacji metody funkcjonalnej, której zadaniem miał być pomiar wielkości wszystkich rodzajów systemów informatycznych, dodatkowo zgodny z wymaganiami stawianymi w tym względzie przez *ISO*. Prace nad nową wersją metody *FFP* rozpoczęto w listopadzie 1998 roku. Po licznych testach, przeprowadzanych zgodnie z zaleceniami tej organizacji standaryzacyjnej w bankowości, lotnictwie, przemyśle samochodowym oraz w korporacjach telekomunikacyjnych, w połowie 2001 roku *COSMIC* wydała zbiór zasad i norm określających wykorzystanie metody *COSMIC-FFP* w wersji 2.1.



Metoda *FFP* w wersji obecnej wykorzystuje dorobek metody punktów funkcyjnych *IFPUG* w wersji 4.1, podejścia *Mark II*, analizy PF opracowanej przez *NESMA* (holenderski odpowiednik *IFPUG*) oraz sposobu postępowania opartego na pełnych punktach funkcyjnych z 1997 roku. Uwzględniając standardy *ISO* co do pomiaru rozmiaru funkcjonalnego  $SI^2$  zawiera też nowe idee. Jest przeznaczona dla większości rodzajów oprogramowania, w tym dla (por. [2]):

- ▶ **Aplikacji**, czyli programów, które dostarczają zaprogramowane funkcje, przy których użytkownik wprowadza lub otrzymuje dane, nie tworząc własnych funkcji. Tego typu oprogramowanie jest zwykle wykorzystywane do wspomagania zarządzania biznesowego w takich organizacjach gospodarczych, jak banki, instytucje ubezpieczeniowe, czy działach zajmujących się na przykład księgowością, kadrami, zakupami, dystrybucją, czy produkcją. Takie systemy określa się często mianem "systemów sterowanych danymi", ponieważ ich złożoność wynika przede wszystkim z konieczności zarządzania dużą ilością danych, które dotyczą zdarzeń mających miejsce w świecie rzeczywistym.
- ▶ **Systemów czasu rzeczywistego**, zadaniem których jest realizacja lub sterowanie zdarzeniami mającymi miejsce w świecie rzeczywistym. Wśród przykładów oprogramowania tego typu można wyróżnić systemy central telefonicznych, komutacji komunikatów, oprogramowanie wbudowane w urządzenia w celu sterowania pracą maszyn (np. wind, silników samochodowych, urządzeń wykorzystywanych w gospodarstwie domowym), systemy sterowania procesami i automatycznego gromadzenia danych oraz systemy operacyjne komputerów.
- ▶ **Systemów hybrydowych**, tj. łączących cechy obu wyżej wymienionych rodzajów oprogramowania, których przykładem mogą być systemy rezerwacji w czasie rzeczywistym stosowane w hotelach, czy liniach lotniczych.

Jednak wersja 2.1 metody *COSMIC-FFP*, jak na razie, nie może być stosowana jako ogólny standard do pomiaru i estymacji rozmiaru funkcjonalnego oprogramowania, bądź jego części, które:

- ▶ Charakteryzuje się skomplikowanymi algorytmami matematycznymi lub innymi wyspecjalizowanymi i złożonymi zasadami działania, jakie można spotkać na przykład w systemach ekspertowych, oprogramowaniu symulacyjnym, systemach samouczących się, czy w oprogramowaniu wspomagającym prognozowanie pogody.
- ▶ Przetwarza zmienne ciągłe, jak dźwięki audio lub obrazy wideo, które można znaleźć na przykład w grach komputerowych lub oprogramowaniu instrumentów muzycznych<sup>3</sup>.

---

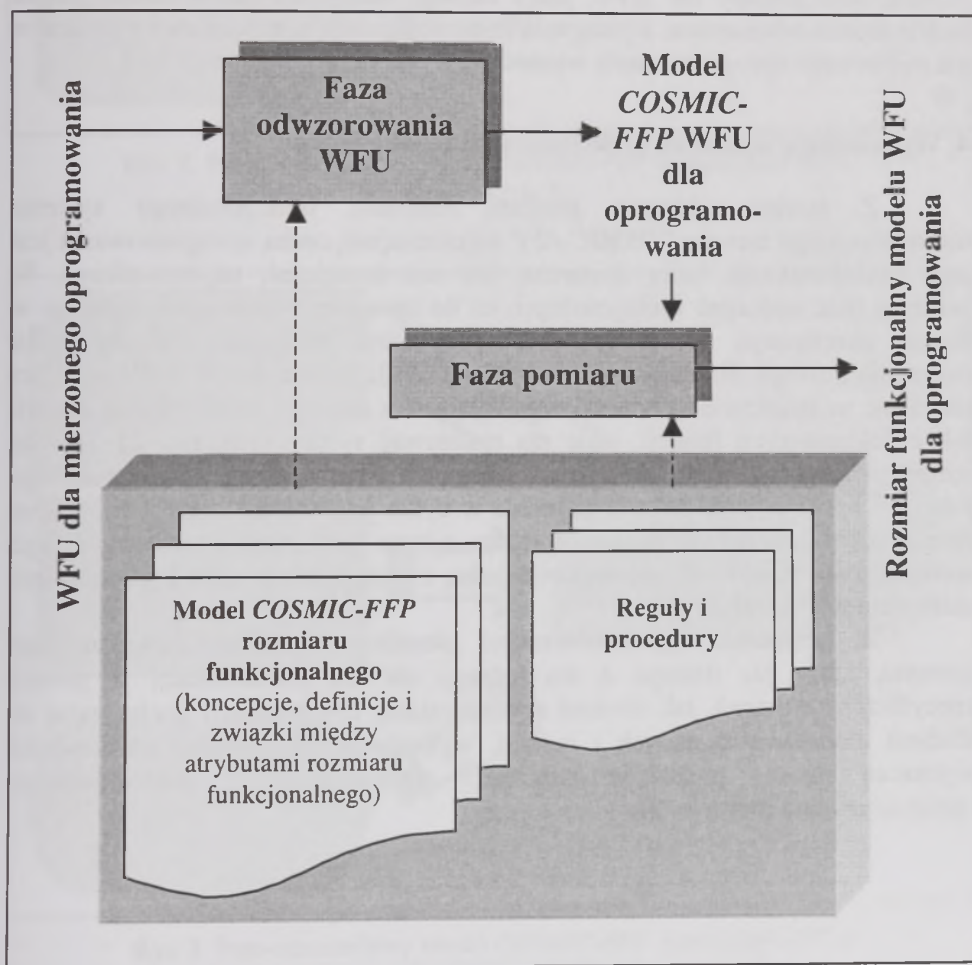
<sup>2</sup> Chodzi o standard *ISO* 14143/1.

<sup>3</sup> Oba wymienione rodzaje oprogramowania mogą być jednak uwzględnione poprzez zdefiniowanie standardów lokalnych będących rozszerzeniami dla metody *COSMIC-FFP*, o których można przeczytać w [COSM01].



### 3. Model procesu pomiaru w metodzie *COSMIC-FFP*

Metoda pomiaru rozmiaru funkcjonalnego *COSMIC-FFP* została tak zaprojektowana, aby stanowić podejście niezależne od decyzji związanych z implementacją. Dlatego pomiar odbywa się zgodnie z ogólnym modelem wymagań funkcjonalnych użytkownika dla oprogramowania, odwzorowującym tylko niezbędne do ich wyznaczenia cechy mierzonego systemu. Przy czym przez **wymagania funkcjonalne użytkownika (WFU)** rozumie się termin zgodny z interpretacją *ISO*, oznaczający "podzbiór wymagań użytkownika oparty na jego punkcie widzenia. WFU reprezentują praktyki i procedury użytkownika, które muszą być realizowane przez oprogramowanie w celu zaspokojenia jego potrzeb. Wymagania funkcjonalne użytkownika nie obejmują wymagań jakościowych oraz wymagań technicznych." (por. [6]).



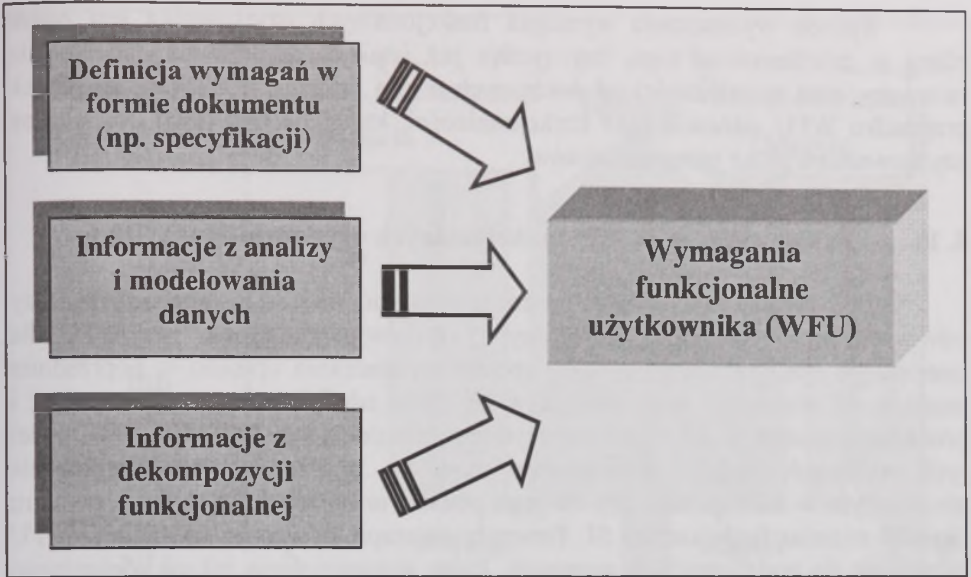
Rys. 1. Model procesu pomiaru w metodzie *COSMIC-FFP*  
Źródło: [2, s. 13]

Model procesu pomiaru wykorzystywany w opisywanej metodzie przedstawiono na rys. 1. Jak wynika z tego ogólnego schematu pierwszy krok procedury stanowi odwzorowanie wymagań funkcjonalnych użytkownika dla mierzonego oprogramowania w ich specyficzny dla podejścia *COSMIC-FFP* model. Ów model WFU uwzględnia różne koncepcje, definicje i relacje (strukturę funkcjonalną), które są niezbędne do właściwego wyznaczenia rozmiaru funkcjonalnego SI. Określono również definicje, zasady i reguły dla samego procesu odwzorowywania wymagań w ten specyficzny model wymagań. Zgodnie z jedną z zasadniczych zasad, takie przekształcenie musi się odbyć jeszcze przed zastosowaniem reguł i procedur pomiaru. Dopiero po stworzeniu modelu wymagań funkcjonalnych *COSMIC-FFP* dla oprogramowania można rozpocząć proces pomiaru, który polega na zastosowaniu kolejnego zbioru zasad i reguł w odniesieniu do tego modelu, a jego efektem jest wartość liczbową reprezentująca rozmiar funkcjonalny tak WFU, jak i samego oprogramowania. Zanim jednak będzie można odwzorować wymagania za pomocą modelu, należy owe wymagania dla mierzonego oprogramowania wyznaczyć.

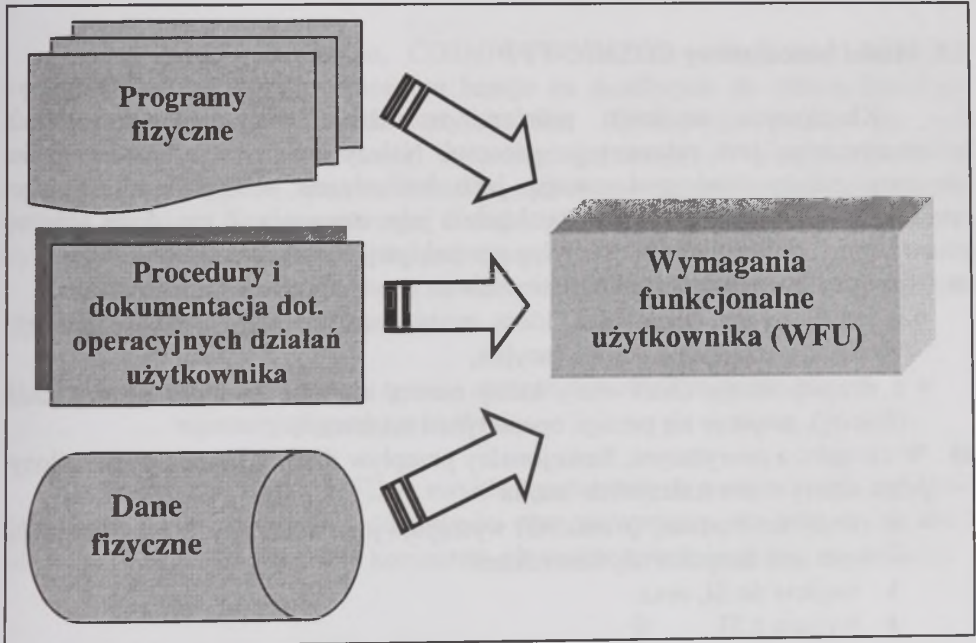
#### **4. Wyznaczanie wymagań funkcjonalnych użytkownika**

Z punktu widzenia pomiaru rozmiaru funkcjonalnego systemu informatycznego metodą *COSMIC-FFP* najistotniejszą cechą oprogramowania jest jego funkcjonalność, którą dostarcza, lub ma dostarczać, użytkownikowi. W praktyce lista wymagań funkcjonalnych co do oprogramowania rzadko istnieje w formie określonego dokumentu (np. specyfikacji wymagań). W przypadku tworzenia nowego SI zdecydowanie częściej WFU muszą być w rzeczywistości określane na podstawie informacji wynikających z analizy i modelowania danych lub z dekompozycji funkcji, jakie ma realizować system (por. rys. 2). Daje to jednak możliwość wyznaczenia listy wymagań, a przez to również funkcjonalnego rozmiaru systemu, stosunkowo wcześniej w cyklu jego życia, a na pewno zanim jeszcze oprogramowanie obsługujące ów system powstanie (czyli przed jego wdrożeniem). Stanowi to niewątpliwie jedną z zasadniczych cech i jednocześnie zalet metody *COSMIC-FFP*.

W przypadku zaś konieczności określenia rozmiaru funkcjonalnego systemu, który już istnieje, a dla którego nie ma dokumentacji w postaci specyfikacji wymagań, jak również wystarczającej dokumentacji pochodzącej ze stadium modelowania danych i funkcji, wymagania funkcjonalne użytkownika wyznacza się na podstawie informacji pochodzących z zainstalowanego oprogramowania (por. rys. 3).



Rys. 2. Przed-wdrożeniowy model *COSMIC-FFP* wyznaczania WFU  
 Źródło: [2, s. 14]



Rys. 3. Post-wdrożeniowy model *COSMIC-FFP* wyznaczania WFU  
 Źródło: [2, s. 14]



Sposób wyznaczania wymagań funkcjonalnych użytkownika jest zatem różny w zależności od tego, czy system już istnieje, czy też ma być właśnie tworzony, oraz w zależności od dostępnych o nim informacji. W każdym jednak przypadku WFU stanowią opis funkcjonalności, która ma być (jest) dostarczana użytkownikom przez oprogramowanie.

## 5. Faza odwzorowania wymagań funkcjonalnych użytkownika

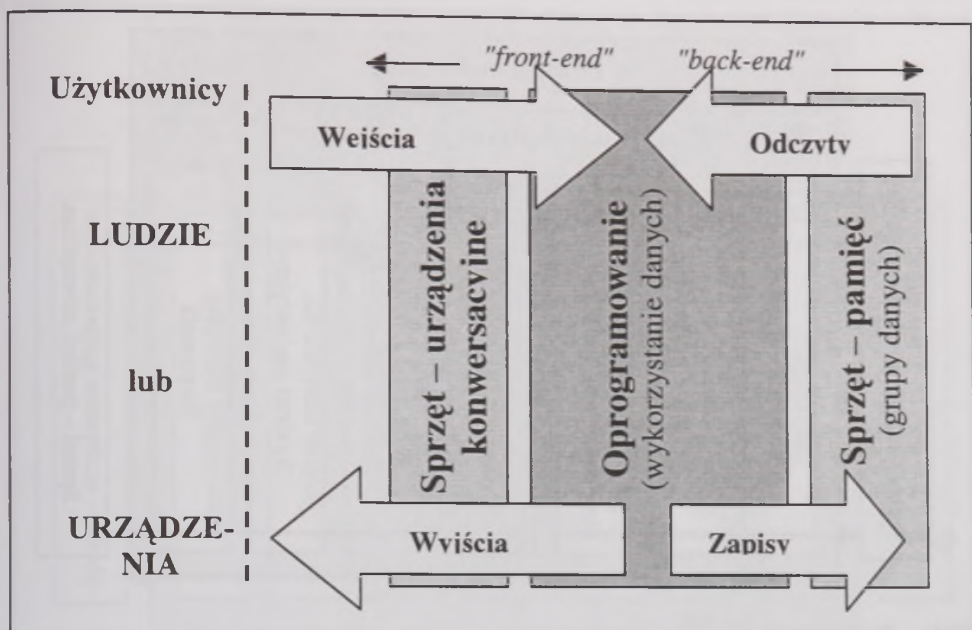
Jak wynika z modelu procesu pomiaru (rys. 1), wejście do fazy odwzorowania wymagań funkcjonalnych użytkownika stanowi lista WFU dla mierzonego oprogramowania, której sposób wyznaczania opisano w poprzednim punkcie. W rezultacie zastosowania w tej fazie zdefiniowanego zbioru reguł i procedur powstaje w jej wyniku specyficzny dla podejścia *COSMIC-FFP* model tych wymagań, będący jednocześnie modelem mierzonego oprogramowania stworzonym w taki sposób, aby na jego podstawie można było w fazie pomiaru określić rozmiar funkcjonalny SI. Powstały na etapie odwzorowania model WFU odnosi się do podzbioru tych wymagań, które stanowią mają zakres konkretnego pomiaru.

Model *COSMIC-FFP* funkcjonalnych wymagań użytkownika dla oprogramowania składa się z dwóch części: modelu kontekstowego i modelu oprogramowania.

### 5.1. Model kontekstowy *COSMIC-FFP*

Kluczowym aspektem pomiaru rozmiaru funkcjonalnego systemu informatycznego jest zakres tego procesu. Należy mianowicie ustalić, jakie elementy należy brać pod uwagę jako wchodzące w skład mierzonego oprogramowania, jakie zaś uznać za składnik jego otoczenia. Z rys. 4, na którym zilustrowano ogólny przepływ danych z perspektywy funkcjonalnej, wynika, iż:

- ▶ Oprogramowanie napotyka ograniczenia ze strony sprzętu komputerowego:
  - z jednej strony (*front-end*), którą można nazwać *zewnętrzną* (przednią), występują urządzenia konwersacyjne,
  - z drugiej strony (*back-end*), którą można określić mianem *wewnętrznej* (tylnej), znajduje się pamięć operacyjna i masowa.
- ▶ W związku z powyższym, funkcjonalny przepływ danych może być określony przez cztery różne rodzaje ich "ruchu":
  - ze strony zewnętrznej (*front-end*) występują jego dwa typy, które pozwalają na wymianę danych z użytkownikami:
    - ▶ wejścia do SI, oraz
    - ▶ wyjścia z SI,
  - ze strony wewnętrznej (*back-end*) kolejne dwa rodzaje, które pozwalają na wymianę danych z pamięcią komputerową:
    - ▶ zapisy, oraz
    - ▶ odczyty.



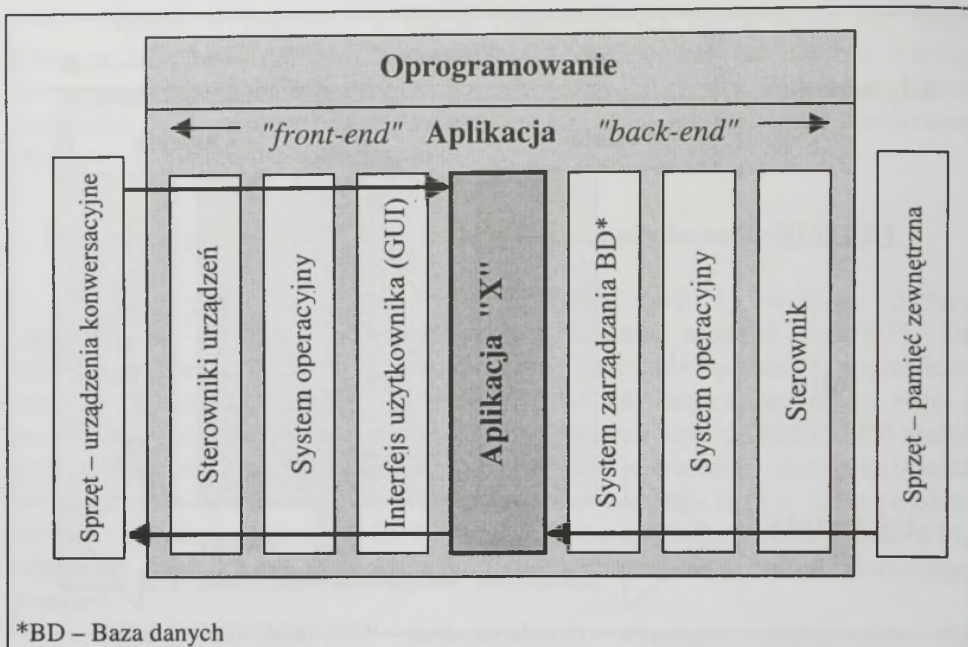
Rys. 4. Ogólny przepływ danych z punktu widzenia funkcjonalnego  
 Źródło: [2, s. 15]

Jak wyżej zauważono, *COSMIC-FFP* ma na celu pomiar wielkości oprogramowania, którego procedura bazuje na możliwych do zidentyfikowania funkcjonalnych wymaganiach użytkownika. Owe wymagania, po ich zdefiniowaniu, przypisywane są do całego systemu, a więc zarówno do oprogramowania, jak i sprzętu. Jednakże opisywana metoda ma na celu pomiar jedynie rozmiaru oprogramowania, dlatego tylko WFU przypisane do tej części systemu powinny być brane w niej pod uwagę.

Poniżej opisane zostaną dwa przykłady dotyczące funkcjonalnych wymagań odnośnie oprogramowania.

#### Przypadek 1: Oprogramowanie typowe dla systemów informatycznych wspomagających zarządzanie

W tym przypadku WFU są zwykle wyznaczane na poziomie aplikacji (por. "aplikacja X" na rys. 5). Pozostałe części oprogramowania, stanowiące otoczenie dla tej aplikacji, są oczywiście konieczne, ale wykorzystuje się je bez modyfikacji.



Rys. 5. Funkcyjny warstwowy model architektury oprogramowania – przykład alokacji wymagań dla oprogramowania systemów wspomagających zarządzanie  
 Źródło: [2, s. 16]

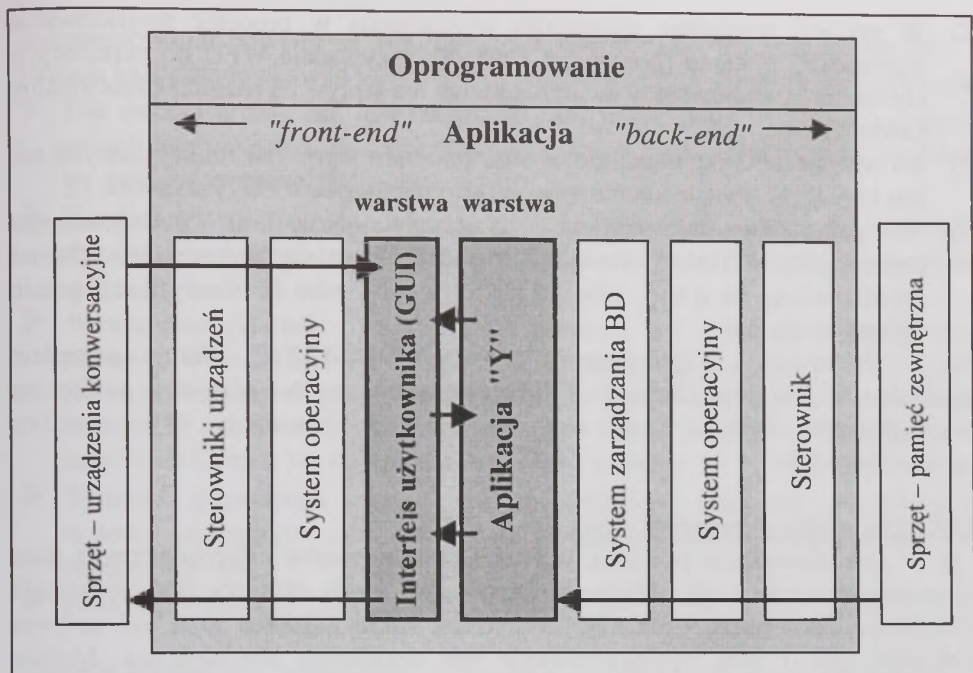
W powyższym przykładzie identyfikacja funkcjonalności dostarczanej przez aplikację X, a przez to również pomiar funkcjonalności określonej przez granice owej aplikacji, jest stosunkowo prosta. Wynika to z następujących faktów:

- ▶▶ Wszystkie wymagania funkcjonalne użytkownika przypisane są do jednej wyróżnionej części oprogramowania;
- ▶▶ Owa część oprogramowania może być zwykle w łatwy sposób wyodrębniona z pozostałych części oprogramowania, stanowiących jej środowisko, dzięki dobrze zdefiniowanej granicy o charakterze funkcjonalnym. W takim przypadku ignoruje się oprogramowanie inne niż aplikacja X, przez co granice tej aplikacji będą pokrywać się z granicami mierzonego oprogramowania (por. rys. 5).

### Przypadek 2: Oprogramowanie "wieloskładnikowe" (ang. multi-piece)

W takim przypadku WFU mogą być po części przypisane do aplikacji (por. "aplikacja Y" na rys. 6), zaś inna ich część do tworzonego na zamówienie użytkownika graficznego interfejsu. Pozostałe części oprogramowania są wykorzystywane, gdy znajdzie taka potrzeba ze strony mierzonego oprogramowania, ale bez modyfikacji.





Rys. 6. Funkcjonalny warstwowy model architektury oprogramowania – przykład alokacji wymagań dla oprogramowania "wieloskładnikowego"

Źródło: [2, s. 16]

W omawianym przykładzie występuje zależność pomiędzy aplikacją Y oraz graficznym interfejsem użytkownika (ang. *Graphical User Interface* – GUI). Jest to bowiem przypadek, w którym w celu dostarczenia funkcjonalności przez aplikację konieczne jest wywołanie przez nią interfejsu użytkownika, a zatem wykorzystanie jego funkcjonalności. Z powodu tej zależności właśnie, funkcjonalność samego GUI nie jest dla użytkownika widoczna. Dostrzega on jedynie funkcjonalność aplikacji. Jednak w celu pomiaru rozmiaru funkcjonalnego wszystkich WFU należy rozróżnić te dwa składniki oprogramowania. W tym celu w metodzie *COSMIC-FFP* umieszcza się każdy składnik w innej warstwie, w związku z czym każdy z nich będzie miał swój własny, oddzielny zakres. W sytuacji, gdy funkcjonalna granica pomiędzy owymi składnikami jest dobrze zdefiniowana i ustrukturalizowana, osoba odpowiedzialna za pomiar nie powinna w praktyce napotkać problemów w określeniu zakresu obu tych części. Każdy składnik jest następnie przedmiotem oddzielnego pomiaru, w którym traktuje się go jako *użytkownika* części pozostałej.

### Wnioski

- 1) W praktyce często mamy do czynienia właśnie z przypadkiem 2, tj. z przypisywaniem funkcjonalnych wymagań użytkownika do wielu części oprogramowania;

- 2) W sytuacji względnie wczesnego wyznaczenia w procesie projektowania architektury systemu (por. rys. 5 i rys. 6), przypisanie WFU do określonych części oprogramowania w tej architekturze ma wpływ na rozmiar funkcjonalny każdej części;
- 3) Do wyznaczenia ogólnego globalnego rozmiaru wymagań funkcjonalnych nie jest konieczne uwzględnienie sposobu przypisywania WFU (Przypadek 1);
- 4) W przypadku konieczności wyznaczenia rozmiaru funkcjonalnego poszczególnych części wymagań (Przypadek 2) należy wykorzystać schemat przedstawiony na rys. 6, jako że jest on odpowiedni do identyfikacji granic oprogramowania.

W związku z powyższym metoda *COSMIC-FFP* oferuje narzędzie pozwalające na rozróżnienie WFU przypisywanych różnym poziomom wyodrębnienia funkcji. Jest ono związane z koncepcją tzw. *warstwy oprogramowania*.

### Definicja warstwy oprogramowania

Jak zaznaczono powyżej, w klasyfikacji systemów informatycznych duże znaczenie przypisuje się rodzajowi użytkownika, czyli odbiorcy funkcjonalności oferowanej przez poszczególne typy systemów. Takim odbiorcą może być zarówno człowiek, jak i inne oprogramowanie lub urządzenia mechaniczne. Metoda zwykłych punktów funkcyjnych *IFPUG* może uwzględnić tylko ten rodzaj funkcjonalności, której bezpośrednim odbiorcą jest człowiek. Z punktu widzenia takiej osoby bowiem funkcjonalność na przykład systemów czasu rzeczywistego, graficznego interfejsu użytkownika, czy oprogramowania infrastrukturalnego będzie pomijana, jako że jest dla niego niezauważalna w sposób bezpośredni.

Tymczasem okazuje się, że w celu dostarczenia funkcjonalności różnym użytkownikom różnych kategorii SI często należy wykorzystać funkcjonalność innych rodzajów oprogramowania – nie tylko oprogramowania aplikacyjnego, co do funkcjonalności którego użytkownik będący człowiekiem zwykle nie ma wątpliwości. Tę pozostałą funkcjonalność zatem należy również mierzyć i estymować, gdyż bez niej nie byłoby często możliwe działanie oprogramowania aplikacyjnego (por. Przypadek 2). W związku z tym, z punktu widzenia różnych poziomów funkcjonalności, istotne staje się pojęcie *warstwy oprogramowania*, przez którą rozumie się *"rezultat podziału środowiska oprogramowania z punktu widzenia funkcjonalnego, tak aby wszystkie wewnętrzne procesy funkcjonalne zachodziły na tym samym poziomie wyodrębnienia"* [2, s. 8]. Przy czym przez *proces funkcjonalny* rozumie się *"unikalny zbiór wejść, wyjść, odczytów i zapisów będący implementacją spójnego i niepodzielonego zbioru WFU"* [2, s. 7]. *"W środowisku oprogramowania wielowarstwowego, oprogramowanie jednej warstwy wymienia dane z oprogramowaniem innej warstwy poprzez odpowiednie dla nich procesy funkcjonalne. Te interakcje są z natury hierarchiczne (...): warstwa, nazywana "klientem" wykorzystuje funkcjonalność innych podporządkowanych warstw"* [2, s. 7 i 8].

Warstwa oprogramowania stanowi zatem pewną niezależną część funkcjonalną składającą się na strukturę oprogramowania. Elementy

oprogramowania bowiem są właśnie ustrukturalizowane w poziome warstwy, z których każda stanowi świat sam w sobie, co sprowadza się do następujących cech:

- Dla danej warstwy nie jest istotny sposób generowania jej wejść, a jedynie sposób właściwego z nimi postępowania. Nie jest ważne również to, co dzieje się dalej z danymi wyjściowymi.
- Każda warstwa dostrzega warstwy poniżej siebie, postrzegając je jako zbiór wyrażań pierwotnych, ale nie może zobaczyć warstw znajdujących się powyżej niej.
- Wewnętrzna operacja danej warstwy nie musi być znana przez żadną inną warstwę. W konsekwencji wewnętrzne szczegóły danej warstwy są chronione przed zewnętrznymi zmianami. Uzależnienie jednej warstwy od wewnętrznej organizacji innej warstwy nie jest właściwe - taka zależność bowiem ogranicza możliwości zmian lub doskonalenia warstwy zależnej.
- Systemy zarządzania bazami danych, interfejsy graficzne użytkownika, systemy operacyjne oraz sterowniki urządzeń są zazwyczaj uważane za odrębne warstwy.
- Poziom aplikacji jest zwykle uważany za warstwę najwyższego poziomu.

#### Warunki identyfikacji warstwy

1. Oprogramowanie we wszystkich warstwach dostarcza funkcjonalność użytkownikom (człowiekowi, urządzeniu lub innemu oprogramowaniu);
2. Oprogramowanie w warstwie podrzędnej dostarcza funkcjonalność oprogramowaniu z warstw nazywanych "klientami";
3. Oprogramowanie w warstwie podporządkowanej może funkcjonować bez oprogramowania warstwy nadrzędnej.
4. Oprogramowanie w warstwie nadrzędnej może nie funkcjonować prawidłowo, jeżeli oprogramowanie w warstwie podporządkowanej nie funkcjonuje prawidłowo;
5. Oprogramowanie warstwy nadrzędnej nie musi wykorzystywać całej funkcjonalności oferowanej przez oprogramowanie warstwy podrzędnej;
6. Oprogramowanie w warstwie "klienta" i w warstwie podporządkowanej może dzielić między sobą i wspólnie wymieniać dane, jednakże oprogramowanie każdej warstwy interpretuje te dane w różny sposób;
7. Oprogramowanie współdzielące dane z innym oprogramowaniem nie może być uznane za należące do innej warstwy, jeżeli interpretuje współdzielone dane w identyczny sposób.

Metoda zwykłych punktów funkcyjnych nie jest zatem właściwa dla tych kategorii SI, które składają się z wielu warstw - nie uwzględnia bowiem funkcjonalności warstw znajdujących się poniżej warstwy położonej na najwyższym poziomie i oferującej funkcjonalność bezpośrednio człowiekowi.



## Charakterystyki modelu kontekstowego COSMIC-FFP

### ➤ **Charakterystyka 1 – Warstwy:**

Oprogramowanie, które ma być przedmiotem pomiaru może składać się z jednej lub wielu części, przy czym każda część operuje na różnym poziomie wyodrębnienia funkcji (oczywiście w określonym środowisku oprogramowania). Pomiędzy każdym poziomem występuje zależność hierarchiczna wynikająca z wymiany danych między nimi. Każdy taki poziom jest właśnie uważany za oddzielną warstwę. Każda warstwa obejmuje funkcjonalność użyteczną dla tej warstwy przy wykorzystaniu zależności hierarchicznej, tj. zarówno swojej funkcjonalności, jak i funkcjonalności dostarczanej przez warstwę podrzędną w tym związku.

### ➤ **Charakterystyka 2 – Granice:**

W każdej warstwie część oprogramowania, która podlega pomiarowi, może być w prosty sposób wyodrębniona spośród innych otaczających ją warstw poprzez ustanowienie jej granic. Domniemane granice bowiem istnieją pomiędzy każdą zidentyfikowaną warstwą. Granice ustanawia się na podstawie zbioru kryteriów, postrzeganych z punktu widzenia wymagań funkcjonalnych użytkownika co do oprogramowania, które pozwalają na jasne określenie elementów stanowiących część mierzonego oprogramowania (wewnątrz granic) oraz elementów stanowiących składniki środowiska (na zewnątrz granic). W praktyce przyjmuje się zasadę, iż wszyscy użytkownicy danej części oprogramowania (np. inne oprogramowanie) znajdują się poza jego granicami.

### ➤ **Charakterystyka 3 – Użytkownicy oprogramowania:**

Identyfikacja jednego bądź wielu użytkowników wykorzystujących funkcjonalność dostarczaną przez tę część oprogramowania, która stanowi warstwę, odbywa się na podstawie definicji użytkownika przyjętej w metodzie COSMIC-FFP. Przypomnijmy, iż wynika z niej, że użytkownikiem może być człowiek, urządzenie lub inne oprogramowanie. Części oprogramowania znajdujące się w bezpośrednio sąsiadujących warstwach są właśnie dlatego uważane za użytkowników, o ile oczywiście wchodzi w interakcje z tą częścią, która jest przedmiotem pomiaru.

### ➤ **Charakterystyka 4 – Wymagania funkcjonalne użytkownika:**

Wymagania odnośnie oprogramowania opisujące naturę funkcji, które mają być dostarczane przez owo oprogramowanie są wyznaczane jako WFU (por. pkt 4). Są one wykorzystywane jako perspektywa, z jakiej mierzy się rozmiar funkcjonalny. Jednak te części wymagań, które opisują sposób implementacji funkcji (np. jakościowe lub techniczne) nie są brane pod uwagę przy pomiarze rozmiaru oprogramowania. Jest to zgodne ze standardem ISO.

## **5.2. Model oprogramowania COSMIC-FFP**

Model ów zakłada, iż dla oprogramowania będącego przedmiotem pomiaru prawdziwe są następujące generalne zasady:

### ➤ **Zasada 1:**

**Oprogramowanie będące przedmiotem odwzorowania i pomiaru jest zasilane przez wejście (ang. *input*) i generuje użyteczne dla użytkowników wyjście (ang. *output*).**

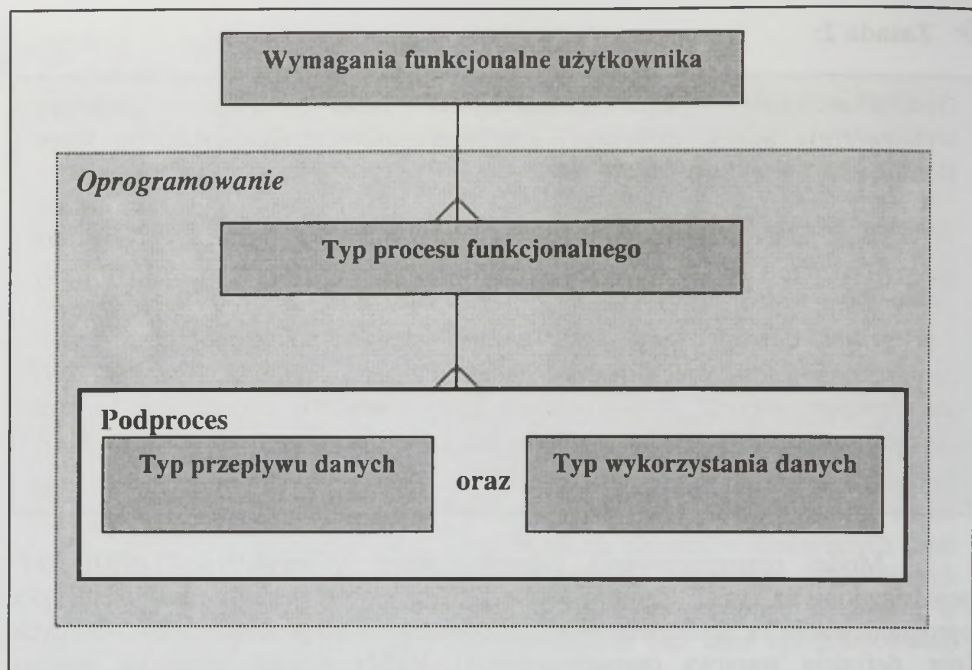
➤ **Zasada 2:**

Oprogramowanie będące przedmiotem odwzorowania i pomiaru wykorzystuje porcje informacji określane mianem grup danych, które składają się z atrybutów danych.

Przy czym przez *atrybut danych* (ang. *data attribute*, inaczej: *typ elementu danych*) rozumie się "najmniejszą porcję informacji w ramach zidentyfikowanej grupy danych, mającą znaczenie z perspektywy WFU dla oprogramowania". Zaś *grupa danych* (ang. *data group*) stanowi "wyróżniony, niepusty, nieuporządkowany i nieredundantny zbiór atrybutów danych, w którym każdy uwzględniony atrybut danych opisuje komplementarny aspekt tego samego obiektu zainteresowania, zidentyfikowanego z perspektywy WFU" (np. encji) [2, s. 7].

Model oprogramowania zaproponowany w metodzie *COSMIC-FFP* przedstawiono na rys. 7. Zgodnie z tym schematem wymagania funkcjonalne dla oprogramowania są implementowane za pomocą zbioru procesów funkcjonalnych (por. definicję warstwy oprogramowania). Każdy z tych procesów stanowi unikalny zbiór podprocesów polegających albo na przepływie danych, albo na ich wykorzystaniu w ramach oprogramowania. Przy czym przez *przepływ danych* rozumie się "podproces, wyróżniony w mierzonym oprogramowaniu, który powoduje ruch co najmniej jednego atrybutu danych należącego do pojedynczej grupy danych". Jak wynika z modelu kontekstowego, wyróżnia się cztery typy (podprocesy) takiego przepływu, które przemieszczają dane zawarte w pojedynczej grupie danych (por. rys. 4 oraz [2, s. 7, 9]):

- **Wejście** (ang. *entry*) – przepływ atrybutów danych z jednej grupy danych w kierunku od strony użytkownika do wnętrza oprogramowania - nie aktualizuje danych, które przemieszcza;
- **Wyjście** (ang. *exit*) – przepływ atrybutów danych z jednej grupy danych w kierunku od wnętrza oprogramowania do użytkownika – nie odczytuje danych, które przemieszcza;
- **Odczyt** (ang. *read*) – podproces odczytujący atrybuty danych z jednej grupy danych, sprowadzający (przemieszczający) dane z pamięci do wnętrza oprogramowania;
- **Zapis** (ang. *write*) – podproces zapisujący atrybuty danych w jednej grupie danych, wysyłający (przemieszczający) dane z wnętrza aplikacji do pamięci.



Rys. 7. Ogólny model oprogramowania wykorzystywany do pomiaru rozmiaru funkcjonalnego<sup>4</sup>  
 Źródło: [2, s. 18]

Wykorzystanie definicji i struktury zaproponowanej przez autorów opisywanej metody umożliwia odwzorowanie WFU wyznaczonych na podstawie modelu przed- lub post-wdrożeniowego (por. rys. 2 i rys. 3) w model oprogramowania, który obejmuje wszystkie elementy wymagane dla pomiaru rozmiaru funkcjonalnego, jednocześnie ukrywając te, które dla takiego pomiaru znaczenia nie mają. Model oprogramowania *COSMIC-FFP* jest następnie podstawą do obliczenia wartości, oczywiście przy zastosowaniu określonych reguł i procedur, która reprezentuje rozmiar funkcjonalny oprogramowania. Zatem w celu wyprowadzenia owego rozmiaru konieczne jest przeprowadzenie dwóch oddzielnych, ale powiązanych ze sobą, faz (por. rys. 1):

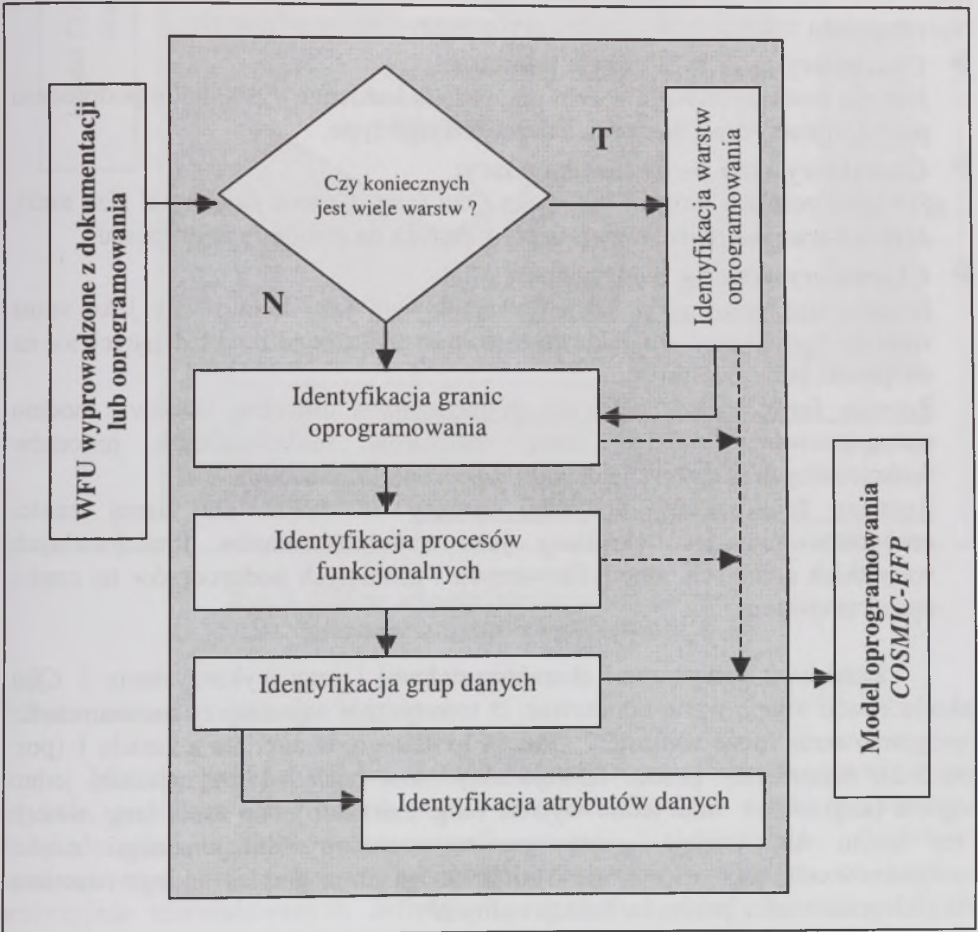
- Fazy odwzorowania WFU w indywidualny model oprogramowania, przedstawionej w sposób schematyczny na rys. 8, do której wejście stanowią

<sup>4</sup> Przy czym w obecnej wersji metody *COSMIC-FFP* podprocesy określane mianem typów wykorzystania danych są uwzględniane w ramach definicji dla przepływu danych, ponieważ definicje niezbędne do pomiaru wykorzystania danych są ciągle przedmiotem dyskusji.



WFU ustalone na podstawie dokumentacji (por. rys. 2) lub wyprowadzone z funkcjonującego oprogramowania (por. rys. 3);

- Fazy pomiaru specyficznych elementów owego modelu oprogramowania, do której wejście stanowi rezultat fazy poprzedniej. Ów model oprogramowania *COSMIC-FFP* jest zwykle przedstawiany w postaci macierzy, w której wiersze reprezentują procesy funkcjonalne, często pogrupowane ze względu na warstwy, zaś kolumny opisują grupy danych. W takim układzie w punktach przecięcia umieszcza się zidentyfikowane podprocesy. Krótki opis tej fazy przedstawiono poniżej.



Rys. 8. Ogólny schemat przebiegu fazy odwzorowania WFU  
Źródło: Opracowanie własne na podstawie [2, s. 22]

## 6. Faza pomiaru w metodzie COSMIC-FFP

Podstawowa zasada pomiaru w metodzie COSMIC-FFP jest następująca:

**Rozmiar funkcjonalny oprogramowania jest wprost proporcjonalny do liczby jego podprocesów przepływu danych.**

Wśród charakterystyk zbioru zasad i procedur rządzących fazą pomiaru (por. rys. 9), w rezultacie której wyprowadzana jest wartość liczbową reprezentująca rozmiar funkcjonalny oprogramowania, wyróżnia się:

➤ **Charakterystyka 1 - Funkcja pomiaru:**

Jest ona wykorzystywana w celu przypisania każdemu wystąpieniu podprocesu pewnej numerycznej wartości, zależnej od jego typu.

➤ **Charakterystyka 2 - Jednostka miary:**

Standard pomiaru stanowi jednostka *Cfsu* (ang. *Cosmic functional size unit*), definiowana jako pojedynczy przepływ danych na poziomie podprocesu<sup>5</sup>.

➤ **Charakterystyka 3 - Addytywność:**

Rozmiar funkcjonalny procesu funkcjonalnego jest definiowany jako suma wartości funkcji pomiaru, jakie zastosowano do każdego ze składających się na ów proces podprocesów.

Rozmiar funkcjonalny części oprogramowania w dowolnej warstwie modelu oprogramowania stanowi sumę rozmiarów funkcjonalnych procesów funkcjonalnych realizowanych w tej części oprogramowania.

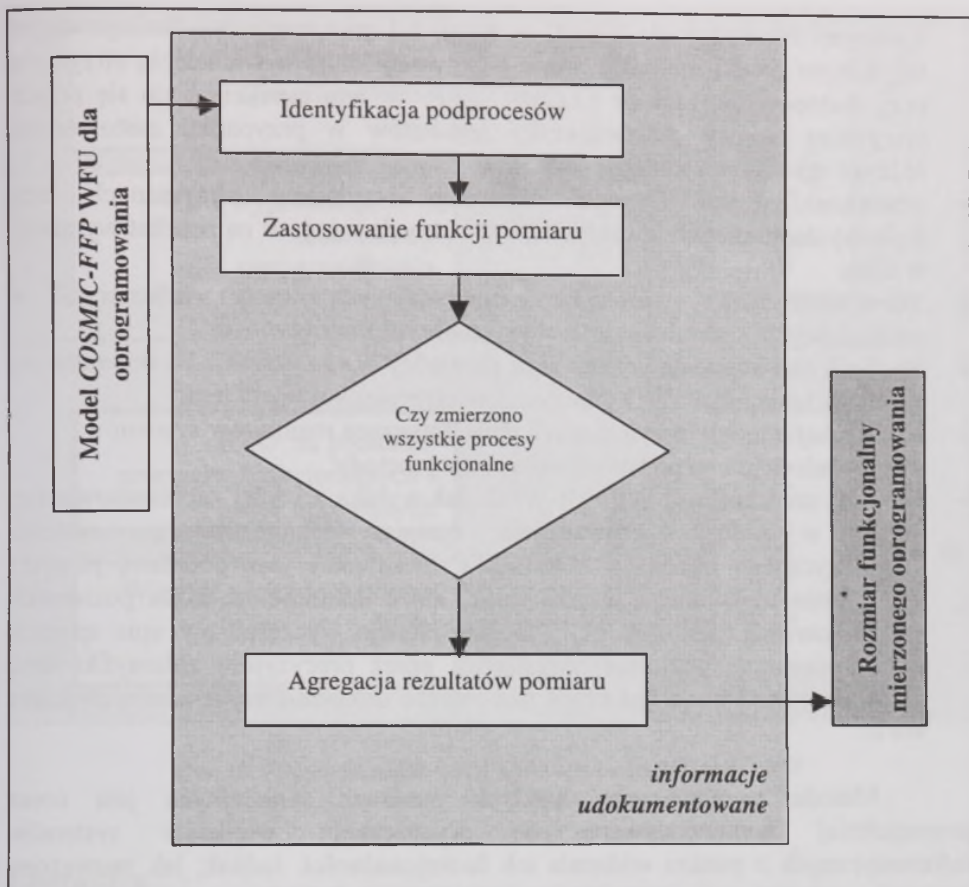
Rozmiar funkcjonalny dowolnej zmiany w WFU dla danej części oprogramowania jest określany jako suma rozmiarów funkcjonalnych wszystkich dodanych, zmodyfikowanych i usuniętych podprocesów tej części oprogramowania.

Zgodnie z powyższymi charakterystykami i przy wykorzystaniu 1 *Cfsu* jako jednostki miary, warto odnotować, iż teoretycznie najmniejszy rozmiar części oprogramowania może wynosić 2 *Cfsu*. A to dlatego, iż zgodnie z zasadą 1 (por. pkt 5.2), najmniejszy proces funkcjonalny musi posiadać przynajmniej jedno wejście (ang. *entry*) oraz jedno wyjście (ang. *exit*) lub jeden zapis (ang. *write*). Przy czym nie istnieje górna granica rozmiaru funkcjonalnego części oprogramowania, a co więcej, nie istnieje nawet górna granica takiego rozmiaru dla jakiegokolwiek z procesów funkcjonalnych<sup>6</sup>.

---

<sup>5</sup> Obecnie trwają prace nad wyróżnieniem podjednostki, którą miałyby stanowić przepływy pojedynczego atrybutu danych.

<sup>6</sup> Dokładny opis reguł i zasad wykorzystywanych do wyprowadzenia rozmiaru funkcjonalnego oprogramowania znajdzie czytelnik w [COSM01].



Rys. 9. Ogólny schemat przebiegu fazy pomiaru  
 Źródło: Opracowanie własne na podstawie [2, s. 33]

## 7. Posumowanie

Metoda, której podstawowe idee i zasady zaprezentowano w niniejszym artykule, powstała jako rezultat ponad dwudziestoletniego międzynarodowego doświadczenia w stosowaniu innych metod funkcjonalnych. Czytelnik zainteresowany szczegółowym sposobem postępowania, opartym na bardzo precyzyjnie sformułowanych definicjach, regułach i zasadach może znaleźć dokładne informacje w cytowanym tu niejednokrotnie dokumencie, pochodzącym z połowy ubiegłego roku (por. [2]). Dokument ten przedstawia bieżącą wersję opisywanej metody, której jednak nie uznaje się jeszcze za jej wersję ostateczną i pozbawioną błędów. Obecnie bowiem ciągle trwają prace nad udoskonaleniem opisywanego sposobu postępowania. Skupiają się one w szczególności na:

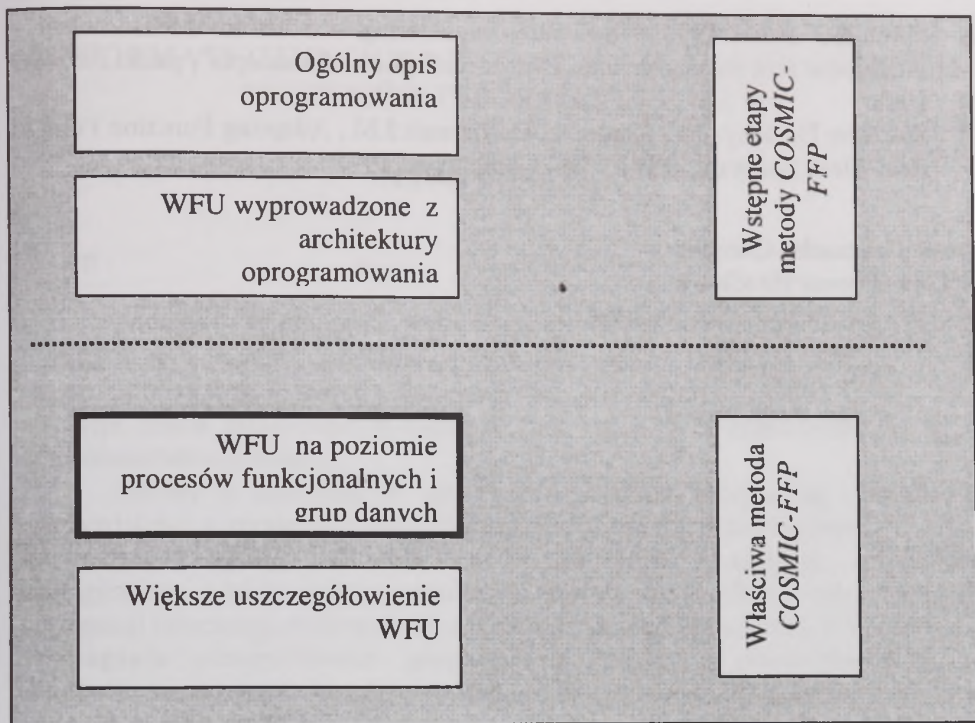
- Testowaniu spójności między różnymi miarami rozmiaru funkcjonalnego SI, w tym przede wszystkim spójności pomiędzy zwykłymi punktami funkcyjnymi



zgodnymi z metodologią *IFPUG* w wersji 4.1 oraz rozmiarem funkcjonalnym uzyskanym dzięki metodzie *Mark II* w wersji 3.1.1 a wielkością otrzymaną przy zastosowaniu metody *COSMIC-FFP*. W tym aspekcie bada się przede wszystkim stopień powtarzalności rezultatów w przypadku zastosowania różnych miar dla tej samej specyfikacji;

- Stworzeniu zasad i reguł konwersji rezultatów otrzymanych przy wykorzystaniu metody zwykłych PF w różnych wersjach na rezultat wyrażony w *Cfsu*;
- Stworzeniu zasad i procedur umożliwiających ocenę wielkości SI w początkowych stadiach cyklu życia systemu informatycznego;
- Studiach nad korelacją występującą pomiędzy oceną eksperta co do wielkości rozmiaru funkcjonalnego a rzeczywiście otrzymanym wynikiem;
- Konstrukcji modeli utrzymujących dane dotyczące rozmiarów systemów wyprowadzonych na podstawie opisywanej metody;
- Bardziej szczegółowej definicji WFU. Jak wynika z wyżej zaprezentowanego opisu, w celu wyprowadzenia rozmiaru funkcjonalnego systemu informatycznego metoda *COSMIC-FFP* uwzględnia jako podstawę pomiaru wymagania funkcjonalne użytkownika, które definiuje na kilku poziomach szczegółowości (por. rys. 10). Obecnie istnieje wyczerpujący opis sposobu postępowania na poziomie określonym przez precyzyjnie zidentyfikowane grupy danych. Trwają zaś prace nad jeszcze dokładniejszym zdefiniowaniem WFU.

Metoda bazująca na zwykłych punktach funkcyjnych jest coraz powszechniej wykorzystywana do wyznaczania wielkości systemów informatycznych z punktu widzenia ich funkcjonalności. Jednak, jak zauważono powyżej, stanowią one wartościowe rozwiązanie jedynie w odniesieniu do systemów informatycznych zarządzania, z myślą o których powstała. Funkcjonalność innych kategorii SI nie może być w pełni uwzględniona w ostatecznej liczbie takich punktów ze względu na odmienny charakter tych aplikacji. Stąd ciągłe poszukiwania nowego, w pełni uniwersalnego i zgodnego ze standardami *ISO*, rozwiązania, które z jednej strony wykorzystywałaby logikę klasycznej metody punktów funkcyjnych, z drugiej zaś umożliwiałaby prawidłowy pomiar i estymację rozmiaru funkcjonalnego tych systemów, których podstawowym zadaniem nie jest wspomaganie zarządzania. Spośród wielu różnych wariantów analizy opartej na PF najbardziej obiecującą w tym względzie koncepcję stanowi właśnie przedstawiona powyżej technika *pełnych punktów funkcyjnych* opracowana przez organizację *COSMIC*.



Rys. 10. Poziom szczegółowości WFU w metodzie *COSMIC-FFP*

Źródło: Opracowanie własne na podstawie [2, s. 21]

## Literatura

1. Allan Albrecht., Measuring Application Development Productivity, Proceedings IBM Application Development Symposium, Monterey, CA., Oct 14-17, 1979.
2. COSMIC-FFP Measurement Manual, version 2.1, Common Software Measurement International Consortium, May 2001.
3. Czarnacka-Chrobot Beata, Porównanie metod pomiaru i szacowania projektów informatycznych – jednostki programowe a jednostki umowne, w: Efektywność zastosowań systemów informatycznych, tom I, materiały konferencyjne XIII Szkoły Górskiej PTI pod red. J. K. Grabary i J. S. Nowaka, WNT Warszawa – Szczyrk 2001, s. 27 – 55.
4. Czarnacka-Chrobot Beata, Metoda punktów funkcyjnych – bieżące standardy, w: Efektywność zastosowań systemów informatycznych, tom I, materiały konferencyjne XIII Szkoły Górskiej PTI pod red. J. K. Grabary i J. S. Nowaka, WNT Warszawa – Szczyrk 2001, s. 57 – 83.
5. Tom DeMarco, Controlling Software Projects, New York, Yourdon Press, 1982.

6. "ISO/IEC 14143-1998 – Software Engineering – Software Measurement – Functional size measurement – Part 1: Definition of concepts", punkt 3.8, ISO 1998.
7. St-Pierre D., Maya M., Abran A., Desharnais J.M., Adapting Function Point to Real-Time Software, IFPUG Fall Conference, 1997.

Beata Czarnacka-Chrobot  
Szkoła Główna Handlowa  
Katedra Informatyki Gospodarczej  
Al. Niepodległości 162  
02-554 Warszawa  
e-mail: bczarn@sgh.waw.pl



# WIZUALIZACJA INFORMACJI W SYSTEMACH INFORMATYCZNYCH NA PRZYKŁADZIE LOGISTYCZNYCH SYSTEMÓW INFORMATYCZNYCH

Iwona GRABARA

## Wstęp

Zmiany w otoczeniu, które inspirują do poszukiwania ciągłych nowych rozwiązań w zakresie zarządzania przedsiębiorstwem, zwłaszcza teraz kiedy w obliczu globalizacji i silnej konkurencji we wszystkich sektorach rynku, trwa ewolucja metod zarządzania w kierunku stosowania nieskomplikowanych, wielu komplementarnych metod.

Zmiany te zachodzą w głównych obszarach, którymi są wymagania rzeczywistości gospodarczej, logistyka i jej wspomaganie przez logistyczny system informatyczny (LIS), co stało się nieodzownym elementem zarządzania strategicznego, a także zmiana mentalności decydentów w odniesieniu do sposobu prezentacji informacji analitycznych oraz rozwój metod wizualizacji w biznesie.

**Wymagania rzeczywistości gospodarczej** zmuszają przedsiębiorstwa do zwrócenia uwagi na integrację wewnętrznych procesów z otoczeniem poprzez myślenie w skali globalnej, nastawianie się na rynki światowe, ciągłe poszukiwanie rozwiązań innowacyjnych w zakresie metod zarządzania jak i w zakresie wspomaganie podejmowania decyzji, modyfikowanie działalności w zależności od zmieniającego się otoczenia, charakteru rynku i rozwijającej się konkurencji. Dominującym determinantem zmian w polityce i strategii przedsiębiorstwa jest ukierunkowanie się na klienta. Takie zmiany możliwe są jedynie przy zastosowaniu nowoczesnych metod zarządzania, organizacji jak i oczywiście maksymalnego wykorzystania technologii informacyjnej w celu podniesienia efektywności i skuteczności działania przedsiębiorstwa.

## Logistyka i logistyczny system informatyczny (LIS)

To właśnie logistykę i systemy ją wspomagające należy traktować jako koncepcję strategicznego zarządzania przedsiębiorstwem. Przedsiębiorstwo powinno działać jako otwarty system zarządzający wszystkimi procesami gospodarczymi. Dzięki przepływowi w systemach logistycznych dwóch równoległych strumieni informacyjnych następuje integracja wszystkich procesów gospodarczych wewnątrz przedsiębiorstwa i tworzą się powiązania z innymi współpracującymi przedsiębiorstwami, z innymi systemami informacyjnymi oraz z całym otoczeniem.

Logistyka w przedsiębiorstwie powinna integrować wiele rodzajów działalności i nastawiona powinna być na osiągnięcie maksymalnej płynności i elastyczności procesu przepływu wszelkich dóbr wraz z informacjami. Natomiast

jakość podejmowanych decyzji logistycznych wpływają na kondycję całego przedsiębiorstwa.

Zmiany zachodzące w otoczeniu wpływają także na zmiany zachodzące w dziedzinie logistyki. Parametry charakterystyczne dla logistyki, które ulegają zmianom a mające znaczący wpływ na sukces firmy to: szybkość/czas, jakość, zwiększanie produktywności aktywów, reinżynieria organizacji, zadowolenie klienta.

## Zmiany mentalności decydentów i analityków

**Mentalność menedżerów** zwłaszcza młodych w odniesieniu do sposobu odbioru i prezentacji informacji analitycznych. Szczególnie młodego pokolenia uległa zmianie. Młodzi ludzie od najwcześniejszych lat stykają się z obrazem. Związane jest to między innymi ze zmianami programów nauczania, nowymi podręcznikami, z dostępem do komputerów i oprogramowania graficznego, z nabywaniem umiejętności szybkiego czytania i zapamiętywania, gdzie wykorzystywana jest wyobraźnia i zdolność łączenia zapamiętywanych informacji z wyobrażanym sobie obrazem.

Simulationsergebnisse des Materialflusssystems Linia

\*\*\*\*\*

Intervallstatistik : Bausteinleistungsgrößen  
 Materialfluss - System : Linia  
 Statistik fuer den Zeitraum : 0.0 - 15.0

Nr.	Baustein- name	Durch- satz	Akt. Inh.	Max. Inh.	mittl. Bel.	prozent. Belegung	Aus- lastung
1	PRAS_1	2155	96	96	11.98		54.46
30	DEKO_1	1616	1	1	0.91	90.73	90.73
41	NECK_1	1514	1	1	0.78	78.27	78.27
42	OUT_1	1500	1	1	0.67	66.67	66.67
44	LFB_1	1799	416	416	265.03	63.71	82.93
45	BODY_1	1739	1	1	0.97	97.00	97.00
47	LNO_1	1501	13	14	11.73	10.66	10.66
49	LED_1	1617	121	122	143.59	13.18	19.69
50	LIN_1	1616	101	101	88.73	11.30	11.30

\*\*\*\*\*

Intervallstatistik : Von Quellen abgegebene Objekte  
 Materialfluss - System : Linia  
 Statistik fuer den Zeitraum : 0.0 - 15.0

Rys.1 Standardowy sposób prezentacji danych w systemach informatycznych [9]

Współczesne pokolenie analityków i decydentów prezentuje nieco inny styl myślenia, rozumie informacje przekazywane za pomocą obrazów, na ich

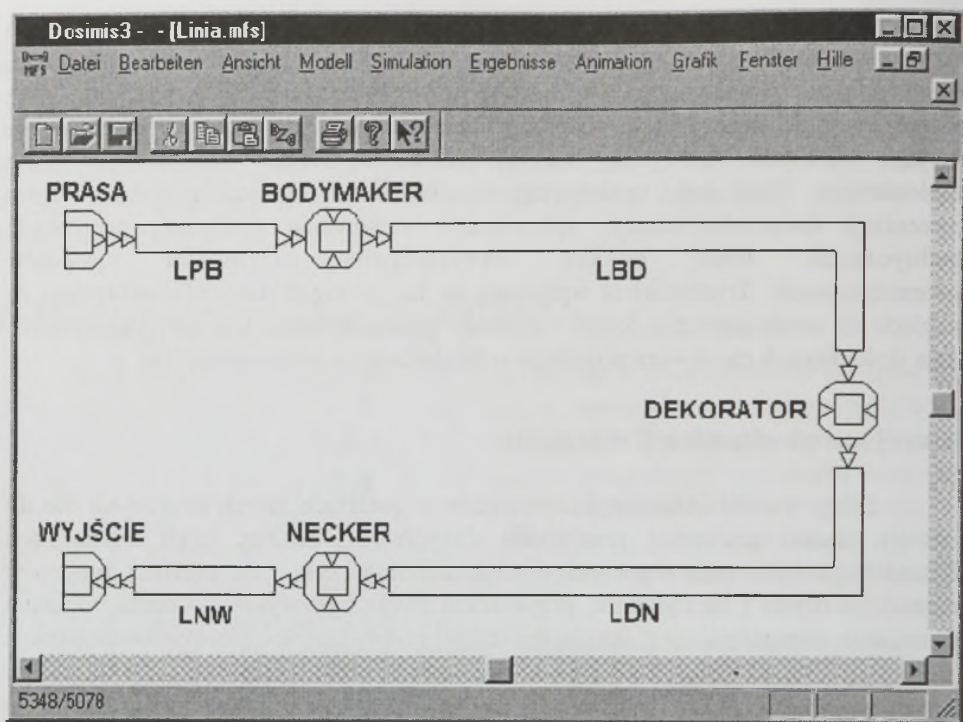
podstawie potrafią kojarzyć odpowiednie zjawiska, rozumieją problemy do rozwiązania, szybko wyciągają prawidłowe wnioski i umieją podejmować decyzje. Niestety ludzie ci nie mogą się specjalnie wykazać swoimi umiejętnościami, gdyż w tradycyjnych prezentacjach danych/informacji wspomagających podejmowanie decyzji, opartych zazwyczaj na urządzeniach znakowych, istnieje wiele niedostatków. Niedostatki te dotyczą: stosowania tradycyjnych sposobów i form prezentacji danych/informacji, sporadyczne stosowanie graficznej prezentacji analitycznych, niski poziom wykorzystania możliwości systemów informatycznych. Trudności te wpływają na to, że część danych i informacji ze względu na niedostateczne formy i sposoby przekazywania jest niewykorzystana, duża ilość danych na zawsze pozostaje w bazie danych przedsiębiorstw.

### **Rozwój metod wizualizacji w biznesie**

Silny rozwój informatyki, zwłaszcza w ostatnich latach przyczynił się do rozwoju metod graficznej prezentacji danych i informacji czyli wizualizacji. Wizualizacja była stosowana od dawna, ale dotyczyła ona głównie procesów technologicznych i zarządzania, przykładem mogą być wykresy Gantta. Obecnie stosowanie wizualizacji jest łatwiejsze dzięki nowej technologii. Ponieważ istnieje konieczność stosowania takich metod i form prezentacji danych, zwłaszcza w dostarczaniu analitykom i decydentom na każdym poziomie organizacyjnym informacji czytelnych, precyzyjnych, konkretnych, pozwalających na przeprowadzanie analiz, selekcji, spostrzeganie szczególnie ważnych zjawisk i faktów, dlatego wizualizację należy wykorzystywać jako metodę wspomagającą podejmowanie decyzji.

Dzięki wizualizacji można dostosować sposób prezentacji informacji do możliwości percepcyjnych człowieka, który największą ilość informacji dostarczonych do centralnego układu informacji bo około 87% odbiera przy pomocy wzroku, dla porównania, za pomocą słuch odbiera ok. 10%, a za pomocą innych kanałów takich jak: dotyk, smak, węch, ból, ciepło itp. zaledwie 3%. Badania te przeprowadzili w latach 90-tych H. Nieman, R. DeMori, G. Handrider, przedstawili oni także percepcję informacji u człowieka w zależności od formy odbioru informacji i tak przy czytaniu tekstu percepcja informacji wynosi 150-300 bit/s, przy odczytywaniu obrazu ok.  $10^6$  bit/s, a przy słuchaniu wypowiedzi ok.  $10^6$  bit/s. Oczy są głównym wejściem ludzkiego rozumu, dlatego nowe paradygmaty zastosowania komputerów często dotyczą szukania i projektowania lepszych rozwiązań odwzorowań graficznych i ożywionych interfejsów.





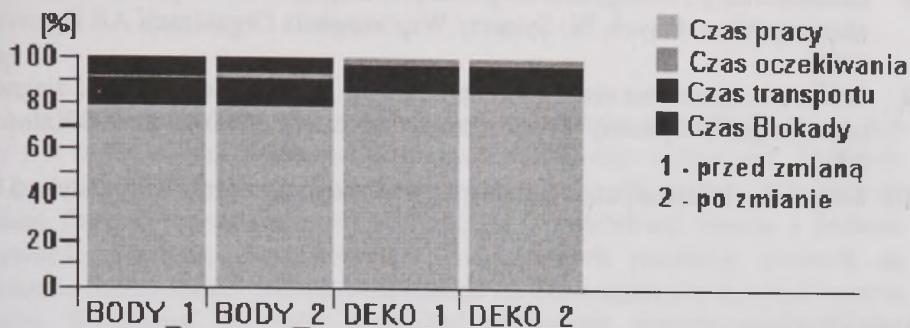
Rys2. Przykład wizualizacji procesów produkcyjnych [9]

Wizualizacja jako interdyscyplinarna dziedzina obejmuje takie dyscypliny naukowe jak: psychologię, informatykę, ekonomię, ergonomię.

Poza tym wizualizacja wymaga rzetelnej wiedzy z zakresy takich obszarów jak:

- ✓ grafika komputerowa,
- ✓ przetwarzanie obrazów,
- ✓ przetwarzanie znaków,
- ✓ projektowanie wspomagane komputerem,
- ✓ badania ergonomiczne interfejsu użytkownika.

### Histogram pracy Bodymakera i Dekoratora



Rys.3. Przykład prostej prezentacji wizualnej

### Podsumowanie

Istnieją uwarunkowania przemawiające za stosowaniem graficznej prezentacji danych/informacji, są nimi: czas, szum informacyjny, złożoność istniejących zjawisk i rozwiązywanych problemów, potrzeby współczesnych analityków i decydentów. Podstawnym jest zatem stosowanie wizualizacji informacji analitycznych w LSI, a tym samym rozwiązanie tego problemu w proponowanej pracy. Tym bardziej, że wizualizacja posiada wiele zalet, które korzystnie wpływają na proces podejmowania decyzji, zaletami tymi są: zainteresowanie odbiorcy prezentowanym materiałem, możliwość prezentacji w sposób uproszczony, skondensowany, czytelny, precyzyjny, zwrócenie uwagi na szczególnie ważne informacje, możliwość zauważenia prawidłowości występowania różnych zjawisk oraz ustalenie związków między nimi na podstawie szybkiej analizy i syntezy, możliwość wykrycia przyczynowych zależności między wieloma zjawiskami.

### Literatura

1. Abt S., Zarządzanie logistyczne w przedsiębiorstwie, PWE Warszawa 1998.
2. Abt S., Woźniak H. Podstawy logistyki, Uniwersytet Gdański 1999.
3. Adamczewski P., Zintegrowany system informatyczny, Logistyka 4/99,5/99 .
4. Borowiecki R., Romanowska M. Pr.zb. Systemy informacji strategicznej, Difin, Warszawa 2001.
5. Bukowski L., Inteligentne systemy wspomaganie decyzji logistycznych, Pr. Wyzd. Zarządzania Politechniki Częstochowskiej, 1999.
6. Gołębska E., Szymczyk M. Informatyzacja w logistyce przedsiębiorstw, PWN Warszawa 1997.

7. Gołuchowski J., Inteligentne systemy diagnoz ekonomicznych, Akademia Ekonomiczna Katowice 1987.
8. Gołuchowski J., Inteligentna diagnoza przedsiębiorstwa w systemach zbiorczych baz danych, W: Systemy Wspomagania Organizacji AE Katowice 1999.
9. Grabara I., Skuteczne rozwiązania prezentacji graficznej do analizy i diagnozy organizacji, W: Systemy Wspomagania Organizacji SWO'2000, AE Katowice 2000.
10. Korzeń Z., Informatyczne systemy logistycznego zarządzania, Logistyka 3/96.

Iwona Grabara  
Instytut Informatyki i Ekonometrii  
Wydział Zarządzania; Politechnika Częstochowska  
Ul. Armii Krajowej 19b  
42-200 Częstochowa



# ANALIZA PROCESÓW PRACY

Wojciech KULIK, Zdzisław SZYJEWSKI

## Wstęp

Poszukiwania nowych, efektywniejszych struktur organizacji pracy jest ciągły i w dużej mierze zależy od aktualnych możliwości technologii. Aktualnie wiele organizacji preferuje procesową organizację pracy. Procesowe podejście do realizacji funkcji przedsiębiorstwa jest zgodne z potrzebami klienta i podnosi efektywność działania firmy. Analiza realizowanych procesów pozwala na wyszukanie oszczędności oraz szybszą reakcję na zmieniające się zapotrzebowanie klientów. Procesowe podejście może być efektywnie wspomagane środkami technologii informatycznej.

Systemy do obsługi procesów pracy umożliwiają zaplanowanie organizacji obsługi zdarzeń związanych z przebiegiem procesu gospodarczego, na przykład obsługą klienta przez zespół firmy. Proces biznesowy jest zaplanowany w czasie i poddany rygorom dyscypliny formalnej. Informacja generująca proces zostaje zarejestrowana, przesłana w odpowiednim czasie na odpowiednie stanowisko, powiązana z informacją archiwalną, i co najważniejsze – wiadomo kiedy, kto i co powinien z nią dalej zrobić, aby osiągnąć cele procesu. Po zakończeniu całego procesu wiadomo również czy osiągnięto oczekiwane korzyści lub dla czego nie było spodziewanych korzyści. Przechowywanie i łatwy dostęp do informacji o kontaktach z danym klientem są szczególnie ważne, gdy są to kontakty wielokrotnie i trwające wiele lat, gdy w obsługę klienta zaangażowanych jest wielu pracowników firmy, o różnym okresie zatrudnienia i różnym doświadczeniu zawodowym.

Zanim zaczniemy zajmować się szczegółami związanymi z opracowaniem odpowiedniej aplikacji i integracją oprogramowania, trzeba dokładnie przeanalizować procesy realizowane w firmie i usytuować je dokładnie w stosunku do wymagań. Należy analizować procesy, używając ich trójwymiarowego modelu, rozdzielić je na trzy wymiary strukturalne: drogi (marszruty), role i reguły, zastanowić się jak odwzorować strukturalne procesy pracy na platformę sieci komputerowej czy też w końcu ocenić potrzeby co do liczby pracowników niezbędnych do opracowania i administrowania systemu do obsługi procesów pracy.

## 1. Zdefiniowanie strukturalnych wymiarów procesów biznesowych

Najważniejszą pracą koncepcyjną zespołu zajmującego się restrukturyzacją i wspomaganie procesów pracy, po wykonaniu analizy, powinna być diagnoza, co źle działa w badanym procesie. Użytkownik każdej aplikacji, umożliwiającej obsługę procesów pracy, powinien zastanowić się dlaczego niektóre osoby znajdują się na liście odbiorców dokumentów związanych z obsługą procesu, jaka

powinna być kolejność obiegu dokumentu i uprawnienia uczestników procesu. Oznacza to skoncentrowanie się na istniejących i idealnych strukturach – drogach, rolach i regułach – procesów biznesowych przedsiębiorstwa.

Wskazane jest w pierwszej kolejności zajmować się procesami, które są względnie dobrze zdefiniowane i dla których przewidywane zyski ich przeprojektowania przeważają nad kosztami związanymi z przerwami w pracy. Należy rozważyć wszystkie procesy za pomocą takich istotnych kryteriów jak: szybkość, koszt, dokładność, jakość, zadowolenie klienta i elastyczność.

#### *Analiza czasowa projektu.*

Czas realizacji procesu jest jednym z najistotniejszych czynników w procesach biznesowych. Z tego powodu należy najpierw przeprowadzić czasową analizę procesu. Zdefiniować dla własnych potrzeb oczekiwania końcowego klienta, przebieg trwania procesu, dostawy czy usługi. Należy również podzielić procesy na subprocesy, czynności, zależności i uczestników. Ponadto zebrać dane o trwaniu każdej wydzielonej czynności, ilości czasu, którą potrzebuje dokument na przechodzeniu między stanowiskami pracy. Warto wszystkie te dane opracować w postaci diagramu PERT, wykresu Gantta albo struktury podziału pracy.

Wykres PERT ma postać diagramu przepływów, podkreślającego zależności logiczne między różnymi czynnościami. Istotne jest określenie ścieżki krytycznej sieci, co pozwala na wyznaczenie czynności, które stanowią o czasie realizacji procesu. Jest to szczególnie ważne przy analizie czasowej procesów pracy, gdyż pokazuje sekwencje czynności krytycznych, wyznaczających minimalny czas wykonania procesu. Skracając ścieżkę krytyczną np. przez eliminację kilku niepotrzebnych etapów oceny, spowodujemy skrócenie czasu trwania procesu.

Struktura podziału pracy to macierz zawierająca hierarchiczny wykaz wszystkich subprocesów i czynności. Powinna ona zawierać czas trwania czynności, określenie momentu rozpoczęcia (zakończenia) oraz wymieniać osoby odpowiedzialne za te czynności i subprocesy. Struktura podziału pracy to zarys logiczny pozwalający na ocenę czy pewne procesy nie zawierają nadmiarowych lub zbędnych czynności, które można usunąć lub skonsolidować, skracając czas procesu. Pozwala również na logiczne pogrupowanie czynności, przydzielenie zasobów oraz ogólną ocenę kosztów. Łącząc kategorie pracowników z określonymi czynnościami, można definiować role i uprawnienia, które będą wymuszone za pomocą automatycznych narzędzi do obsługi procesów pracy.

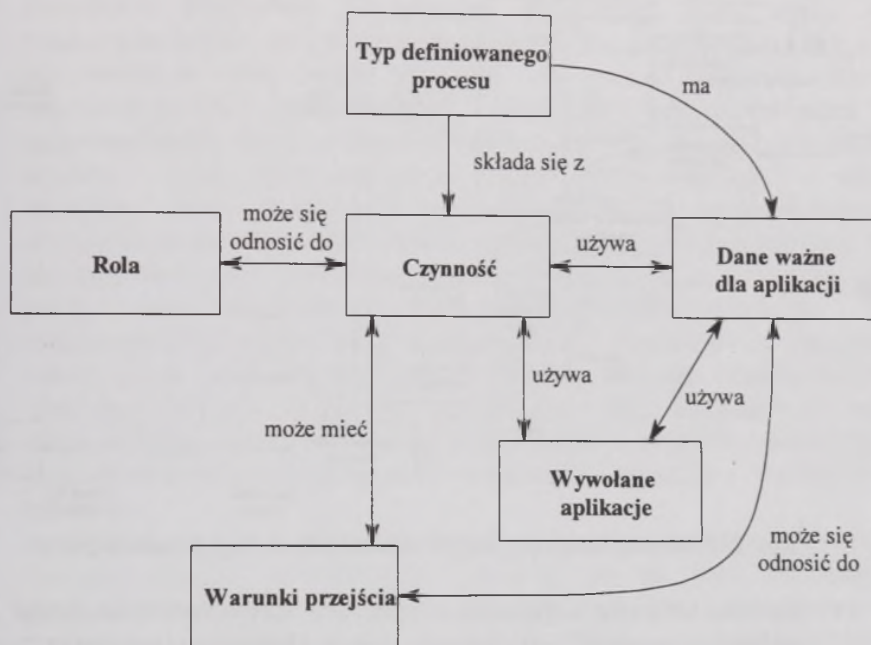
Procesy biznesowe mają charakter wielowymiarowy i trudno je ująć za pomocą jednego diagramu, wykresu lub macierzy. Warto więc w trakcie analizy używać wszystkich tych środków. Końcowym wynikiem analizy powinna być szczegółowa mapa aktualnych procesów biznesowych które należy obsługiwać za pomocą aplikacji oraz nowych procesów, które mają być docelowo obsługiwane.

#### *Zdefiniowanie elementów składowych procesu.*

Projektowanie procesów wymaga modelowania i symulowania wielu alternatywnych przebiegów procesów pracy i ich oceny na podstawie kryteriów

zawartych w uzasadnieniu projektu systemu do obsługi procesów pracy. Zmienne które należy wziąć pod uwagę to:

- czynności
- role
- warunki przejścia
- dane istotne dla procesu
- reguły



Rys. 1. Metamodel podstawowego procesu

Czynności – to jednostki pracy, które mają zdefiniowane warunki rozpoczęcia i zakończenia. Po zakończeniu czynności produkt końcowy jest przesyłany innym uczestnikom procesu. Czynności można podzielić na poszczególne etapy, wykonywane przez konkretnych pracowników, w zgodzie z ustalonymi procedurami lub standardami obowiązującymi w firmie.

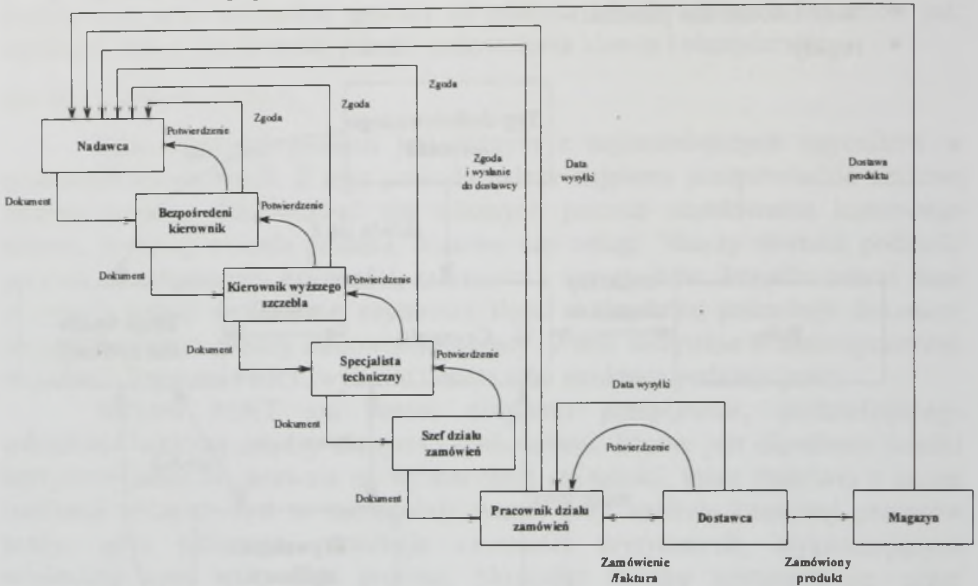
Role – opisują czynności, które mogą wykonywać uczestnicy procesu i precyzują przysługujące im uprawnienia

Warunki przejścia – określają kontekst lub zbiór zdarzeń realizowanych w procesie biznesowym. Możemy podzielić je na kilka kategorii:

- inicjowanie – to inaczej warunki wymagane do rozpoczęcia procesu
- ograniczenia czasowe – warunki opisujące maksymalny i minimalny czas wykonania czynności, łącznie z czasem oczekiwania w kolejce wejściowej, przetwarzaniem i oczekiwaniem w kolejce wyjściowej
- warunki wstępne czynności – opisujące kiedy konkretny uczestnik procesu może rozpocząć określoną czynność



- warunki wykonywania – podające rodzaj narzędzi, aplikacji, metodykę i techniki potrzebne do przetworzenia obiektu pracy
- warunki powiadamiania – określające automatyczne powiadamianie różnych uczestników procesu o zdarzeniach, takich jak akceptacja dokumentu, jego odrzucenie lub powrót do nadawcy



Rys. 2. Powiadamiania w typowym systemie do obsługi procesów pracy

- warunki końcowe czynności – opisujące kiedy konkretna osoba może zakończyć czynność i co się wtedy dzieje. Marszruta obiektów pracy może być opisana jako połączenie warunków: szeregowego, równoległego lub warunkowego przesyłania obiektu pracy
- warunki bezpieczeństwa – określające które osoby mogą uczestniczyć w procesie, jakie funkcje mogą realizować i do jakich informacji mają dostęp
- warunki audytu – podające które zdarzenia, w jakim stopniu szczegółowości, jak często i gdzie należy zapisywać dla potrzeb audytu
- warunki zakończenia – określające kiedy czynność może być zakończona lub przerwana i co zrobić z informacjami które powstały w procesie

Dane istotne dla procesu – to informacje używane w procesie w celu wyliczenia warunków – na przykład w celu inicjowania, audytu lub zakończenia procesu – i określenia sposobu przesyłania, przetwarzania obiektu pracy. Prototypem wszystkich danych istotnych dla procesu jest standardowy formularz firmowy

Reguły – to stwierdzenia warunkowe, opisujące jak dane istotne dla procesu będą przetwarzane, przesyłane i kontrolowane

*Określenie sposobów usprawniania procesu.*

Po określeniu aktualnego procesu biznesowego należy zastanowić się nad sposobami usprawnienia go. Jest to skomplikowana procedura, wymagająca operowania na wielu zmiennych oraz uwzględniania wielu czynników ludzkich, technicznych i zarządczych. Sposoby usprawnienia procesu pracy to między innymi:

- maksymalizacja danych istotnych dla procesu zebranych u jego źródła – proces istnieje po to, aby zebrać dane i przekształcić je w wyniki biznesowe. Można go przyspieszyć, dostarczając osobie która go rozpoczyna zestaw narzędzi umożliwiających jego lokalizację, pobranie i wprowadzenie dużej ilości informacji, co pozwoli następnej osobie zajmować się tylko oceną, akceptacją i przesyłaniem dalej obiektu pracy
- maksymalizacja liczby czynności zakończonych w miejscu powstania procesu – osobę która rozpoczyna proces należy wyposażyć w takie narzędzia, które pozwolą na realizację czynności, które wcześniej wymagały pracy wielu osób. Złożone transakcje nie muszą opierać się na skomplikowanych procesach pracy, jeśli większą jej część może wykonać jeden wysoko kwalifikowany i odpowiednio wyposażony pracownik
- maksymalizacja liczby usług wykonywanych równolegle – znacznie więcej procesów można wykonać równolegle, skracając długość ścieżki krytycznej. Procesy, które były sekwencyjne, gdyż wymagały dostępu wielu osób do jednego dokumentu można radykalnie skrócić, zapewniając wszystkim jednoczesny dostęp do dokumentów (jedynie z możliwością odczytu)
- minimalizacja ilości dokumentów na papierze – procesy biznesowe powinny działać na informacji cyfrowej, nie na papierze. Nośnik papierowy to strata czasu, miejsca i pieniędzy. Jest to drogie medium przy zakupie, transportowaniu, przetwarzaniu, przechowywaniu, archiwizowaniu, wyszukiwaniu i utylizacji. Eliminacja lub ograniczenie dokumentacji na papierze obniża koszty, przyspiesza procesy i lepiej wykorzystuje pracę ludzką oraz zasoby firmy
- ograniczenie do minimum liczby czynności lub uczestników procesu – wiele procesów zawiera czynności zbędne lub jedynie marginalnie użyteczne, które można logicznie połączyć z innymi czynnościami
- zredukowanie liczby formatek w procesie – jednym z symptomów nieprawidłowego procesu jest to, że wymaga on kilku lub kilkunastu formularzy, podczas gdy wystarczyłby jeden zintegrowany formularz. Z powodu konieczności wypełniania wielu formularzy dane w procesie mogą okazać się niekompletne, nieaktualne lub sprzeczne. Zwykle lepiej jest używać jednego wieloczęściowego formularza, zawierającego kompletne informacje o procesie
- ograniczenie czasu oczekiwania na rozpoczęcie procesu – wadą procesów jest fakt, że osoby oceniające wynik pracy mogły to robić dowolnie długo. Można temu zapobiec wymagając aby pewne działania (rozpoczęcie pracy, zakończenie lub przesłanie innej osobie) zostały podjęte w określonym czasie lub pozostały niezrealizowane. Inna metoda polega na umożliwieniu

nadawcy pracy eskalacji jej ważności i przesłania do komórki nadrzędnej w przypadku opóźnienia

- zredukowanie rozmiaru kolejki wejściowej uczestnika – wiele procesów opóźnia się, ponieważ wiele osób ma zbyt dużo obiektów pracy w swojej skrytce wejściowej, a liczba godzin pracy w ciągu dnia jest ograniczona. Można temu zapobiec przenosząc obiekty pracy między pracownikami przeciążonymi i niedociążonymi. Można też dopuścić do przekazywania obiektów pracy między pracownikami równorzędnymi. Innym rozwiązaniem jest pobieranie obiektów pracy ze wspólnej skrytki wejściowej i obsługiwanie ich w kolejności pojawiania się w niej
- skrócenie czasu uruchamiania aplikacji użytkownika po pojawieniu się obiektu pracy – odpowiednia aplikacja powinna uruchamiać się automatycznie z chwilą pobrania obiektu pracy. Powinna ona móc czytać, redagować i operować na wszystkich dokumentach i danych bezpośrednio. Aplikacja powinna zapisywać dokumenty w formacie stanowiącym standard korporacyjny, dzięki czemu cała praca jest wykonywana w ustalonym standardzie, minimalizując konieczność zwrotów do poprawki. Zintegrowany zestaw standardowych narzędzi zwiększa wydajność pracowników i redukuje czas spędzony na nieefektywnych próbach uruchamiania i używania nietypowych aplikacji
- ograniczenie czasu potrzebnego uczestnikom procesu na dostęp do dokumentacji i baz danych – wiele czasu przeznaczonego na przetwarzanie typowego obiektu pracy poświęca się na poszukiwanie właściwej informacji, niezbędnej do jego oceny. Czasami cała potrzebna informacja jest włączona do formatki, ale często ma ona postać rozproszonych dokumentów, plików, baz danych. Uczestnicy procesu pracy muszą mieć dostęp on-line nie tylko do dokumentacji korporacyjnej, ale również do motoru przeszukiwania umożliwiającego znalezienie każdej dokumentacji
- minimalizacja czasu wykonania każdej czynności – podstawowa metoda skrócenia czasu wykonania czynności polega na wymuszeniu większej wydajności pracowników. Można to osiągnąć przez zdefiniowanie procedury powodujące rozsyłanie notatek przypominających o zbliżającym się terminie pracy oraz upominających o przekroczeniu terminu. Notatki będą przesyłane do bezpośredniego przełożonego, który musi podjąć działania zapewniające wykonanie pracy w określonym czasie lub przesłanie jej innej osobie
- minimalizacja czasu potrzebnego na przesłanie obiektu pracy między czynnościami – długi czas przekazywania obiektów pracy to typowa cecha pracy na dokumentach papierowych. Technologie informatyczne pozwalają na szybkie przesyłanie plików. Można pewne procesy ustawić tak, by przesyłanie następowało z pewnym, z góry zadany opóźnieniem. Do tej chwili obiekt znajduje się w skrytce wyjściowej poprzedniego uczestnika procesu, czekając na chwilę, gdy następna osoba będzie dostępna do przyjęcia obiektu pracy. Minimalizacja czasu na przesyłanie obiektów wydłuża czas na ich przetwarzanie



- zredukowanie czasu potrzebnego administratorowi i uczestnikom procesu na uzyskanie informacji o stanie obiektu pracy – dużym problemem procesów wykorzystujących dokumenty papierowe jest trudność w kontroli ich stanu na bieżąco. Wiele narzędzi do obsługi procesów pracy pozwala na monitorowanie obiegu dokumentów i ścisłą kontrolę. Systemy do obsługi procesów pracy można skonfigurować tak, aby przesyłały notatki o zdarzeniach w procesach pracy właściwym uczestnikom procesu
- upewnienie się, czy standardowe marszruty, role i reguły są stosowane automatycznie – zmienna dynamika procesów pracy powoduje szkody w standardowych procedurach pracy. Wiele firm działa w taki sposób, że każda rzecz wykonywana jest inaczej z tygodnia na tydzień lub nawet z dnia na dzień. Powoduje to, że najważniejsze procesy firmowe są wykonywane na zasadzie improwizacji i są wciąż zmieniane. Powoduje to że, każdy zaprojektowany proces może być ignorowany, używany inaczej lub zapomniany już w chwili startu. Aplikacje do obsługi procesów pracy mają tę zaletę, że zmuszają użytkowników do postępowania zgodnie z wytyczonymi procedurami. Minimalizacja czasu spędzonego na zapoznawaniu się z procesami wydłuża czas na uczestniczenie w nich
- zapewnienie możliwości modyfikacji procesów pracy – systemy do obsługi procesów pracy dostarczają solidnych narzędzi zmuszających do posługiwania się sztywnymi procedurami. Jest to jednak podejście skuteczne jedynie w przypadku strukturalnych, często powtarzanych czynności. Jednakże wiele procesów ma charakter bardziej prowizoryczny, zmienia się na skutek zmieniających się warunków lub okoliczności. Systemy do obsługi procesów pracy coraz częściej dopuszczają zmianę definicji procesu i zmianę marszruty obiektu bez konieczności całkowitego zaniechania standardowych procedur
- zapewnienie stałego udziału wszystkich uczestników procesu – efektywność procesów biznesowych zależy od aktywnego udziału wszystkich jego uczestników.

## 2. Wybór właściwej technologii

Po dokonaniu analizy strukturalnej procesów należy spróbować znaleźć aplikację do obsługi procesów pracy, która najlepiej obsłuży nowo zaprojektowane i docelowe procesy. Współczesne rozwiązania do obsługi procesów pracy różnią się znacznie w zakresie struktury, platformy i mediów. Należy zdecydować jak najlepiej obsługiwać planowane procesy: przy użyciu systemu produkcyjnego, opartego na wymianie komunikatów, technikach Web czy zestawach aplikacyjnych.

*Procesy najlepiej obsługiwane przez systemy produkcyjne.*

Najlepsza cecha systemów produkcyjnych to wsparcie klasycznych procesów biurokratycznych – złożonych, niezmiennych, z dużą ilością

dokumentów obsługiwanych przez licznych uczestników procesu, o różnych rolach i zakresach odpowiedzialności. Narzędzia te są czasami określane jako narzędzia do zarządzania przypadkami, ponieważ są stosowane przez agencje rządowe, firmy ubezpieczeniowe i inne instrukcje biurokratyczne, w których automatyzuje się rutynową obsługę klientów i ich przypadków.

Tradycyjne zastosowanie produkcyjnego systemu do obsługi procesów pracy w organizacji to ograniczenie dokumentacji papierowej. W zwykłym procesie z dużą liczbą dokumentów papierowych usuwa się całą taką dokumentację, zachowując proces w praktycznie niezmiennym stanie. Obecnie proces wykonuje się poczynając od skanowania dokumentów papierowych, zapamiętania ich na dysku na początku procesu oraz przetwarzania dalej tylko ich cyfrowych obrazów. Można to także wykonywać, usuwając całkowicie papier, a przysyłać jedynie pliki, formatki elektroniczne i inne obiekty pracy w całym procesie.

Postać papierowa

Postać elektroniczna

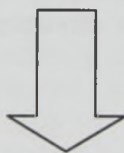
Dokument wejściowy

Skanowanie, digitalizacja



Zarządzanie dokumentem

Marszruta dokumentu i reguły obróbki



Dokument wyjściowy

Archiwizowanie

Rys. 3 Struktura obsługi dokumentu

*Procesy najlepiej obsługiwane przez systemy wykorzystujące wymianę komunikatów.*

Systemy do obsługi procesów pracy wykorzystujące wymianę komunikatów są często prezentowane jako aplikacje najmniej wydajne, rozwiązania przeznaczone do sprawdzania pomysłu w firmach nie zaznajomionych z technologią. Te produkty, często zwane administracyjnymi systemami do obsługi

procesów pracy, służą zwykle do automatyzacji procesów, które nie są zbyt złożone ani nie wymagają przetwarzania skanowanych obrazów dokumentów papierowych. Typowy proces w tych aplikacjach jest definiowany na bieżąco.

Systemy do obsługi procesów pracy wykorzystujące wymianę komunikatów działają na lokalnych i rozległych systemach poczty elektronicznej (w przeciwieństwie do systemów produkcyjnych działających na wspólnych składnicach plików). Narzędzia pozwalają na tworzenie prostych modeli procesów. Możliwości przesyłania są zwykle ograniczone do dróg sekwencyjnych lub jednoczesnych, z pewnym wsparciem kierowania warunkowego, zależnie od wartości pól przesyłanej formatki.

Typowy proces w aplikacjach wykorzystujących wymianę komunikatów jest definiowany na bieżąco. Stosunkowa łatwość używania i niski koszt narzędzi do obsługi takich procesów pracy zachęca użytkowników do eksperymentowania z definiowaniem procesów u ich źródła.

*Procesy najlepiej obsługiwane przez systemy oparte na technikach Web.*

Aplikacje oparte na technikach Web najlepiej nadają się do obsługi takich procesów, w których uczestniczą osoby z wielu lokalizacji firmy. Aplikacje takie działają w Internecie lub korporacyjnym intranecie i są obsługiwane z lokalizacji Web użytkowników, działającej jako bardzo specjalizowana składnica plików. Graficzne narzędzia do modelowania procesów do aplikacji działających w sieciach Web pozwalają na modelowanie zarówno prostych, jak i złożonych procesów w sposób zbliżony do modelowania procesów w systemach produkcyjnych. Aplikacje do obsługi procesów oparte na technikach Web przesyłają głównie samodzielne formatki elektroniczna HTML, ale brakuje im wsparcia folderów złożonych z wielu dokumentów (jak w systemach produkcyjnych) i załączonych plików (jak w systemach wykorzystujących wymianę komunikatów)

Typowa aplikacja oparta na technikach Web musi być zaplanowana we wszystkich szczegółach i na ogół nie powstaje jako wynik spontanicznych działań użytkowników. Zwykle automatyzuje ona proces używany i dostępny dla użytkowników całego przedsiębiorstwa, w tym również użytkowników z odległych miejsc. Środowiska oparte na technikach Web to idealne rozwiązanie współpracy dla firm korzystających z elektronicznej wymiany dokumentów.

*Procesy najlepiej obsługiwane przez systemy oparte na zestawach aplikacyjnych.*

Aplikacje do obsługi procesów pracy oparte na zestawach aplikacyjnych pozwalają na obsługę najbardziej spontanicznych procesów. Są one odpowiednie do obsługi każdego procesu, inicjowanego przez użytkownika czasem tylko jeden raz. Aplikacje tego typu to przyszłość przetwarzania rozproszonego, którego możliwości obsługi procesów pracy będą wbudowane w kliencie poczty elektronicznej, edytorze tekstu, arkuszu kalkulacyjnym i innych programach biurowych lub obsłudze pracy grupowej. Funkcjonalnie podobne do aplikacji wykorzystujących wymianę komunikatów, aplikacje do obsługi procesów pracy



oparte na zestawach aplikacyjnych dają masowemu odbiorcy dogodne możliwości przesyłania dokumentów.

### 3. Określenie wymagań sieciowych i informacyjnych systemu

Po zdefiniowaniu wymagań funkcjonalnych należy określić również wymagania systemowe, związane z implementacją nowych procesów w sieciach korporacyjnych i ich zasobach komputerowych. Określając wymagania co do platformy dla systemu należy wziąć pod uwagę następujące czynniki:

- zasięg geograficzny – w sytuacji gdy większość uczestników procesów pracy znajduje się w jednym miejscu, należy zastanowić się nad wykorzystaniem istniejącego systemu poczty elektronicznej lub systemu do obsługi pracy grupowej bądź używanych zestawów aplikacyjnych jako platformy systemu do obsługi procesów pracy. W miarę jak użytkownicy znajdują się na coraz rozleglejszym obszarze geograficznym, należy rozważyć możliwość umieszczenia aplikacji do obsługi procesów pracy na korporacyjnym intranecie, pozwalającym na stosowanie rozproszonej składnicy plików
- mobilność – jeśli uczestnicy procesów pracy podróżują często należy przewidzieć możliwość łączenia się przez komutowane sieci telekomunikacyjne, sieć telefonów komórkowych lub Internet(intranet). W takim wypadku najlepsze będą rozwiązania wykorzystujące wymianę komunikatów lub zestawach aplikacyjnych. Rozwiązania oparte na technikach Web będą wygodne tylko pod warunkiem zapewnienia połączeń o dużej przepustowości pozwalających na przesyłanie dokumentów HTML z załączonymi elementami graficznymi
- terminale dostępu – wszystkie typy aplikacji do obsługi procesów pracy działają w sieci komputerów PC. Jednakże procesy w firmie mogą wymagać komunikacji z użytkownikami wyposażonymi w pocztę głosową, fax, telefon komórkowy lub pager. W tym przypadku należy brać pod uwagę rozwiązania posiadające bramy do tych systemów
- środowisko operacyjne – należy upewnić się czy aplikacje do obsługi procesów pracy działają na infrastrukturze aplikacyjnej zamierzonych procesów w sieci lokalnej i rozległej. Wybór systemów wykorzystujących wymianę komunikatów i zestawach aplikacyjnych trzeba ograniczyć do istniejących systemów pocztowych i używanych zestawów. Zależnie od stosowanych przeglądarek Web i serwerów stosowanych w firmie może okazać się, że trudno jest osiągnąć pełną współpracę z pewnymi produktami, opartymi na technikach Web
- sieć komunikacyjna – należy upewnić się czy proponowane rozwiązanie do obsługi procesów pracy nie zajmie zbyt dużego pasma sieci. Z punktu widzenia aplikacji do obsługi procesów pracy szybkość sieci nie jest bardzo istotna, ponieważ przesyłanie obiektów pracy nie wymaga dostępu do sieci w czasie rzeczywistym, ale aplikacje te wymagają dużego pasma,

a to może spowodować spadek wydajności innych aplikacji działających w sieci

#### 4. Rozwój i administrowanie aplikacją

Po określeniu wymagań co do platformy i struktury aplikacji do obsługi procesów pracy należy wyznaczyć osoby odpowiedzialne za opracowanie i administrowanie aplikacją. Trzeba przy tym mieć na uwadze kto i jak będzie modyfikował, utrzymywał i wspierał aplikację w jej środowisku pracy. Integralną częścią każdej aplikacji jest wiele czynności związanych z jej ciągłym utrzymywaniem, kontrolowaniem i rozszerzaniem.

Analiza powinna pozwolić na zidentyfikowanie poszczególnych użytkowników, kierowników, analityków i osób personelu technicznego, którzy będą utrzymywali działanie aplikacji. Im bardziej skomplikowana aplikacja, tym trudniej będzie skoordynować działania wszystkich zainteresowanych osób.

Wyboru administratora systemu należy dokonać w zależności od realizowanej technologii, zarówno w aspekcie z góry-na-dół – od kierownictwa do pracowników, jak i między różnymi grupami pracowniczymi. Jeżeli dokonamy podejścia z góry-na-dół, trzeba będzie wyznaczyć administratora w każdej jednostce. Będzie on odpowiedzialny za zdefiniowanie standardowych procesów w tej jednostce i przesyłania ich do głównego administratora systemu oraz ich włączanie w model standardowych procesów w przedsiębiorstwie. Przy podejściu poziomym każdy zespół zdefiniuje własny model procesów, a następnie wynegocjuje połączenie go z pracą innych zespołów.

Niezależnie jednak od przyjętego podejścia w zakresie ustalania obowiązków administratora, musi on stosować się do kilku zaleceń:

- oceniać i zmieniać procesy tylko w regularnych, dobrze zdefiniowanych okresach, aby nie dopuścić do chaosu operacyjnego
- stosować jedno narzędzie do projektowania systemów, co umożliwi ujednolicenie projektów
- uzgadniać wszystkie zmiany w procesach pracy z administratorami tych systemów, na które mają one wpływ, łącznie z wpływem na środowisko operacyjne, sieci, bazy danych, usługi katalogowe i pocztę elektroniczną
- szkolić uczestników procesów i uaktualniać ich wiedzę w regularnych odstępach czasu w zakresie aplikacji do obsługi procesów pracy, jej właściwości i procedur operacyjnych

#### Podsumowanie

Przedstawiony powyżej zbiór zaleceń przy definiowaniu właściwości funkcjonalnych systemu do obsługi procesów pracy, wymagania co do zespołu pracowniczego pozwoli na odpowiednie zaprojektowanie aplikacji oraz modeli procesów. Ponadto pozwoli on na określenie jaki typ aplikacji – produkcyjna,

wykorzystująca wymianę komunikatów, technikach Web czy zestawach aplikacyjnych – jest najlepsza w zależności od wymagań.

Procesowa organizacja pracy z wykorzystaniem technologii informatycznej jest rodzajem „taśmy produkcyjnej” dla działań administracyjnych, biurowych. Wymaga szczegółowego zdefiniowania, sprawnej organizacji i dużej konsekwencji i dyscypliny realizacyjnej. Stosowanie jej pozwala nie tylko efektywniej prowadzić działalność biznesową firmy ale dodatkowo wymusza większą wydajność poszczególnych pracowników. Tak zorganizowana firma jest mobilna na rynku i może szybciej i efektywniej reagować na zmiany strukturalne.

## Literatura

1. Glossary The Workflow Management Coalition Specification, November 1996
2. Zdzisław Szyjewski, Automatyzacja procesów biznesowych – terminologia i klasyfikacje, "Informatyka" nr 1, 1999
3. Witold Staniszkis, Sterowanie przepływem pracy w rozproszonych systemach informatycznych, "Informatyka" nr 2, 1997
4. Edward Kolbusz, Antoni Nowakowski, Informatyka w zarządzaniu. Metody i systemy. Wydawnictwo Zachodniopomorskiej Szkoły Biznesu, 1999
5. Wojciech Kulik, Architektura systemów do obsługi procesów pracy, Materiały konferencyjne „Informatyk zakładowy”, Kazimierz Dolny, maj 2000
6. Zdzisław Szyjewski, Wojciech Kulik: Systemy do obsługi procesów pracy. Efektywność zastosowań systemów informatycznych, Tom 2, WNT Warszawa-Szczyrk 2001

Wojciech Kulik; Politechnika Lubelska  
Zdzisław Szyjewski; Uniwersytet Szczeciński



# NIEPOWODZENIA W PROJEKTOWANIU SYSTEMÓW INFORMATYCZNYCH - CZYNNIKI WPLYWAJĄCE NA SUKCES LUB NIEPOWODZENIE

Tomasz LIS, Stefan NOWAK

**Streszczenie:** Praktycznie każda dziedzina życia codziennego związana jest w mniejszym lub większym stopniu z techniką komputerową. Codziennie w każdej chwili tysiące ludzi z całego świata pracują przy tworzeniu specjalistycznego oprogramowania. Pomimo faktu, iż systemy komputerowe projektowane i wykonywane są przez wysokiej jakości specjalistów wiele z nich kończy się niepowodzeniem. Jest wiele przyczyn i czynników decydujących o szansie osiągnięcia sukcesu w procesie projektowania systemów komputerowych. Oprócz spraw technicznych ważne są również różne, czasami przypadkowe sprawy ludzkie. Celem artykułu jest zaprezentowanie oraz analiza przyczyn nieudanych projektów informatycznych, a także próba rozważenia możliwości zapobiegania podejmowania pracy nad projektem, który w efekcie będzie nieudany.

## Wstęp

Od czasu kiedy zbudowano pierwszy komputer zaistniała potrzeba tworzenia specjalistycznego oprogramowania, które umożliwiło będzie wykorzystywanie w jak największym stopniu jego możliwości techniczne. Każdy nawet najszybszy komputer bez systemu informatycznego jest tylko bezużytecznym narzędziem w rękach jego użytkownika. Wraz z rozwojem sprzętu komputerowego rozbudowywane jest również jego oprogramowanie.

Oprogramowanie komputerowe możemy podzielić na szereg grup różniących się między sobą przeznaczeniem, możliwościami czy zakresem funkcjonalności. Oprócz charakteru i rodzaju pracy systemy informatyczne dzielimy ze względu na ich wielkość i skomplikowanie, tak więc możemy rozróżnić systemy komputerowe:

- małe (przeznaczone przede wszystkim dla małych przedsiębiorstw: sklepów, hurtowni, rodzinnych zakładów usługowych);
- średnie (stosowane w firmach o bardziej skomplikowanej strukturze organizacyjnej);
- duże (występujące w firmach zatrudniających dość znaczną liczbę pracowników – rzędu 1000 lub więcej pracowników).

## Dlaczego interesujemy się przyczynami powstawania nieudanych projektów informatycznych

Badanie przyczyn powstawania nieudanych projektów informatycznych jest warunkiem koniecznym dla zmniejszenia ich ilości. Wśród powodów, dla

których podejmowane są prace badawcze nad nieudanymi projektami informatycznymi wymienić możemy:[1]

- powszechność zjawiska. „Badania prowadzone przez Amerykańską Rządową Agencję Rachunkowości w 1979 r. Pokazują, że mniej niż 3% oprogramowania zakupionego przez rząd Stanów Zjednoczonych było rzeczywiście używane w takim stanie, w jakim zostało dostarczone.”[1] W różnych źródłach literaturowych możemy znaleźć informacje, że w zasadzie ponad 70% rozpoczętych prac nad systemami informatycznymi kończy się niepowodzeniem.
- Wykorzystanie wiedzy praktycznej zdobytej przy nieudanych projektach. Analiza przyczyn powstawania nieudanych projektów informatycznych pozwala na ich unikanie przy podejmowaniu prac nad nowymi systemami.
- Ocena i ryzyko. Niepowodzenie systemu informatycznego może być traktowane jako ważna część obszaru związanego z oceną systemów informatycznych oraz oceną i zarządzaniem ryzykiem. Niepowodzenie jest nieodłącznie związane z ryzykiem, które określić możemy jako negatywny wynik, mający znane lub oszacowane prawdopodobieństwo zaistnienia, oparte na doświadczeniu lub teorii.
- Profesjonalizm. Niepowodzenia w tworzeniu systemów informatycznych są istotnym zagadnieniem w badaniach z powodu ich decydującego wpływu na profesjonalizm w inżynierii systemowej. Powszechnym jest zjawisko ukrywania błędów, niż ich uwidaczniania i oceniania. Większość firm zajmujących się tworzeniem oprogramowania nie gromadzi informacji o niepowodzeniach, a ponadto nie czyni żadnych kroków zmierzających do zrozumienia tego zjawiska.

Możemy powiedzieć, iż analiza przyczyn powstawania oraz podejmowania się tworzenia nieudanych systemów informatycznych prowadzi do zmniejszenia ich liczby. Dzieje się tak przede wszystkim dzięki zrozumieniu przyczyn decydujących o tym, że dany projekt skończy się niepowodzeniem. Gromadzenie informacji o nieudanych projektach powinno stać się ważnym elementem w działalności każdej firmy informatycznej zajmującej się tworzeniem specjalistycznego oprogramowania.

### **Czynniki wpływające na czas w jakim projektowany i wykonywany jest system informatyczny**

Czas w którym wykonywany jest dany system informatyczny zależy od szeregu czynników, wśród których wyróżnić możemy[2]:

- poziom skomplikowania projektu;
- wielkość przeznaczonych na projekt nakładów finansowych;
- możliwości wzajemnej współpracy w grupie informatyków uczestniczących w projekcie;
- niewystarczająca liczba osób biorących udział w projekcie;
- nieoczekiwane wydarzenia losowe (choroba, zdarzenia rodzinne, śmierć);

Oczywiste jest, że czas poświęcony na stworzenie danego systemu informatycznego zależy od jego wielkości, systemy małe i średnie są wykonywane szybciej niż systemy duże. Wynika to z faktu, że systemy małe to przeważnie proste programy mające obsługiwać sklep, hurtownię, czy niewielką firmę usługową (zakres ich funkcjonalności jest stosunkowo niewielki, a wielkość kodu źródłowego niezbyt duża). Programy komputerowe przeznaczone do pracy w dużych przedsiębiorstwach składają się z pewnej ilości mniejszych modułów (z których każdy spełnia oddzielną funkcję) współpracujących ze sobą. Program komputerowy zbudowany modułowo nosi nazwę zintegrowanego systemu informatycznego. Długość kodu źródłowego takiego systemu to kilkanaście tysięcy linii (czasem jeszcze więcej).[2]

Czas wykonania projektu powiązany jest w sposób bezpośredni z nakładami finansowymi przeznaczanymi na jego wykonanie. Jeśli zleceniodawca jest w stanie przeznaczyć odpowiednią sumę pieniędzy to może wynająć firmę informatyczną o odpowiedniej renomie, która jest w stanie zapewnić duże prawdopodobieństwo powodzenia projektu. Decydując się na wybór odpowiedniej firmy projektowej należy zwrócić uwagę na szereg czynników decydujących o jej wiarygodności i rzetelności. Przed przystąpieniem do rozmów kontraktowych powinno sprawdzić się:

- czy firma robiła już systemy o podobnej klasie trudności,
- czy tworzone projekty kończą się sukcesem,
- czy klienci korzystający z jej usług są zadowoleni,
- jakie są warunki utrzymania i konserwacji zakupywanego produktu.

## **Powodzenie projektu**

Ważnym czynnikiem decydującym o powodzeniu w projektowaniu systemów informatycznych jest umiejętność współpracy członków zespołu projektowego. Jeśli zespół informatyków mających wzajemnie współpracować jest zespołem zgranym (każdy zna swoje możliwości i wie jakie jest jego miejsce w grupie) wówczas oprogramowanie przez nich stworzone powstanie szybciej niż w przypadku wzajemnej niezgody. Podobnie jak w każdej dziedzinie życia (gdzie efekt końcowy jest wynikiem pracy większej niż jedna liczby osób) do uzyskania pożądanego celu potrzebna jest zgoda i zrozumienie. Jeżeli w grupie informatyków znaleźć możemy chociaż jedną osobę, której z pewnych przyczyn nie zależy na odniesieniu sukcesu (zawiść, zazdrość niezgoda), wówczas szanse na stworzenie udanego (poprawnie działającego) systemu informatycznego są niewielkie, a w zasadzie nikłe. Czasami zdarza się, że człowiek na kierowniczym stanowisku nie mający bezpośredniego wpływu na przebieg projektu koniecznie chce wiedzieć, jak postępują prace wówczas stara się „wcisnąć” do zespołu swojego człowieka (którego zadaniem jest raczej donoszenie niż praca). Członek zespołu, który dołączony jest do niego przed rozpoczęciem prac nie jest zgrany z pozostałą jego częścią, nie zna możliwości pozostałych członków grupy, a przez to nie może określić swojego w niej miejsca.[2]



Przed przystąpieniem do prac projektowych nad nowym systemem informatycznym należy zdecydować jaka liczba osób powinna brać w nim udział. Zadaniem dobrania odpowiedniej ilości programistów (programistów których kwalifikacje i umiejętności pozwolą na udane zakończenie przedsięwzięcia) zajmuje się kierownik projektu. Dobierając ludzi do grupy należy kierować się ich umiejętnościami merytorycznymi i umiejętnością współpracy. Decydującą rolę w powodzeniu ma określenie liczby osób mających ze sobą współpracować, liczba programistów nie może być zbyt duża, ale również nie powinna być zbyt mała.

Niemożliwością jest na przykład ukończenie prac nad dużym systemem w określonym terminie przez grupę, która w tym samym okresie czasu ukończyłaby z powodzeniem system mały (prosty). Zdarza się czasami, że „grupa projektowa” zostaje w trakcie pracy zmniejszona (ponieważ np. firma otrzymuje inne intratne zlecenie i konieczne staje się przerzucenie paru informatyków do nowego zadania), fakt ten nie jest optymistyczny dla prowadzonych prac, a konsekwencją takiego postępowania w najlepszym razie może być przesunięcie terminu zakończenia projektu (inni członkowie grupy muszą przejąć zadania, którymi zajmowali się przesunięci programiści).

## **Czynnik ludzki w procesie projektowania**

Życie ludzkie charakteryzuje się tym, że nie wszystkie wydarzenia w nim zachodzące są możliwe do przewidzenia. Czasami bywa tak, że w zaplanowanym z góry określonym okresie czasu zachodzą nieprzewidziane zdarzenia, które w efekcie zmieniają ustalony uprzednio grafik postępowania. Musimy wówczas zrezygnować z pewnych planów skupiając się raczej na rozwiązaniu nieoczekiwanych sytuacji. Do takich właśnie nieoczekiwanych sytuacji należą:[2]

- choroba czy śmierć członka zespołu, czy osoby z najbliższego otoczenia (rodziny);
- bardzo interesująca np. pod względem finansowym czy ambicjonalnym oferta pracy;
- zaprzestanie działalności przez firmę wykonującą lub zamawiającą projekt;
- ślub czy rozstanie z bliską osobą.

Systemy komputerowe, których poziom skomplikowania jest stosunkowo niewielki mają większe szanse na wykonanie ich w terminie niż systemy złożone. Czas stworzenia projektu jest bezpośrednio powiązany z szansą jego powodzenia

## **Podsumowanie**

Twierdzenie, iż niepowodzenia występujące w procesie projektowania systemów informatycznych są zjawiskiem nieistotnym i występującym stosunkowo rzadko jest dalekim od prawdy. Zjawisko to jest zjawiskiem powszechnym i ze względu na to bardzo ważne są prowadzone nad nim badania. Zrozumienie wszystkich czynników towarzyszących powstawaniu systemów informatycznych

prowadzi do zmniejszenia liczby niepowodzeń, a co za tym idzie zmniejszenia kosztów działalności firmy informatycznej.

Przy tworzeniu oprogramowania należy pamiętać, że oprócz spraw czysto technicznych takich jak sprzęt, czy narzędzia programistyczne ważne są również inne czynniki wpływające na efektywność pracy. Zaliczyć możemy do nich np. czynnik ludzki, czyli wszelkie przewidywalne i nieprzewidywalne wydarzenia mające związek z zatrudnionymi przy danym projekcie ludźmi.

O tym, że dany projekt systemu informatycznego jest nieudany świadczyć może[1]:

- niezadowolenie wśród osób wewnątrz lub z zewnątrz organizacji bezpośrednio zainteresowanej działaniem stworzonego systemu,
- przerwanie prac nad projektem przed jego całkowitym ukończeniem,
- określenie projektu za nieudany po całkowitym jego zakończeniu, a przed dostarczeniem finalnego produktu do odbiorcy,
- wycofanie poparcia dla danego projektu,
- niezadowolenie końcowych użytkowników z dostarczonego im produktu systemu informatycznego – dzieje się tak zazwyczaj w sytuacji niedokładnego rozpoznania konkretnego zapotrzebowania wśród osób mających pracować na nim.

## Literatura

1. Paul Beynon Davies; „Inżynieria systemów informacyjnych”; Wydawnictwa Naukowo-Techniczne; Warszawa 1999 r.
2. Edward Yourdon; „Marsz ku klęsce. Poradnik dla projektanta systemów”; Wydawnictwa Naukowo-Techniczne; Warszawa 2000 r.
3. Kisielnicki J., Sroka H. „Systemy Informacyjne Biznesu”, Placet, Warszawa 1999
4. Celia Burton, Norman Michael „Zarządzanie projektem”, Wydawnictwo Astrum, Wrocław 1996

Tomasz Lis  
Politechnika Częstochowska,  
Wydział Zarządzania  
Instytut Informatyki i Ekonometrii  
42-200 Częstochowa; ul. Armii Krajowej 19b

Stefan Nowak  
Politechnika Częstochowska,  
Wydział Zarządzania  
Instytut Informatyki i Ekonometrii  
42-200 Częstochowa; ul. Armii Krajowej 19b





# O STAROŚWIECKIM PODEJŚCIU DO PROJEKTOWANIA SYSTEMÓW INFORMATYCZNYCH

Jerzy Stanisław NOWAK

Informatycznym problemem nr 1 jest od wielu lat kwestia nieudanych wdrożeń. W celu uniknięcia tego rodzaju nieszczęść poszukuje się od wielu lat środków zaradczych, ale zdaniem autora, poszukiwania te ograniczają się wyłącznie do kwestii formalnych, a nie do rzeczywistych przyczyn tego rodzaju niepowodzeń.

Aby zdać sobie sprawę z powagi sytuacji przypomnijmy kilka danych amerykańskich – wg nich tylko 28,17% dużych przedsięwzięć informatycznych ukończono w terminie, 23,83% opóźniono, a 48% zaniechano. Natomiast w przypadku przedsięwzięć bardzo dużych tylko 13,67% zrealizowano w terminie, 21,33% opóźniono, a 65% zaniechano [1].

Kwestie formalne to próby wskazania, że przyczyną niepowodzenia (nieudrożenia systemu) jest złe kierowanie tzw. projektem, czyli mówiąc po polsku przedsięwzięciem polegającym na wdrożeniu w życie projektowanego systemu informatycznego. Podaje się w licznych artykułach prasowych, a także w literaturze specjalistycznej wskaźniki procentowe nieudanych wdrożeń (z reguły są to liczby w granicach 80%! ). Co dziwniejsze, zdecydowana większość tych wskaźników pochodzi ze Stanów Zjednoczonych AP, ojczyzny „wzorowego kierowania projektem”. To zdziwienie bierze się stąd, że właśnie w tym kraju powstały standardy prowadzenia prac wdrożeniowych (jakże znany PMI – *Project Management Institute*), które z kolei zaowocowały innymi standardami, zarówno amerykańskimi, jak i powstałymi w innych częściach świata (z wyjątkiem Związku Radzieckiego). Pomijając standardy innych krajów, musi zaskakiwać mnogość podobnych norm w samych Stanach Zjednoczonych (np. standard GAO – US General Accounting Office, istnieją również analogiczne standardy w niektórych stanach i miastach (Floryda, Nowy York) czy ministerstwach i agendach rządowych (NASA, Dep. Obrony i inne). Okazuje się więc, że opracowanie regulaminu zbiurokratyzowanego sposobu prowadzenia prac wdrożeniowych nic nie pomaga – dalej się nie udaje! Im więcej zasad, reguł i przepisów, tym większy odsetek nieudanych wdrożeń. Wynika stąd, że sztuka wypełniania protokołów zdawczo-odbiorczych etapów prac (zwana niekiedy kierowaniem projektem) na nic się zdaje i nie tu należy poszukiwać przyczyn niepowodzeń. O tym, że nie są to gołosłowne stwierdzenia, mogą przekonywać realizacje systemów informatycznych w Polsce w latach 70-ych i jeszcze na początku lat 80-ych – nie podaje się przykładów nieudanych wdrożeń, natomiast powszechnie narzekano na „nadmiar” rozwiązań – wytykano (nie bardzo wiadomo komu) np. że opracowano w Polsce ok. 420 systemów gospodarki materiałowej (o dziwo, wszystkie działały). Warto tu podkreślić, że nikt wówczas nie mówił o „kierowaniu projektem”, natomiast wszystkie ośrodki przetwarzania danych przyzwyczajone były do ciężkiej i żmudnej pracy, szczególnie na etapie wdrożenia systemu.

Sprawą podstawową, warunkującą powodzenie każdego planu jest wzorowe przygotowanie podstaw (założeń) do planowanego zadania i nie ma tu znaczenia, jak ono jest duże. To podkreślali szczególnie mocno autorzy opracowań informatycznych z lat 70-ych. Wypracowane wówczas podejście do projektowania systemów nie było szczególnie oryginalne – po prostu praktyka projektowania systemów informatycznych skrupulatnie wykorzystywała rozwiązania stosowane w projektowaniu technicznym – projektowanie systemu informatycznego niczym się nie różniło od projektowania hali, maszyny czy procesu produkcyjnego.

Zamierzeniem autora jest ogólne porównanie tej staroświeckiej koncepcji podejścia do projektowania systemów informatycznych, która, jak się wydaje, gwarantowała więcej sukcesów niż aktualne, błyskotliwe metody kierowania projektem informatycznym. Ze względu na prostotę ujęcia ograniczono się do rzeczy najważniejszej tzn. do budowy założeń systemu informatycznego.

Proces budowy systemu informatycznego przewidywał wykonywanie ściśle określonych faz projektowania, dokładnie dokumentowanych na każdym etapie. Należały do nich:

- opracowanie założeń systemu informatycznego,
- opracowanie projektu technicznego systemu,
- opracowanie projektu oprogramowania,
- opracowanie dokumentacji eksploatacyjnej.

Kluczową sprawą jest opracowanie założeń systemu informatycznego. Zadaniem założeń systemu informatycznego było (jest?) przedstawienie ogólnej koncepcji systemu oraz jego charakterystyki technicznej, ekonomicznej i organizacyjnej. Przygotowanie założeń przebiegało w dwóch fazach:

- wypracowanie koncepcji systemu
- opracowanie charakterystyki technicznej, ekonomicznej i organizacyjnej (dla danej koncepcji systemu).

Wykonane w ten sposób założenia stanowiły:

- podstawowy materiał dla kierownictwa klienta przedstawiający koncepcję rozwiązania informatycznego wraz z **oceną spodziewanych korzyści oraz niezbędnych nakładów**,
- podstawę do opracowania przez użytkownika planu organizacyjno-technicznego przygotowania instytucji klienta do wdrożenia systemu informatycznego,
- podstawę dla zespołu projektantów do opracowania projektów technicznych (**warunek – zatwierdzenie założeń systemu przez użytkownika**).

Taki tryb pracy obowiązywał przy tzw. dużych systemach; w przypadku rozwiązań mniejszych możliwe było podjęcie realizacji projektu technicznego systemu, ale koncepcja systemu i inne uwarunkowania techniczne, organizacyjne i ekonomiczne musiały być przedstawione użytkownikowi przed rozpoczęciem właściwego projektowania technicznego. Dla przypomnienia procedur obowiązujących w tym czasie należy zwrócić uwagę na podstawowy kanon opracowania założeń, a mianowicie metody:

- diagnostyczna i
- prognostyczna.

Metoda diagnostyczna zakładała wykonanie analizy firmy (instytucji) przed opracowaniem założeń i na tej podstawie określenie wymagań do przyszłego systemu. Metoda prognostyczna zakładała przyjęcie koncepcji systemu idealnego (przyszłościowego) i na tej podstawie określano wymagania dla rzeczywistego klienta. Upraszczając nieco problem, można powiedzieć, że przy wyborze metody diagnostycznej wykonywano wszechstronną analizę przedsiębiorstwa i na tej podstawie podejmowano decyzję, czy w firmie potrzebny jest system klasy MRP, CRM czy np. SCM itd. (używając aktualnych określeń systemów). Projektant systemu w metodzie prognostycznej orzekł, że skoro firma jest firmą produkcyjną, to niezbędnie potrzebny jest system klasy MRP – to myślenie jest nadal aktualne. Oszczędzono przy tym czas przewidziany na analizę firmy, co pozornie pozwalało skrócić czas wdrożenia systemu. Warto zauważyć, że zwracano wtedy uwagę na fakt zwiększonego wysiłku intelektualnego, wyobraźni i kwalifikacji informatyków przy stosowaniu metody prognostycznej, co jest oczywiste.

Założenia systemu informatycznego odgrywały więc bardzo ważną rolę przedstawiając wizję systemu oraz warunki techniczno-organizacyjne jego opracowania i wdrożenia.

Zakres założeń projektowych obejmował m.in.:

- wypracowanie koncepcji systemu,
- ogólna charakterystyka techniczna, ekonomiczna i organizacyjna systemu,
- charakterystyka techniczna, ekonomiczna i organizacyjna podsystemów (modułów),
- załączniki (a w tym):
  - dokumentację analizy,
  - założenia teleprzetwarzania,
  - protokoły uzgodnień itp.

Należy tu zwrócić uwagę, że ważnym elementem dokumentu jest określenie wymagań użytkownika do systemu i systemu do użytkownika (warunki organizacyjne funkcjonowania systemu) oraz próba oszacowania efektów ekonomicznych wdrożenia systemu. Cokolwiek by o tym nie myśleć – problem szacowania efektów wdrożenia systemu informatycznego jest nadal tak samo ważny jak 25 lat temu. Załącznik nr 1 zawiera wyciąg z podręcznika autorstwa Henryka Zygiera [2] i określa szczegółową charakterystykę techniczno-organizacyjną i ekonomiczną. Na podstawie treści załącznika staje się jasne, że projektant systemu informatycznego musiał przewidzieć wszystko (czego praktycznie nie robi obecnie decydując się na zakup i wdrożenie tzw. gotowego pakietu MRP, CRM, SCM,.....itd.).

Można więc stwierdzić, że poprawne przygotowanie założeń systemu informatycznego, w których jasno określa się cele i warunki wdrożenia gwarantują efektywne wdrożenie, w trakcie którego będą potrzebne doświadczenia z zakresu kierowania projektem. Nieznane są obecnie przykłady, że wybór i wdrożenie systemu informatycznego dokonuje się wg podobnych zasad. Szczególnie rażące jest pomijanie w aktualnych wyborach systemów wymagań wprowadzenia zmian or-



ganizacyjnych po wdrożeniu systemu, co należy uznać jako podstawowa przyczynę niepowodzeń we wdrażaniu – nowy system jest nakładany na istniejącą strukturę organizacyjną i nie należy wobec tego oczekiwać na powodzenie takiej operacji.

Dla uzasadnienia celowości opracowywania założeń systemu informatycznego podaje się poniżej dwa przykłady (rzeczywiste), pochodzące z połowy lat 80-tych:

1. Zakład X – ok. 12.000 zatrudnionych. Kierownictwo przedsiębiorstwa zleca opracowanie systemu tzw. wewnętrznego rozrachunku gospodarczego WRG (socjalistyczny prekursor *controllingu*). Służby ekonomiczno-księgowe wskazują, że koncepcje systemu powinien opracować dział informatyki (!), który podejmuje się tego zadania, zastrzegając się, że opracowany dokument przedstawia wyłącznie informatyczny aspekt problemu (algorytmy rozliczeń nie były przedmiotem opracowania). Przygotowane założenia systemu wykazują, że jego wdrożenie w ówczesnych warunkach technicznych (nie znano jeszcze kodów kreskowych, skanerów itp.) jest prawie nierealne i wymaga znacznych nakładów finansowych – m.in. konieczne było stworzenie dodatkowego dokumentu, określającego stan zaawansowania procesów technologicznych przy przekazaniu pozycji z jednego wydziału produkcyjnego na drugi oraz ponowne wprowadzanie danych (innych – m.in. nr serii produkcyjnej) z uprzednio obrobionych kart pracy akordu i dniówki zadaniowej. Ze względu na wielkość zakładu – ilość dodatkowych dokumentów szacowano na ok. 100 tys. miesięcznie, a ilość kart roboczych wynosiła ok. 180 – 200 tys. miesięcznie wykazano, że w sytuacji braku inwestycji w unowocześnienie procesu pobierania danych wyniki systemu WRG będą dostępne praktycznie 3 tygodnie po zakończonym miesiącu, co jak się okazało w trakcie zbierania opinii do założeń – nie miało sensu. Pomysł wdrożenia systemu został więc zawieszony bez kosztownego eksperymentu na nieudane wdrożenie.
2. Zakład X – służba planowania produkcji występuje do działu informatyki z wnioskiem o opracowanie systemu rozliczania wydziałów produkcyjnych z pobranych materiałów bezpośrednio produkcyjnych (był to poważny problem w firmie). Znana była ilość pobranych materiałów na podstawie dokumentów (karta limitowa, kwit Rw i inne pomocnicze). Zużycie materiałów miano określać na podstawie dokumentów odbioru gotowych części i protokołów wydania przez wydziały produkcyjne odpadów produkcyjnych (wióry itp.). Dział informatyki opracowuje założenia, w których wykazuje brak możliwości dokładnego porównania ilości materiałów pobranych i faktycznie zużytych (w procesie produkcji następują straty materiałów) – nikt nie potrafił wskazać (ani podjąć takiego ryzyka) algorytmu oszacowania rzeczywistych ubytków materiałów za zgodne z oczekiwaniami i wobec tego postanowiono zrezygnować z dalszych prac.

Powyższe przykłady, aczkolwiek mające już tylko wartość historyczną, wskazują, że opracowanie dokładnych założeń systemu informatycznego gwarantuje taką ocenę sytuacji, w której można podjąć decyzję o realizacji lub rezygnacji

z projektowanego systemu informatycznego. Tylko powrót do takiej praktyki projektowania pozwoli uniknąć wielu rozczarowań i niepowodzeń w trakcie wdrażania systemów informatycznych. Staroświeckie podejście do realizacji systemu informatycznego okazuje się nadal skuteczne i gwarantujące powodzenie.

*[to be continued]*

## Literatura

1. Czarnacka-Chrobot B. – Błędy w zarządzaniu projektem informatycznym – skala problemu i aspekty metodologiczne. [w:] Efektywność zastosowań systemów informatycznych. Red. J.Grabara i J.S. Nowak. WNT, Warszawa-Szczyrk 2001,
2. Zygier H. – Metodyka projektowania systemów informatycznych, OBRI, Warszawa 1974

Mgr inż. Jerzy S. Nowak  
ComputerLand S.A.,  
40-955 Katowice, ul.Bytkowska 1B tel. 0-32-7899601  
[jnowak@computerland.pl](mailto:jnowak@computerland.pl)

Polskie Towarzystwo Informatyczne  
40-012 Katowice, ul. Św. Jana 10 tel. 0-32-2539 678  
[Jerzy.Nowak@pti.org.pl](mailto:Jerzy.Nowak@pti.org.pl)

**WYCIĄG**  
**Z OPRACOWANIA H.ZYGIER – METODYKA PROJEKTOWANIA SYSTEMÓW INFORMATYCZNYCH (STR. 84 – 89)**

- 3.2**      **Część II - Ogólna charakterystyka techniczna, ekonomiczna i organizacyjna systemu**
- 3.2.1**      Przedmiot, zakres i głównie funkcje systemu (na podstawie zadania projektowego).
- 3.2.2**      Krótka charakterystyka warunków w jakich działa system.
- 3.2.2.1**      Podstawowe cechy i parametry obiekty zarządzania (np. dla przedsiębiorstwa przemysłowego: charakter i typ produkcji, charakterystyka wyrobów finalnych, liczebność załogi, podstawowe wskaźniki techniczno-ekonomiczne, ....
- 3.2.2.2**      Wymagania systemu nadrzędnego (charakterystyka wzajemnych zasileń informacyjnych narzuconych projektowanemu systemowi; w ogólnym przypadku zasilenia te będą realizowane przez odpowiednie podsystemy).
- 3.2.2.3**      Zbiór ograniczeń charakteryzujących system.
- 3.2.2.4**      warunki powtarzalności systemu (opis podstawowych cech i parametrów obiektu zarządzania, przy których możliwe jest zastosowanie danego systemu).
- 3.2.3**      Ogólna charakterystyka systemu.
- 3.2.3.1**      Spis podsystemów i ich elementów składowych - na ogół nie mniej jednostek funkcjonalnych.
- 3.2.3.2**      Wstępne określenie i uzasadnienie potrzebnych środków technicznych; proponowane urządzenia, ich typy podstawowe parametry techniczno eksploatacyjne, ilości (na podstawie oszacowanej pracochłonności eksploatacji - patrz pkt. 3.2.3.4.), ceny jednostkowe, nakłady - w rozbiciu na okresy czasowe, zgodnie z przewidywanym wdrożeniem poszczególnych elementów systemu.
- 3.2.3.3**      Wpływ systemu na działalność obiektu (uwypuklenie zasadniczych pozytywnych cech systemu dla działalności kierownictwa różnych szczebli oraz poszczególnych komórek organizacyjnych i służb, określenie w jaki sposób system będzie zaspakajał potrzeby użytkownika).
- 3.2.3.4**      Zestawienie pracochłonności i kosztów (w rozbiciu na poszczególne lata realizacji systemu);
- projektowania technicznego,
  - programowania (wraz z testowaniem i sprawdzeniem programów),



- prac przygotowawczych i użytkownika,
  - wdrożenia,
  - eksploatacji.
- 3.2.3.5** Oszacowanie zbiorczych efektów (wymiernych i niewymiernych) i nakładów.
- 3.2.3.6** Perspektywy rozwoju systemu.
- 3.2.4** Ważniejsze przedsięwzięcia w zakresie przygotowania obiektu zarządzania do wprowadzenia systemu informatycznego, jak:
- przygotowanie kadr,
  - przygotowanie zmian w organizacji i zarządzaniu (ogólne kierunki i zakres),
  - przygotowanie ośrodka obliczeniowego (jeśli system ma być eksploatowany na własnym komputerze) lub ośrodka przygotowania maszynowych nośników danych wejściowych, stacji transmisji danych itp., z określeniem terminu gotowości (szczegółowe wymagania organizacyjne dla poszczególnych podsystemów ujęte są w ramach ich charakterystyki).
- 3.2.5** Założenia teleprzetwarzania.  
(Jeśli w projektowanym systemie informatycznym ma być zastosowane teleprzetwarzanie).  
Założenia teleprzetwarzania powinny zawierać następujące elementy:
- 3.2.5.1** Przekrój informatyczny strumieni informatycznych i ustalenie warunkowań czasowych, priorytetów i kolejek .
- 3.2.5.2** Określenie typu systemu:
- czasu rzeczywistego,
  - zdalne przetwarzanie partiiowe,
  - wielodostęp.
- 3.2.5.3** Określenie funkcji systemu w podziale zagadnieniowym.
- przydzielenie funkcji:
  - zbieranie i gromadzenie danych,
  - konwersacja,
  - rozprowadzanie danych,
  - zdalne przetwarzanie,
  - wyszukiwanie informacji,
  - obliczenia naukowe i inżynierskie.
- 3.2.5.4** Projekt sieci transmisji danych (patrz przykład - rys. 5).
- 3.2.5.5** Przegląd proponowanych urządzeń komunikacyjnych oraz dostępnego oprogramowania z analizą przydatności w projektowanym systemie.

**3.2.5.6** Oszacowanie kosztów instalacji i eksploatacji.

**3.2.5.7** Bezpieczeństwo banków danych.

**3.2.6** Harmonogram dalszych prac:

- projektowych (z programowaniem),
- organizacyjnych,
- wdrożeniowych.

Określić tu należy kolejność opracowania i wdrożenia podsystemów, jednostek funkcjonalnych, w obiekcie i (lub np. w poszczególnych zakładach wielozakładowego przedsiębiorstwa przemysłowego. Wyszczególnić należy ważniejsze przedsięwzięcia we wszystkich przewidzianych etapach realizacji systemu, wymagające terminowego określenia, ujmując przy tym zadania dla wykonawcy i użytkownika z podaniem terminów rozpoczęcia i zakończenia pracochłonności i form zakończenia; gdzie to tylko jest realne należy jednocześnie określić ostateczne terminy koniecznych ustaleń między użytkownikami a wykonawcą dotyczących projektów. Harmonogram musi być aktualizowany w trakcie jego realizacji.

Dla dużych systemów harmonogram zaleca się opracować na ogół przy pomocy siatki zależności (PERT, ...). W harmonogramie można operować terminami względnymi (a nie konkretnymi datami), tj. liczonymi w miesiącach od czasu „G”, stanowiącego np. datę podjęcia dalszych prac nad realizacją systemu informatycznego. Kolejność projektowania i wdrożenia systemu określają następujące czynniki, które w wielu przypadkach muszą być traktowane łącznie: znaczenie, ciężar gatunkowy,

- ważność danego elementu systemu w konkretnych warunkach obiektu zarządzania,
- wzajemnie powiązania między elementami systemu,
- stopień przygotowania obiektu,
- możliwość wykorzystania gotowych opracowań projektowych (typowe projekty, inne gotowe bądź nawet eksploatowane u innego użytkownika),
- możliwość nabycia i wdrożenia w planowanym terminie odpowiednich urządzeń,
- możliwość szybkiego uzyskania efektów (ekonomicznych i organizacyjnych),
- względy natury psychologiczno-dydaktycznej (jeśli są odpowiednie warunki to w pierwszej kolejności warto wdrożyć jakiś element systemu, który może samodzielnie funkcjonować i swoimi wynikami przekonywać kierownictwo i innych pracowników użytkownika o możliwościach komputera).

**3.2.7** Zasady wprowadzania zmian.

# WYMAGANIA STAWIANE SYSTEMOM INFORMATYCZNYM W ŚWIETLE ZNOWELIZOWANEJ USTAWY O RACHUNKOWOŚCI I PRZEPISÓW WYKONAWCZYCH

Piotr WELENC

**Streszczenie:** Nowelizacja Ustawy o rachunkowości, wchodząca w życie dn. 01.01.2002 stawia znaczące wymagania systemom informatycznym i tym, którzy je tworzą lub eksploatują. Ustawodawca poprzez nowelizację wyraża jeszcze większą troskę o poprawność metodyczną prowadzonych ksiąg i wychodzi naprzeciw konieczności włączenia Polski w międzynarodowe struktury standardów gospodarczych i technicznych. Rachunkowość jest podniesiona do rangi podstawowego narzędzia zarządczego organizacją, w zasadniczy sposób wpływa na *image* jednostki i zbliża wartość księgową do wartości faktycznej firmy. Spełnienie wymogów prowadzi w prostej linii do objęcia systemów wspomagających procesy rachunkowości w firmie audytem informatycznym. Poniżej zostaną omówione wymagania stawiane systemom wspierania rachunkowości w świetle znowelizowanej Ustawy o rachunkowości i przepisów wykonawczych.

## Wstęp

Rola i postęp komputeryzacji postawił nowe wyzwania przed systemami wspierającymi rachunkowość przedsiębiorstw. Z biegiem czasu o rachunkowości w przedsiębiorstwach nie mówi się inaczej jak tylko w kontekście zintegrowanych systemów informatycznych. Nowela Ustawy (Dz. U. Z 2000 r. Nr 113, poz. 1186) weszła w życie 1.01.2002 i wprowadziła dość istotne zmiany z punktu widzenia szeroko rozumianej rachunkowości informatycznej.

Jedną z ważnych cech Ustawy jest to, że rekomenduje ona zachowanie pełnej zgodności z MSR-ami, ponieważ uwzględniają one potrzeby informacyjne szerokiego spektrum klientów (szczególnie dla banków, instytucji finansowych i uczestników rynku kapitałowego) Oznacza to, że nieuregulowane kwestie (m.in. dotyczące systemów informatycznych) będzie można rozwiązywać na wzór standardów międzynarodowych. Zapis ten jest bardzo obiecujący pod względem zastosowania standardów międzynarodowych do audytu informatycznego zintegrowanych systemów wspierających procesy rachunkowości w firmach. Strategia Unii Europejskiej w zakresie rachunkowości zakłada od 2005r. obowiązkowe stosowanie MSR przez notowane na giełdzie spółki akcyjne, a docelowo wszystkie spółki publiczne, banki i instytucje finansowe.

I chociaż w zakres MSR-ów nie wchodzi tematyka

- prowadzenia ksiąg,
- zatwierdzania i kontroli dokumentów,
- sprawdzanie arytmetycznej poprawności zapisów,



- dokonywanie uzgodnień sald,
- kontroli wprowadzania zapisów księgowych,
- dokonywanie uzgodnień i konfrontacja dokumentów,
- kontrole danych wprowadzanych do komputera,
- porównywanie sald kont z zestawieniem sald,
- całość wymogów związanych z stosowaniem elektronicznego przetwarzania dokumentów w rachunkowości,

to zagadnienia powyższe ujęte są w Międzynarodowym Standardzie Rewizji Finansowej (np. MSRF 400). Poświęcony on jest sformułowaniu norm i wskazówek zapewniających zrozumienie systemów rachunkowości i kontroli wewnętrznej oraz ryzyka rewizyjnego i jego komponentów: ryzyka wewnętrznego, ryzyka kontroli i ryzyka niewykrzyca

Systemy rachunkowości ewoluowały przez długi czas w kierunku automatyzacji czynności i rejestracji zdarzeń gospodarczych, a traciły znaczenie w aspekcie wspomagania procesów decyzyjnych. Stale niska funkcjonalność modułów sprawozdawczych powodowała, że komputerowe prowadzenie ksiąg rachunkowych traciło aspekt zarządczy. Brak czasowego osadzenia informacji, brak spójności działań (koordynacji), dezintegracja procesów – to wszystko spowodowało, że rachunkowość w przedsiębiorstwie funkcjonowała przede wszystkim jako sprawozdawczy instrument obsługi podmiotów publiczno-prawnych.

Duża swoboda doboru rozwiązań w zakresie systemów wspomagających proces rachunkowości w firmie, a nade wszystko brak kompleksowego podejścia do informatyzacji procesów gospodarczych doprowadziła do tego, że wiele rozwiązań nadal funkcjonuje lokalnie lub samodzielnie (niezintegrowanie) w obrębie organizacji. W procesie decyzyjnym to właśnie systemy księgowe mają zasadnicze znaczenie. Są systemami o wysokiej użyteczności materialnej, miejscem skrzyżowania się wielu procesów gospodarczych, ponadto charakteryzują się korzystną relacją jakość/cena (niskim kosztem do nieporównywalnie wysokich efektów operacyjnych).

Ustawa o rachunkowości stawia znaczące wymagania przed twórcami systemów. Nakłada również obowiązki na osoby kontrolujące, ponieważ opinia wydawana przez biegłego rewidenta dotyczy również prawidłowości prowadzenia ksiąg rachunkowych, niezależnie od tego, czy księgi prowadzone są ręcznie, czy komputerowo. Ustawa zauważa rolę technologii komputerowych i umożliwia prowadzenia księgowości „bez papieru”, co dotychczas było niemożliwe. Uściślono w niej pojęcie ksiąg rachunkowych, określono sposoby udowodnienia zapisów następujących automatycznie, zdefiniowano wymóg trwałości zapisu maszynowego i określono wymogi w stosunku do elektronicznej dekretacji.

Ustawa nadaje szczególne znaczenie bezpieczeństwu i ochronie danych. Zwiększa znaczenia zapisu elektronicznego i zrównuje moc dowodową zapisu elektronicznego z zapisem tradycyjnym. Zapisy Ustawy o rachunkowości sankcjonują elektroniczne formy potwierdzania tożsamości i są zbieżne z zapisami Ustawy o podpisie elektronicznym.

Ustawa o rachunkowości odwołuje się do informatyki w trzech zasadniczych aspektach:

- a) technologii informatycznego przetwarzania danych,
- b) szczegółowego opisu zawartości systemu informatycznego z uwzględnieniem istotnych elementów jego funkcjonowania,
- c) zabezpieczania zasobów informacyjnych rachunkowości.

Technologii informatycznego przetwarzania danych dotyczą zapisy odnoszące się do kontroli ciągłości zapisów, wielowymiarowości analizy i klasyfikacji zdarzeń. Rzetelność i jasność zapisów i klasyfikacji wynika z art. 4 Ustawy. Zapis jest odzwierciedleniem zdarzenia ekonomicznego o ściśle określonym znaczeniu. System informatyczny ma za zadanie czytelnie i jasno prezentować to znaczenie. Ustawa wprost zawiera zakaz ukrywania faktycznego znaczenia ekonomicznego zapisów, co stanowi odejście od klasycznego rozumienia dokumentowania zdarzeń gospodarczych. Istotnym elementem opisu systemu wspierającego procesy rachunkowości jest udokumentowana jawność procedur, algorytmów, kolejnych etapów przetwarzania danych i nowe wymagania dla elektronicznych dowodów księgowych. Ustawa zatem jest reakcją na problemy dot. niuregulowanych kwestii, a z drugiej strony ułatwia mechanizmy kontroli w zwiększonej liczbie operacji.

## **1. Wykaz wymagań dla systemów informatycznych w świetle zapisów Ustawy**

O księgowości prowadzonej za pomocą komputera możemy mówić wtedy gdy za pomocą komputera prowadzone są conajmniej konta księgi głównej. O systemie zintegrowanym możemy mówić wtedy, gdy za pomocą komputera realizujemy maksymalne spektrum usług, nie tylko finansowo-księgowych, ale także zarządzanie łańcuchem dostaw, polityka kadrowo – płacową, zarządzaniem relacjami z klientem.

Za równoważne z księgami rachunkowymi uważa się oddzielne komputerowe zbiory danych bazy danych lub wyodrębnione ich części, (obojętne jest miejsce ich powstania lub przetworzenia), jeżeli jednostka posiada oprogramowanie umożliwiające uzyskiwanie czytelnych informacji o zapisach dokonanych komputerowo w księgach. Określenie ksiąg zawiera art. 13 Ustawy. W jej świetle księgami podatkowymi są również ewidencja środków trwałych, wartości niematerialnych i prawnych, KPIR, ewidencje, rejestry wyposażenia, zatrudnienia, przebieg pojazdów. Księgi rachunkowe wg. art. 24 powinny być prowadzone rzetelnie, bezbłędnie, sprawdzalnie i bieżąco. Rzetelność ksiąg jest ważniejsza niż ich formalna zgodność z przepisami. Księgi rachunkowe uznaje się za sprawdzalne, jeżeli umożliwiają stwierdzenie poprawności dokonanych w nich zapisów, stanów (sald) oraz działania stosowanych procedur obliczeniowych, a w szczególności:

1. udokumentowanie zapisów pozwala na identyfikację dowodów i sposobu ich zapisania w księgach rachunkowych na wszystkich etapach przetwarzania danych,
2. zapisy uporządkowane są chronologicznie i systematycznie według kryteriów klasyfikacyjnych umożliwiających sporządzenie obowiązujących jednostkę sprawozdań finansowych i innych sprawozdań, w tym - deklaracji podatkowych oraz dokonanie rozliczeń finansowych,
3. w przypadku prowadzenia ksiąg rachunkowych przy użyciu komputera zapewniona jest kontrola kompletności zbiorów systemu rachunkowości oraz parametrów przetwarzania danych,
4. zapewniony jest dostęp do zbiorów danych pozwalających, bez względu na stosowaną technikę, na uzyskanie w dowolnym czasie i za dowolnie wybrany okres sprawozdawczy jasnych i zrozumiałych informacji o treści zapisów dokonanych w księgach rachunkowych.

System powinien zatem kontrolować poprawność i kompletność wprowadzonego dokumentu, uniemożliwiać wprowadzanie dowodów niekompletnych, niepoprawnych merytorycznie lub nieoznaczonych.

Księgi uznaje się za prowadzone bezbłędnie, jeżeli wprowadzono do nich kompletnie i poprawnie wszystkie zakwalifikowane do zaksięgowania w danym miesiącu dowody księgowe, zapewniono ciągłość zapisów oraz bezbłędność działania stosowanych procedur obliczeniowych. Jednostki obowiązane są stosować określone Ustawą zasady rachunkowości, rzetelnie i jasno przedstawiając sytuację majątkową i finansową oraz wynik finansowy, zatem system finansowy powinien być skonstruowany tak, aby spełniał warunki art. 4. Powinien on być zbiorem danych i procedur ujmującym „zapisy zdarzeń w porządku chronologicznym i systematycznym”, stąd też ważnym elementem jest elastyczny dobór numeracji zdarzeń pozwalający na skuteczną kontrolę ciągłości zapisów przy zachowaniu pełnej funkcjonalności rozwiązania. System ma umożliwiać prowadzenie jednolitej polityki prowadzenia rachunkowości, tzn. jednolitego grupowania operacji gospodarczych, metody i forma sporządzania sprawozdań finansowych mają być takie, aby informacja ze sprawozdań była porównywalna z informacją z poprzedniego okresu sprawozdawczego (art. 5 §1). Jeżeli Urząd Skarbowy nie dowiedzie nierzetelności lub wadliwości ksiąg - księgi uznaje się za prowadzone niewadliwie, rzetelnie i prawidłowo (domniemanie prawdziwości).

Ustawa zrównując zapis elektroniczny z zapisem papierowym (z zastrzeżeniem art.13 ust.4) nakłada na systemy informatyczne w całej rozciągłości obowiązki zapisane w art. 4. Z zapisów tych wynika, że wszelkie uproszczenia w prowadzeniu ksiąg rachunkowych za pomocą komputera, mogą być sankcjonowane wyłącznie wtedy, gdy można udowodnić, że nie wywierają istotnie ujemnego wpływu na realizację zapisu art.4 ust.1. Pełną odpowiedzialność za wykonywanie obowiązków w zakresie rachunkowości określonych Ustawą ponosi kierownik jednostki, w tym z tytułu nadzoru, również w przypadku, gdy określone obowiązki w zakresie rachunkowości zostaną powierzone innej osobie za jej zgodą. Przyjęcie odpowiedzialności przez inną osobę powinno być stwierdzone w



formie pisemnej (art. 4 p. 5). W praktyce, polityka w zakresie prowadzenia ksiąg rachunkowych w formie elektronicznej określona jest w zarządzeniu lub uchwale. Należy zadbać o odpowiednie uprawnienia do systemu dla osób upoważnionych.

Przyjęte zasady (politykę) rachunkowości należy stosować w sposób ciągły, dokonując w kolejnych latach obrotowych jednakowego grupowania operacji gospodarczych, (...) tak, aby za kolejne lata informacje z nich wynikające były porównywalne (por. art.5).

Ustawa stawia przed systemem księgowym jeszcze większe wymagania w zakresie automatyzacji związanej z rejestrowaniem, przechowywaniem, przetwarzaniem i prezentacją zdarzeń gospodarczych według ich faktycznego znaczenia ekonomicznego. Istotnym zapisem mającym dalekosiężne konsekwencje praktyczne ma art. 8, który mówi, że należy zapewnić wyodrębnienie w rachunkowości (systemie informatycznym) wszystkich zdarzeń istotnych do oceny sytuacji majątkowej i finansowej oraz wyniku finansowego jednostki. Jest to nowe wyzwanie dla twórców systemów FK. Konsekwencje zapisu w art. 8 można streścić w następującym zdaniu: proste systemy księgowe w znacznej mierze będą musiały przejść głęboką transformację w kierunku większej integracji, skomplikowania i funkcjonalności, zatem system FK którego mocną stroną jest jedynie sprawozdawczość trąci anachronizmem. Art. 8. jest *quasi* „rachunkiem sumienia” za nietrafne inwestycje w systemy finansowo-księgowe, ponieważ kierownik jednostki przedstawia w jakim zakresie zastosowane rozwiązania wpłynęły na wynik finansowy co *explicite* wynika z art. 8 Ustawy. Ocena rozwiązań, a co za tym idzie ocena relacji funkcjonalność/cena, leży w gestii biegłego rewidenta, ponieważ dotyczy sposobu prowadzenia ksiąg rachunkowych i zbadania, w jakim stopniu wdrożenie i eksploatacja rozwiązań zaważyły na wyniku finansowym przedsiębiorstwa. Można zatem założyć, że poważne potraktowanie zapisów Ustawy stwarza szanse na ograniczenie niefrasobliwości w wyborze rozwiązań informatycznych, procesach wdrożeniowych, eksploatacji i administracji systemów księgowych.

Chociaż Ustawa w art. 9 określa język polski jako język prowadzenia ksiąg rachunkowych, to dowody księgowe, operacje rachunkowe, zestawienia dla celów sprawozdawczych winny być przystosowane do prezentacji multiwalutowej. W świetle Ustawy nie jest możliwe prowadzenie ksiąg rachunkowych jedynie w języku obcym. Wielowalutowość ważna jest w kontekście globalizacji i zaawansowanych procesów integracyjnych z UE. Coraz częściej od systemów wymaga się modułów sprawozdawczości opartych o standardy standardy mieszane. np. polskie i US GAAP lub polskie i FRS (brytyjskie).

## 2. Moduły systemu wspierania rachunkowości w firmie

Jednym z podstawowych modułów w systemie księgowym jest dziennik, który powinien zawierać chronologiczne ujęcie zdarzeń, jakie nastąpiły w danym okresie sprawozdawczym. Bez względu na technikę prowadzenia ksiąg rachunkowych dziennik powinien umożliwiać uzgodnienie jego obrotów z

obrotami zestawienia obrotów i sald kont księgi głównej. Zapisy w dzienniku muszą być kolejno numerowane, a sumy zapisów (obroty) liczone w sposób ciągły. Sposób dokonywania zapisów w dzienniku powinien umożliwiać ich jednoznaczne powiązanie ze sprawdzonymi i zatwierdzonymi dowodami księgowymi. Przy prowadzeniu ksiąg rachunkowych przy użyciu komputera zapis księgowy powinien posiadać automatycznie nadany numer pozycji, pod którą został wprowadzony do dziennika, a także dane pozwalające na ustalenie osoby odpowiedzialnej za treść zapisu.

Niemniej ważnym elementem systemu jest księga główna. Konta księgi głównej zawierają zapisy o zdarzeniach w ujęciu systematycznym. Na kontach księgi głównej obowiązuje ujęcie zarejestrowanych uprzednio lub równocześnie w dzienniku zdarzeń, zgodnie z zasadą podwójnego zapisu. Zapisów na określonym koncie księgi głównej dokonuje się w kolejności chronologicznej. Konta ksiąg pomocniczych zawierają zapisy będące uszczegółowieniem i uzupełnieniem zapisów kont księgi głównej. Prowadzi się je w ujęciu systematycznym jako wyodrębniony system ksiąg, kartotek (zbiorów kont), komputerowych zbiorów danych, uzgodniony z saldami i zapisami na kontach księgi głównej. Można tworzyć tabele porównań, wycen, rejestry aktualizowane zdalnie, na podstawie których system będzie dokonywał wyceny uwzględniając zmienne zakłócające np. czas, zużycie, fluktuacje rynkowe, walutowe itp.

Obroty zestawień powinny być zgodne z obrotami dziennika lub obrotami zestawienia obrotów dzienników częściowych. Każdy wydruk dziennika, konta lub zestawienia obrotów i sald powinien być trwale opatrzony nazwą jednostki, rodzaju księgi i programu przetwarzania, zawierać określenie roku obrotowego, okresu sprawozdawczego i daty sporządzenia wydruku składać się z automatycznie numerowanych stron z oznaczeniem pierwszej i ostatniej; kolejne strony powinny być sumowane narastająco w okresie sprawozdawczym. Kontrola ciągłości zapisów oraz przenoszenia obrotów lub sald powinna następować automatycznie. Zestawienia obrotów i sald wymagają wydruku co miesiąc. Zniesiono natomiast taki obowiązek w stosunku do dziennika.

### **3. Sprawozdawczość**

System powinien mieć możliwość porównania bilansu zamknięcia i bilansu otwarcia. Zatwierdzenie bilansu zamknięcia i „przepisanie” go na bilans otwarcia powinno mieć charakter nieodwracalny. Odpowiedni mechanizm bezpieczeństwa powinien umożliwić zrobienie „snapshotu” systemu, poinformować o nieodwracalności operacji i umożliwić zrobienie kopii bezpieczeństwa zapisując dane i stan systemu (kartotek) sprzed operacji zamknięcia okresu sprawozdawczego. Zamknięcie okresów powinno być nieodwracalne, przy czym zamknięcie ksiąg rachunkowych polega na nieodwracalnym wyłączeniu możliwości dokonywania zapisów księgowych w zbiorach tworzących zamknięte księgi rachunkowe, z uwzględnieniem art. 13 ust. 2 i 3. Wadą dzisiejszych systemów jest skomplikowana sprawozdawczość - stąd też system powinien być

zaopatrzone w dodatkowy kreator sprawozdań i rozbudowywalne menu w celu tworzenia raportów nietypowych lub uzupełniania albo rozbudowywania wizualnej strony raportu. Przy prowadzeniu ksiąg rachunkowych przy użyciu komputera należy zapewnić automatyczną kontrolę ciągłości zapisów, przenoszenia obrotów lub sald. Wydruki komputerowe ksiąg rachunkowych powinny składać się z automatycznie numerowanych stron, z oznaczeniem pierwszej i ostatniej oraz być sumowane na kolejnych stronach w sposób ciągły w roku obrotowym. Optymalny system spełnia te założenia w dowolnej jednostce czasu, chociaż nie jest to wymagane przez Ustawę. Ustawa wymaga sporządzenia określonych rodzajów raportów na koniec okresu sprawozdawczego. Zasady dotyczące sporządzania sprawozdań finansowych, gromadzenia i przechowywania dowodów księgowych oraz pozostałej dokumentacji przewidzianej Ustawą, należy kontynuować w kolejnym okresie sprawozdawczym, z zastrzeżeniem art. 8 p. 2.

#### 4. Miejsce prowadzenia ksiąg

Ustawa określa w art. 11 miejsce prowadzenia ksiąg i wskazuje że prowadzi się je w siedzibie jednostki (przedstawicielstwa lub oddziału zagranicznej osoby prawnej). Jednostki posiadające oddziały (zakłady) mogą prowadzić księgi rachunkowe oddziałów (zakładów) w siedzibie jednostki albo oddziałów (zakładów). Księgi rachunkowe mogą być prowadzone na terytorium Rzeczypospolitej Polskiej poza siedzibą jednostki (oddziału, zakładu), w przypadku określonym w art. 13 ust. 2 i 3 oraz w razie powierzenia ich prowadzenia osobie fizycznej, osobie prawnej lub jednostce nie posiadającej osobowości prawnej, uprawnionej do świadczenia takich usług. Należy zapewnić dostępność ksiąg do badania przez upoważnione organy kontroli zewnętrznej w siedzibie jednostki (oddziału, zakładu). W kontekście możliwości outsourcingu informatycznego zapisy Ustawy pozostawiają wiele nierozstrzygniętych wątpliwości.

Komputerowe zbiory danych, niezależnie od ich formy (zbiorów niezintegrowanych, części bazy danych, bazy danych), mogą znajdować się poza jednostką, ale - w myśl art. 11, ust. 3 Ustawy o rachunkowości - tylko na terenie Polski. Już dziś znane są przypadki prowadzenia działalności gospodarczej podmiotów zagranicznych, gdzie znaczna część danych i ich przetwarzanie jest prowadzone poprzez dedykowane łącza, na obcych serwerach. Ustawa sankcjonuje takie rozwiązania pod warunkiem, że na miejscu będą wydruki, dane na nośnikach i odpowiednia infrastruktura zapewniająca lokalne przetwarzanie danych (dane źródłowe i wynikowe (art. 13 p. 3).

O przetwarzaniu możemy mówić wtedy, gdy w Polsce odbywa się:

- wprowadzenie dowodów do systemu,
- przechowywanie oryginałów dokumentów,
- możliwy jest wydruk dowolnych dokumentów za dowolny okres,
- w jednostce są wszystkie niezbędne programy, opisy i dokumentacje,



- jednostka dysponuje sprawozdawczością (wydrukami bądź danymi zapisanymi komputerowo na trwałych nośnikach magnetycznych) za okresy już zamknięte,
- rozwiązania informatyczne (łącza) zapewniają kontrolę tożsamości zapisów w systemie i danych umiejscowionych w jednostce,
- spełnione są wszystkie pozostałe wymogi Ustawy (np. dotyczące bezpieczeństwa, ochrony itp.).

Zapis mówiący o prowadzeniu ksiąg na terytorium RP w kontekście rozwoju technik teleinformatycznych powoduje komplikacje w rozumieniu „fizycznego” miejsca przechowywania ksiąg rachunkowych.

## 5. Dane

Przy prowadzeniu ksiąg rachunkowych przy użyciu komputera za równoważne z dowodami źródłowymi uważa się zapisy w księgach rachunkowych, wprowadzane automatycznie za pośrednictwem urządzeń łączności, komputerowych nośników danych lub tworzone według algorytmu (programu) na podstawie informacji zawartych już w księgach, przy zapewnieniu, że podczas rejestrowania tych zapisów zostaną spełnione co najmniej następujące warunki:

1. uzyskają one trwale czytelną postać zgodną z treścią odpowiednich dowodów księgowych,
2. możliwe jest stwierdzenie źródła ich pochodzenia oraz ustalenie osoby odpowiedzialnej za ich wprowadzenie,
3. stosowana procedura zapewnia sprawdzenie poprawności przetworzenia odnośnych danych oraz kompletności i identyczności zapisów,
4. dane źródłowe w miejscu ich powstania są odpowiednio chronione, w sposób zapewniający ich niezmienność, przez okres wymagany do przechowywania danego rodzaju dowodów księgowych.

Dane powinny zawierać treść przewidzianą dla dowodów księgowych, i umożliwić nadanie im trwale czytelnej postaci oraz ustalenie źródła pochodzenia (osoby wprowadzającej). Dane stanowiące podstawę zapisów w księgach powinny być odpowiednio chronione przez czas obowiązujący do przechowywania dowodów księgowych określonego rodzaju. Powinno być możliwe stwierdzenia zakwalifikowania dowodu do ujęcia w księgach rachunkowych przez wskazanie miesiąca oraz sposobu ujęcia dowodu w księgach rachunkowych (dekretacja), a także podpis osoby odpowiedzialnej za te wskazania. Można zaniechać zamieszczania na dowodzie danych, o których mowa w ust. 1 pkt 5 i 6, jeżeli wynikają one z techniki dokumentowania zapisów księgowych (patrz. Rozp. MF o szczególnych zasadach rachunkowości banków). Ustawa sankcjonuje formy elektronicznego dokumentowania samego procesu wprowadzania danych. Wymaga jednak potwierdzenia niezaprzeczalności dokonania zapisu zdarzenia (np. wystawienia faktury, dekretacji...), np. poprzez mechanizm logowania się do systemu FK. Logowanie się do systemu operacyjnego należy uznać za mechanizm

niewystarczający. Skoro zapisy elektroniczne są zrównane z klasycznymi, zatem podlegają jednocześnie klasycznym mechanizmom kontrolnym osób uprawnionych w całej rozciągłości (również w zakresie kontroli informatycznej). Zastosowanie odpowiednich procedur dostępu zlikwiduje brak odpowiedzialności za pomyłki lub uchybienia. Procedury uwierzytelniania i podpis elektroniczny to dla większości przedsiębiorstw jeszcze zbyt wielki koszt, stąd też istnieje potencjalne niebezpieczeństwo, że zapis o elektronicznym podpisywaniu wprowadzonych dokumentów pozostanie w sferze możliwości nierealizowanych.

## 6. Zapis

Podstawą zapisów w systemie księgowym są:

- sprawdzone dowody źródłowe, zastępcze i zbiorcze wprowadzane manualnie
- dane zewnętrzne pobierane za pomocą urządzeń łączności zewnętrznej
- dowody tworzone automatycznie na podstawie danych wewnętrznych i algorytmów

Warunkiem utrzymywania zasobów informacyjnych systemu rachunkowości w formie określonej w art. 13. ust. 2, jest posiadanie przez jednostkę oprogramowania umożliwiającego uzyskiwanie czytelnych informacji w odniesieniu do zapisów dokonanych w księgach rachunkowych, poprzez ich wydrukowanie lub przeniesienie na inny komputerowy nośnik danych. Komputerowy nośnik danych ma zapewnić trwałość zapisu informacji przez czas nie krótszy od wymaganego dla przechowywania ksiąg rachunkowych. Zostało wyeliminowane określenie „trwały nośnik danych”, pojawia się natomiast nośnik zapewniający „trwałość zapisu” (art. 6.13), który wprowadza uniezależnienie się od zmiennych technologii informatycznych przy założeniu pełnego ich wykorzystania. Dzięki podniesieniu poziomu przechowywania dokumentacji (przenoszeniu na nośniki gwarantujące trwałość zapisu w wymaganym okresie) kierownik jednostki uniknie kosztów związanych z zastosowaniem art. 72 ust.1 (drukowanie ksiąg)

Ustawa równoważy zapisy tradycyjne i komputerowe pod warunkiem:

1. uzyskania w dowolnym momencie trwałej czytelnej postaci (wydruku), zgodnych z treścią odpowiednich dowodów księgowych,
2. możliwości stwierdzenia źródła ich pochodzenia oraz ustalenie osoby odpowiedzialnej za ich wprowadzenie (np. porównania z oryginałem), można zaniechać dekretacji na dowodzie
3. stosowania procedur zapewniających sprawdzenie poprawności przetworzenia danych oraz kompletności i identyczności zapisów,
4. iż dane źródłowe w miejscu ich powstania są odpowiednio chronione, zapewniając ich niezmienną przez czas wymagany do przechowywania danego rodzaju dowodów księgowych.

Widać zatem wyraźnie, że Ustawodawca równoważy funkcjonalnie formę wydruku i zapisu elektronicznego z zastrzeżeniem ust. 4. art. 13 i mówi, że księgi rachunkowe, z uwzględnieniem techniki ich prowadzenia, powinny być:

1. trwale oznaczone nazwą (pełną lub skróconą) jednostki, której dotyczą (każda księga wiązana, każda luźna karta kontowa, także jeżeli mają one postać wydruku komputerowego lub zestawienia wyświetlanego na ekranie monitora komputera), nazwą danego rodzaju księgi rachunkowej oraz nazwą programu przetwarzania,
2. wyraźnie oznaczone co do roku obrotowego, okresu sprawozdawczego i daty sporządzenia,
3. przechowywane starannie w ustalonej kolejności.

Dokument papierowy będzie w całkiem niedalekiej przyszłości miał poważnego konkurenta w postaci dokumentu elektronicznego. Ustawa o rachunkowości komplementarnie z Ustawą o podpisie elektronicznym jest traktowana jako szansa wirtualizacji obrotu.

Zrównanie mocy podpisu elektronicznego z podpisem na dokumencie oznacza dopuszczenie do obiegu wszelkiego typu dokumentów elektronicznych i zrównanie ich z tradycyjnymi dokumentami w formie papierowej. Zrównanie dokumentu elektronicznego i klasycznego rodzi pytanie o zdolność podmiotów publiczno-prawnych do elektronicznej weryfikacji tożsamości oraz o istnienie skutecznych mechanizmów kontrolnych w ręku wykwalifikowanych kadr audytorskich. Pytania te wydają się dzisiaj raczej retoryczne.

Zapisów w księgach rachunkowych dokonuje się w sposób trwały, bez pozostawiania miejsc pozwalających na późniejsze dopiski lub zmiany. Przy prowadzeniu ksiąg rachunkowych przy użyciu komputera należy stosować właściwe procedury i środki chroniące przed zniszczeniem, modyfikacją lub ukryciem zapisu.

Trwałość w rozumieniu Ustawy sprzed nowelizacji zapewniał jedynie nośnik CD-R, ponieważ był to jedyny nośnik z którego usunięcie danych było niemożliwe bez fizycznego jego zniszczenia. Obecnie Ustawa dopuszcza wielość rozwiązań, czym otwiera się na nowe technologie. Składowanie danych jest dzisiaj bardzo rozwinięte poprzez skomplikowane mechanizmy dostępu, zabezpieczeń, redundancji. Zapis księgowy powinien zawierać co najmniej:

1. datę dokonania operacji gospodarczej,
  2. określenie rodzaju i numer identyfikacyjny dowodu księgowego stanowiącego podstawę zapisu oraz jego datę, jeżeli różni się ona od daty dokonania operacji,
  3. zrozumiały tekst, skrót lub kod opisu operacji, z tym że należy posiadać pisemne objaśnienia treści skrótów lub kodów,
  4. kwotę i datę zapisu,
  5. oznaczenie kont, których dotyczy.
  6. dane pozwalające ustalić osobę odpowiedzialną za treść zapisu (art. 14.4)
- Art. 24 zawiera bezwzględny zakaz niszczenia, ukrywania, modyfikacji, ukrywania faktycznej treści znaczeniowej zapisu.



## 7. Opis systemów

Bez względu na to czy księgi są prowadzone za pomocą komputera czy inaczej, należy też „opisać system służący ochronie danych i ich zbiorów, w tym dowodów księgowych, ksiąg rachunkowych i innych dokumentów stanowiących podstawę dokonanych w nich zapisów”. Opis dotyczy ksiąg rachunkowych (w szczególności ZPK), systemu informatycznego i systemu ochrony. Opis taki w formie pisemnej powinien być na bieżąco aktualizowany, powinien zawierać szczegółowy opis środowiska informatycznego, zastosowanych rozwiązań sprzętowych i programowych. Należy sporządzić wykaz zbiorów tworzących księgi rachunkowe, określić strukturę każdego zbioru, opisać powiązania zachodzące między zbiorami i ich funkcje w organizacji całości ksiąg rachunkowych oraz w procesach przetwarzania danych księgowych. W dokumentacji takiej jednostka powinna określić, zgodnie z art. 10 ust. 1 pkt 3 Ustawy o rachunkowości, sposób prowadzenia ksiąg rachunkowych.

Opis sposobu prowadzenia ksiąg rachunkowych, powinien obejmować:

1. wykaz zbiorów danych tworzących księgi rachunkowe na komputerowych nośnikach danych z określeniem ich struktury, wzajemnych powiązań oraz ich funkcji w organizacji całości ksiąg rachunkowych i w procesach przetwarzania danych,
2. opis systemu informatycznego, obejmujący wykaz programów, procedur lub funkcji, w zależności od struktury oprogramowania, wraz z opisem algorytmów i parametrów oraz programowych zasad ochrony danych, w tym w szczególności metod zabezpieczenia dostępu do danych i systemu ich przetwarzania, a ponadto określenie wersji oprogramowania i daty rozpoczęcia jego eksploatacji.

Dokumentację przyjętego sposobu prowadzenia rachunkowości należy przechowywać przez okres nie krótszy od 5 lat od upływu jej ważności.

Zamysłem Ustawodawcy jest taki opis systemu aby zapewniał jasną i czytelną informację o konstrukcji systemu, użytych funkcjach i procedurach (algorytmy przetwarzania) sposobie eksploatacji, konserwacji i zabezpieczenia systemu. Metodyka opisu pozostaje w gestii kierownika jednostki. Algorytm może być opisany słownie, wzorem matematycznym, schematem blokowym lub tablicą decyzyjną. Ustawodawca nie określa wymaganej formy. Nie chodzi bowiem o dosłowność, ale określenie bez żadnej wątpliwości czy zastosowany algorytm czytelnie i w sposób bezbłędny realizuje swoje zadania. Cechami takiego opisu powinny być: dokładność, adekwatność, przejrzystość. Opis środowiska informatycznego powinien obejmować: sprzęt (hardware), środowisko sieciowe (okablowanie i urządzenia), systemy operacyjne, system księgowy, system zapewnienia bezpieczeństwa. Jednostka może sama rozpoznać struktury albo zwrócić się do autora oprogramowania.

Podstawowym pytaniem na jakie powinien odpowiedzieć kierownik jednostki to: czy jesteśmy w stanie własnymi siłami sporządzić taki opis, na ile będzie on adekwatny, nie budzący wątpliwości, dokładnie precyzujący wymagane Ustawą problemy. Problematicznym wydaje się określenie szczegółowości takiego opracowania, zwłaszcza w kontekście poufności informacji i praw autorskich.

Szczegółowy opis rozwiązań, np. kryptograficznych, sam w sobie już mógłby spowodować osłabienie ich zastosowania, stąd też rozsądnym wydaje się wypracowanie takiej formy opisu, aby nie naruszała interesów żadnej ze stron. Opis infrastruktury powinien być uzupełniony o opis ochrony systemu informatycznego i zawierać opis zastosowanej ochrony dostępu do danych i oprogramowania w postaci haseł i kluczy oraz związanych z tym zabiegów organizacyjnych, opis zabezpieczeń antywirusowych, działań organizacyjnych, opis procedur na wypadek zakłóceń lub awarii, umów na awaryjne przetwarzanie w innej jednostce, na konserwację, modyfikację i rozwój systemu. Artykuł 13 Ustawy może być potraktowany jako wskazówka konstrukcji ZSI.

## 8. Ochrona danych i zbiorów

Art. 71. mówi, że przy prowadzeniu ksiąg rachunkowych przy użyciu komputera ochrona danych powinna polegać na stosowaniu odpornych na zagrożenia nośników danych, na doborze stosownych środków ochrony zewnętrznej, na systematycznym tworzeniu rezerwowych kopii zbiorów danych zapisanych na nośnikach komputerowych, pod warunkiem zapewnienia trwałości zapisu informacji systemu rachunkowości, przez czas nie krótszy od wymaganego do przechowywania ksiąg rachunkowych oraz na zapewnieniu ochrony programów komputerowych i danych systemu informatycznego rachunkowości, poprzez stosowanie odpowiednich rozwiązań programowych i organizacyjnych, chroniących przed nieupoważnionym dostępem lub zniszczeniem. Ustawodawca dzieli rozwiązania na: programowe, sprzętowe, techniczne i organizacyjne. Rozumie je jako właściwe procedury i środki chroniące przed zniszczeniem, modyfikacją lub ukryciem zapisu.

Ochrona danych powinna polegać na:

- stosowaniu odpornych na zagrożenia nośników danych,
- doborze środków ochrony zewnętrznej,
- systematycznym tworzeniu kopii zapasowych (backup),
- używaniu nośników odpornych na zagrożenia,
- zapewnieniu trwałości zapisu informacji systemu rachunkowości przez czas nie krótszy niż 5 lat.

## 9. Metody zabezpieczenia i dostępu

Wyżej wymienionym mechanizmom zabezpieczenia podlegają księgi rachunkowe, dowody księgowo-dokumenty inwentaryzacyjne i sprawozdania finansowe. Przykłady zagrożeń na jakie powinno się zwrócić uwagę to np.:

- Błędy w urządzeniach
- Błędy projektowe
- Ukryte wady fabryczne
- Starzenie się podzespołów
- Błędy obsługi

- Wyładowania atmosferyczne
- Zasilanie
- Zdarzenia losowe
- Akty sabotażu lub terroryzmu
- Brak lub uszkodzenie backupu

Optymalnym rozwiązaniem w zakresie bezpieczeństwa i ochrony jest konstrukcja polityki bezpieczeństwa. Polityka bezpieczeństwa powinna być spójna, kompleksowa, kompletna i wdrożona. Zasady konstruowania polityki bezpieczeństwa określone są w standardach międzynarodowych. W wymiarze systemów informatycznych bezpieczeństwo wyróżniamy bezpieczeństwo fizyczne, sieciowe, platformy, danych, aplikacji, ciągłości procesów biznesowych. Kontrolę wymogów bezpieczeństwa mogą spełniać mechanizmy informatyczne np:

- procedury powtarzania zapisu,
- procedury identyfikacji, autoryzacji, autentyfikacji,
- zastosowanie sum kontrolnych ,
- kluczy identyfikacyjnych,
- podpisu elektronicznego,
- zabezpieczenia sprzętowe (np. token, HASP),
- zabezpieczenia organizacyjne, (monitoring, backup),
- zabezpieczenia biometryczne.

Za znamiona wiarygodności w zakresie bezpieczeństwa systemów przyjmuje się:

- gotowość do działania (długość ewentualnych przestojów),
- niezawodność i bezawaryjność,
- bezpieczeństwo środowiska i zasobów,
- zdolność i skuteczność mechanizmów ochrony,
- prawdziwość i legalność,
- zgodność z przepisami Ustawy.

Sama technologia ani polityka bezpieczeństwa nie zapewnią skutecznej ochrony. Nadużywanie uprawnień jest nagminne, pomimo wprowadzania zakazów. Przedsiębiorstwa powinny opracowywać i stale udoskonalać kompleksowe podejście do zagadnień bezpieczeństwa i ochrony systemów informatycznych, obejmujące zarówno wymiar technologiczny, jak i organizacyjny. Stąd też obowiązki w zakresie ochrony powinny obejmować m.in.

- opracowywanie oraz wdrażanie polityki bezpieczeństwa, w tym polityki bezpieczeństwa systemów i sieci teleinformatycznych,
- monitorowanie stanu jej realizacji, opracowywanie wzorów procedur awaryjnych i zastępczych,
- przestrzeganie ich wdrożenia i stosowania,
- wykonywanie kontroli jednostek organizacyjnych w zakresie organizacji zarządzania, administrowania, funkcjonowania i eksploatacji systemów zabezpieczeń technicznych,



- kontrolowanie stosowania procedur bezpieczeństwa, wykonywanie audytów bezpieczeństwa,
- projektowanie, przygotowanie do wdrożenia, zarządzanie, obsługę i nadzór nad eksploatacją zabezpieczeń kryptograficznych w systemach i sieciach teleinformatycznych oraz w systemach łączności,
- bieżący nadzór nad ochroną obiektów, w tym nad stosowaniem środków ochrony fizycznej komputerowych nośników danych,
- kontrola i monitorowanie oraz zapewnienie ciągłości funkcjonowania systemów teleinformatycznych,
- opracowywanie i wprowadzanie do użytkowania procedur eksploatacji systemów informatycznych i telekomunikacyjnych,
- administracja i konserwacja środowiska systemu teleinformatycznego

Bezpieczeństwo zwiększa się poprzez: minimalizację wystąpienia awarii (analiza ryzyka) i stosowanie mechanizmów redundancji (nadmiarowości). Analizując bezpieczeństwo infrastruktury powinniśmy brać pod uwagę:

- ciągłość komunikacji (stan łącz operatorów zewnętrznych),
- redundancję: zasilacze, dyski (mirroring, RAID), hot-plug, clusters, UPS,
- eliminować „punkty krytyczne”,
- podawanie przez producenta parametru MTBF (średni czas pracy bezawaryjnej),
- możliwość określenia „Recovery time” – przywrócenia sprawności po awarii,
- możliwość określenia „Recovery point” – (ile danych stracimy?),
- rozproszenie zasobów w kilku miejscach (remote backup sites),
- nowe typy analizy systemów: (back testing, if scenario, stress testing),
- możliwość przeniesienia danych do ośrodka zapasowego.

Dla rozwiązań o mniejszej skali należy zadbać przynajmniej o podstawowe elementy bezpieczeństwa, takie jak np:

- automatyczna statystyka eksploatacji systemu,
- mechanizmy identyfikacji dowodów i sposobu ich zapisania w księgach rachunkowych na wszystkich etapach przetwarzania danych, tak aby istniał tzw. ślad rewizyjny,
- zapewnienie ciągłości (poprzez jednolitość numeracji) i chronologii (uniemożliwienie wprowadzania dowodów z wsteczną datą),
- kontrola kompletności zbiorów systemu rachunkowości oraz parametrów przetwarzania danych,
- format plików powinien być nieodczytywalny dla programów narzędziowych typu Norton Utility, Norton Commander, notepad itp.,
- odpowiednie zabezpieczenie baz danych przed nieuprawnioną edycją za pomocą komend SQL,
- ochrona w czasie okresu sprawozdawczego,
- ochrona po zamknięciu okresu sprawozdawczego.

Dla rozwiązań pretendujących do miana systemów zintegrowanych zwłaszcza obsługujących podmiot w wielu lokalizacjach geograficznych, dobrym

rozwiązaniem są systemy ochrony aplikacji rozproszonej. W przypadku danych finansowych zachowanie danych o transakcjach jest absolutnie priorytetowe, nawet kosztem przestoju systemu.

Księgi rachunkowe mogą mieć formę, z zastrzeżeniem art. 13 ust. 2 i 3, zbiorów utrwalonych na nośnikach komputerowych, pod warunkiem stosowania rozwiązań wymienionych w art. 71 ust. 2.

Jeżeli system ochrony zbiorów danych rachunkowości, utrwalonych na nośnikach komputerowych, nie spełnia wymagań określonych w art. 71 ust. 2, zapisy te powinny być wydrukowane w terminach przewidzianych w art. 13 ust. 6. Przechowywanie ksiąg rachunkowych na innym nośniku niż wymieniony w ust. 2, jest dopuszczalne pod warunkiem zapewnienia odtworzenia ksiąg w formie wydruków. Treść dowodów księgowych może być przeniesiona na nośniki danych, pozwalające zachować w trwałej postaci zawartość dowodów. Warunkiem stosowania tej metody przechowywania danych jest posiadanie urządzeń pozwalających na odtworzenie dowodów w postaci wydruku, o ile inne przepisy nie stanowią inaczej. Aktualnie funkcjonujące systemy informatyczne, o ile dość dobrze spełniały warunki Ustawy sprzed nowelizacji (pomogła temu zjawisku ścisła zależność rachunkowości i prawa podatkowego), o tyle słabo wypadają w rankingach zabezpieczeń. Stanowiły one tzw. back-office organizacji i funkcjonowały w oderwaniu od innych procesów biznesowych organizacji. Wdrożenie zapisów Ustawy daje szansę na zmianę wyżej opisanej sytuacji.

## 10. Dokumentacja

Ustawa o rachunkowości nakłada na kierownika jednostki obowiązek opracowania dokumentacji przyjętych zasad rachunkowości, ze szczególnym uwzględnieniem systemu przetwarzania danych stosowanego przez jednostkę oraz przyjętego przez nią systemu ochrony tych danych i ich zbiorów. Celem dokumentacji jest zapewnienia wystarczającego stopnia poprawności i wiarygodności ksiąg rachunkowych prowadzonych techniką komputerową. Ustawodawca nie wskazuje formy tej dokumentacji. Może to być np. zbiór (\*.hlp), baza danych, kontekstowa pomoc. Niewykluczone, że dokumentacja systemu będzie umieszczana na stronie WWW producenta, z odpowiednimi mechanizmami uwierzytelniania osób upoważnionych, chociaż obecnie Ustawa nie przewiduje takiej możliwości.

Ustawa nakłada na właściciela oprogramowania do korzystania z odpowiedniej wersji programu do wspierania procesów rachunkowości w firmie. Dokumentacja i opis, o którym mowa w art. 10, powinny obejmować ciągłość numeracji wersji, datę, od której obowiązuje nowa wersja, odróżnienie wersji testowych, datę rozpoczęcia eksploatacji, bieżący numer wersji. Niektórzy dostawcy na stronie WWW publikują aktualny numer wersji oraz udostępniają możliwość elektronicznego uaktualniania wersji. Producent ma obowiązek publikacji „łat” i odpowiedni sposób dokumentowania poprawek. W ramach umowy o przystosowywanie oprogramowania do zmieniających się potrzeb

czasami producent powinien zapewnić ewidencję kolejnych wersji oprogramowania i daty ich wdrożenia.

## 11. Badanie sprawozdania finansowego

Opinia biegłego rewidenta, powinna w szczególności stwierdzać, czy badane sprawozdanie finansowe zostało sporządzone na podstawie prawidłowo prowadzonych ksiąg rachunkowych.

Problemy, na jakie można natrafić najczęściej w badaniu rewidenckim, to:

- działania niezgodne z założeniami,
- bezużyteczność i nadmiarowość informacji,
- niedopracowane szczegóły,
- błędy interfejsu,
- bezskuteczność, niecelowość i nieterminowość danych wyjściowych.

Badanie biegłego ma na celu weryfikację takich elementów, jak:

- kompletność (fizyczna zgodność dokumentacji z zapisami elektronicznymi)
- dokładność (stwierdzenie czy całe dokumenty są wprowadzone do systemu)
- autoryzacja (nazwa programu, data, czas, osoba odpowiedzialna za ich wprowadzenie).

Podstawowymi kryteriami w opinii biegłego rewidenta o zgodności systemu ze znowelizowaną Ustawą o rachunkowości jest stwierdzenie poprawności, kompletności i wiarygodności wprowadzanych danych, przetwarzania i prezentacji wyników, ustalenie źródeł pochodzenia zapisu i ustalenie ścieżki rewizyjnej, poprawności, kompleksowości i skuteczności ochrony danych i zasobów. Są to zatem krytyczne dla całości badania sprawozdania finansowego, które *explicite* powinny zostać zawarte w opinii biegłego rewidenta. Naprawdę jednak dla rewidenta jest najważniejsze co innego: czy w firmie zapewniono na każdym etapie przetwarzania danych dostateczną kontrolę spójności, rzetelności i integralności danych, kontrolę systemów informatycznych przed dokonywaniem nielegalnych modyfikacji w oprogramowaniu, czy istnieje podział funkcji, a więc hierarchizacja dostępu do systemu, oraz czy limitowany jest dostęp do nośników, dokumentacji i sprzętu.

Rewident powinien też stwierdzić, czy otwarcie i zamknięcie ksiąg rachunkowych zostały w świetle znowelizowanej Ustawy dokonane poprawnie, i czy podsystemy informatyczne rachunkowości wymuszają kompletność i poprawność udokumentowania wszystkich operacji gospodarczych wprowadzonych do systemu. Powinien on także znać sposoby ochrony danych i ich przechowywania i sprawdzić praktycznie, czy dane można przenieść na inny nośnik, czy po prostu wydrukować. Niewykluczone jest powstanie w przyszłości modułów audytorskich jako dodatku do systemów rachunkowych wykorzystywanych w pracach rewidenckich.

Biegły rewident staje się często partnerem i doradcą firmy. Rewizja finansowa postrzegana do tej pory jako nieproduktywna konieczność lub zbędny



koszt i stres staje się nieodzownym narzędziem kompleksowej analizy kondycji finansowej przedsiębiorstwa i kultury techniczno-organizacyjnej

Samo badanie sprawozdań finansowych ewoluuje w kierunku audytu, którego podstawą jest analiza ryzyka, optymalizacja procesów w organizacji i zarządzaniu przedsiębiorstwem oraz wszechstronna ocena organizacji. Ma także wpływ na wzrost wartości badanej firmy, wydobywając wiedzę z organizacji, która jest wartością dodaną.

## **12. Rozporządzenie Ministra Finansów z dnia 10 grudnia 2001 r. sprawie szczególnych zasad rachunkowości banków**

Rozporządzenie 7 razy odnosi się do technik komputerowych, przy czym nie wprowadza oczekiwanych uściśleń, a raczej powtarza wymagania już zdefiniowane w noweli Ustawy. Dotyczą one zasady zachowania ciągłości zapisów, bezbłędności działania systemów i identyfikacji osób odpowiedzialnych za wprowadzanie danych do systemu.

Księgi rachunkowe uznaje się za prowadzone bezbłędnie, jeżeli wprowadzono do nich kompletnie i poprawnie wszystkie zakwalifikowane do księgowania w danym dniu operacyjnym dowody księgowe, zapewniono ciągłość zapisów oraz bezbłędność działania stosowanych procedur obliczeniowych. Podstawą zapisów w księgach rachunkowych mogą być zarówno zbiory danych sporządzone za pośrednictwem technik komputerowych, jak również zbiory danych otrzymywane przez bank w formie elektronicznej. (§ 11.3) Dowody księgowe sporządzane przy użyciu techniki komputerowej oznaczają się w sposób zapewniający możliwość stwierdzenia ich autentyczności. Przy prowadzeniu ksiąg rachunkowych przy użyciu komputera zapisowi księgowemu wymagany jest numer identyfikacyjny, nadawany automatycznie, pod którym został wprowadzony do dziennika, i który umożliwi ustalenie osoby odpowiedzialnej za treść zapisu (§ 7.3.)

Wymaganie uzgodnienia w zakresie § 10.7 spowodowało konieczność uzgodnienia między bankami numerów rachunków, co spowodowało kolejną zmianę postaci rachunków bankowych.

Spójność z Ustawą można zauważyć poprzez zaakceptowanie techniki dokumentowania zapisów. Wg rozporządzenia można zaniechać zamieszczania na zleceniu płatniczym danych, o których mowa w ust. 2 pkt 5 i 6, jeżeli wynikają one z techniki komputerowego dokumentowania zapisów księgowych. Nie podlega opatrzeniu odbitką stempla dziennego zestawienie sporządzane automatycznie techniką komputerową w sposób określony dla uczestników izby rozliczeniowej, na podstawie dowodów zarejestrowanych do zapisania lub zapisanych w księgach rachunkowych banku sporządzającego.

Rozliczenia pieniężne bezgotówkowe przeprowadzane są przy użyciu papierowych lub elektronicznych nośników informacji, w tym również za pomocą kart płatniczych. Oświadczenia woli składane w związku z dokonywaniem czynności bankowych mogą być wyrażone za pomocą elektronicznych nośników

informacji (patrz art. 7. 1 Prawo Bankowe). Związane z czynnościami bankowymi dokumenty mogą być sporządzane za pomocą elektronicznych nośników informacji, jeżeli dokumenty te zostaną w sposób należyty utrwalone i zabezpieczone. Jeżeli Ustawa zastrzega dla czynności prawnej formę pisemną, uznaje się, że czynność dokonana w formie, o której mowa w ust. 1, spełnia wymagania formy pisemnej.

Warunkiem honorowania takiego zapisu jest uzyskanie przez zbiór trwale czytelnej postaci, odpowiadającej treści dowodów księgowych i możliwe jest stwierdzenie źródła ich pochodzenia oraz ustalenie osoby odpowiedzialnej za ich wprowadzenie (§8). Nakłada to na system bezwzględny obowiązek implementacji mechanizmów służących weryfikacji tożsamości osób wprowadzających dane do systemu. Unikalny identyfikator i hasło jest „pieczęcią” potwierdzającą, kto, co i kiedy wprowadził do systemu. W powiązaniu z metodami szyfrowania dostępu może to stanowić silny mechanizm bezpieczeństwa, powszechnie stosowany już od dawna, a usankcjonowany formalnie Ustawą. Mechanizm uwierzytelniania jest ważny elementem tzw. śladu rewizyjnego.

W myśl Ustawy Bank powinien posiadać procedury zapewniające sprawdzenie poprawności przetwarzania zbioru danych oraz kompletność i identyczność zapisów. Dane źródłowe podlegają ochronie w miejscu ich powstania, w sposób zapewniający ich niezmienność przez okres wymagany do przechowywania danego rodzaju dowodów księgowych. Wg. § 8. niedopuszczalne jest nanoszenie poprawek na zbiorczych zestawieniach sporządzanych techniką komputerową. Banki zapewniają jednocześnie prawidłowy obieg i kontrolę dowodów księgowych bez względu na technikę dokumentowania zdarzeń rachunkowych. Wydaje się, że w świetle Ustawy implikuje to kompleksowy audyt informatyczny systemów wspierających procesy rachunkowości w organizacji.

### 13. Podsumowanie

Nowelizacja Ustawy nadaje nowe znaczenie mechanizmom informatycznym w obszarze biznesowym organizacji.

Do mocnych cech noweli Ustawy należy zaliczyć przywiązanie większej wagi do faktu obecności technik komputerowych we współczesnej rachunkowości. Technika ta obecna jest szczególnie mocno, w czasie, gdy wzrasta ranga rachunkowości jako podstawowego narzędzia zarządzania organizacją. Ustawa pozytywnie i dość spójnie odnosi się do technik informatycznych, określa systemowo wymagania wobec systemów wspierających procesy rachunkowości oraz, co jest bardzo istotne, nie narzuca szczegółowych rozwiązań, pozostawiając wiele swobody interpretacyjnej. Niektórzy odbiorcy we wspomnianej ogólności widzą rażącą ogólnikowość i to, co w pierwotnej formie miało być atutem Ustawy, obecnie może być poczytywane za jej słabość. Niewątpliwie brak jest rozporządzeń w zakresie sposobu kontroli rozwiązań informatycznych. Z pewnością zmiany wprowadzone nowelą Ustawy w swoim wymiarze ogólnym idą



we właściwym kierunku, chociaż do doskonałości mają jeszcze przed sobą całkiem niemałą drogę.

Wydaje się, że Ustawa ogólnie zarysowuje prawidłowy kierunek zmian, chociaż niedociągnięciami są z pewnością: brak odniesień do poprawnej obsługi procesów biznesowych, brak zapisu o rachunkowości jako modelu wspierania celów strategicznych organizacji, zarządzaniu ryzykiem operacyjnym w rachunkowości, brak odwołania się do Ustawy o podpisie elektronicznym. Wydaje się że unikając narzucania rozwiązań szczegółowych, Ustawodawca zatroszczył się o wysoki stopień elastyczności Ustawy.

Czynności kontrolne w zakresie oceny prawidłowości prowadzenia ksiąg (systemów informatycznych) zostały włożone na barki biegłych rewidentów, których merytoryczne przygotowanie jest w pewnym zakresie na pewno kwestią dyskusyjną. W przeważającej większości przypadków podstawą badania sprawozdań finansowych są wydruki papierowe. Rozszerzenie zakresu sprawozdania finansowego sprawia, że istnieje realne zagrożenie marginalizacji oceny środowiska i systemów księgowych pod kątem zgodności z zapisami Ustawy. Brak jest w Ustawie *explicite* odniesienia do audytu informatycznego, który powinien być wykonywany przez audytorów informatycznych o potwierdzonych umiejętnościach i kwalifikacjach, brak jest również odniesienia do standardów międzynarodowych w zakresie kontroli i audytu systemów informatycznych.

Rozwiązania Ustawowe, jeżeli zostaną potraktowane z więcej niż należytą starannością, przysporzą niektórym organizacjom sporo kłopotów, niemniej jednak wydaje się że bardziej realnym zagrożeniem jest zbyt luźne stosowanie Ustawy i przez nikogo nie kontrolowana niestaranność realizacji jej zapisów. Ustawa wymusza na kierownictwie opracowanie i przyjęcie zasad zawartych w dokumentacji systemu rachunkowości, a także dokumentacji i zabezpieczeń technicznych w zakresie zastosowania eksploatacji narzędzi informatycznych.

System finansowo księgowy przedsiębiorstwa należy do systemów „silnej użyteczności materialnej”. Przepisy Ustawy o rachunkowości wprowadzone i egzekwowane zbyt rygorystycznie mogą uczynić firmową egzystencję koszmarem (strajk włoski). Już dziś publikatorzy wydawnictw rachunkowych stwierdzają znaczną dowolność w stosowaniu i interpretowaniu przepisów Ustawy. Kolejnym problemem jest brak adekwatnych przepisów prawnych, zwłaszcza wykonawczych np. dotyczących kwestii zabezpieczeń, niespójne przepisy prawne itp.

Na rynku aplikacji jest wiele rozwiązań i trudno wybrać rozwiązanie optymalne, dostosowane całkowicie do specyfiki i potrzeb firmy. Proces wroźeniowy jest nierozzerwalnie związany ze specyfiką organizacji w której system FK funkcjonuje.

Pozytywny wymiar ma fakt, że Ustawa porządkuje i docelowo zapobiega rozwiązaniom prowizorycznym. Uderza przede wszystkim w tych którzy swobodnie traktują prawidłowe prowadzenie ksiąg rachunkowych lub wykorzystują rozwiązania „chałupnicze”, niestaranne, niewłaściwie zaprojektowane, wdrażane lub chronione. Ustawa zmusza w końcu odbiorcę do analizy własnego sposobu prowadzenia rachunkowości, opracowaniu procedur



kontroli wewnętrznej w zakresie procesów rachunkowych. W świetle zapisów Ustawy wybór i eksploatacja informatycznego systemu księgowego nareszcie może będzie się opierał na gruntownej analizie systemowej, a nie na tzw. skutecznych metodach marketingowych i, o czym Ustawa, niestety, nie mówi wprost, poprzedzony będzie audytem informatycznym.

Sama Ustawa, o ile zostanie potraktowana z należytą powagą, może nie tylko zwiększyć konkurencyjność produktów dostarczanych na rynek oprogramowania, ale również wyeliminować dostawców rozwiązań niekompletnych, słabych jakościowo lub w skrajnym przypadku niezgodnych z wymaganiami jakie nakłada Ustawa o rachunkowości. Konsekwencją Ustawy może stać się jeszcze mocniejsze zacieśnienie współpracy z producentem oprogramowania, co na pewno będzie dodatnio skorelowane z podnoszeniem kultury organizacyjno – technicznej firmy i budowaniem zrębów społeczeństwa informatycznego.

Oprócz dokumentów elektronicznych w komputerowych systemach prowadzenia rachunkowości istnieje cały szereg dokumentów pomocniczych, które w istotny sposób istotny wpływają lub wręcz tworzą znaczenie ekonomiczne zdarzeń. Pojawia się zatem potrzeba stworzenia systemu zarządzania dokumentami elektronicznymi zintegrowanego z systemem rachunkowości, rozwoju i poprawy funkcjonalności modułów budżetowania, wspieranie rachunkowości ochrony środowiska. Systemy rachunkowości powinny się pogodzić z tym, że przestały być centrum zainteresowania. Są ważne i potrzebne, natomiast funkcjonują jako część w otoczeniu równie ważnych mechanizmów i narzędzi, które dopiero zintegrowane generują wartość dodaną. Oczekiwanie że system księgowy załatwi wszystko, jest nieporozumieniem. Jest szansa, że uporządkowanie środowiska informatycznego w którym funkcjonuje system finansowo – księgowy zapoczątkuje analogiczny proces wobec innych systemów.

Ustawa jest z pewnością dobrym krokiem prowadzącym o powstania sytuacji, w której to nowoczesny, zintegrowany system informatyczny stanie się kluczowym zadaniem dla przedsiębiorstwa, a wbudowany w niego system FK, jako kompleksowe narzędzie wspierania procesów rachunkowych organizacji, będzie jednocześnie systemem inżynierii finansowej potrafiącym minimalizować ryzyko podejmowania decyzji.

## Literatura

1. USTAWA z dnia 29 września 1994 r. o rachunkowości. (Dz. U. z 1994 r. Nr 121, poz. 591, z 1997 r. Nr 32, poz. 183, Nr 43, poz. 272, Nr 88, poz. 554, Nr 118, poz. 754, Nr 139, poz. 933 i 934, Nr 140, poz. 939, Nr 141, poz. 945, z 1998 r. Nr 60, poz. 382, Nr 106, poz. 668, Nr 107, poz. 669, Nr 155, poz. 1014, z 1999 r. Nr 9, poz. 75 i Nr 83, poz. 931 oraz z 2000 r. Nr 60, poz. 703) + zmiany dokonane w 2000 r. przyjęte przez Sejm i Senat RP w X 2000' r.
2. USTAWA z dnia 29 sierpnia 1997 r. Prawo bankowe. (Dz. U. z dnia 21 listopada 1997 r.)

3. ROZPORZĄDZENIE MINISTRA FINANSÓW z dnia 10 grudnia 2001 r. w sprawie szczególnych zasad rachunkowości banków.
4. Depa A., *Rachunkowość informatyczna pod rządami znowelizowanej Ustawy o rachunkowości* [w:] Monitor Rachunkowości i Finansów 2001,5(29).
5. Mejsnner M., *Księgowy z komputerem*, Teleinfo 28/1999
6. Welenc P., *Wymagania stawiane systemom informatycznym w świetle znowelizowanej Ustawy o Rachunkowości i przepisów wykonawczych*, [w:] *Zmiany szczególnych zasad rachunkowości instytucji finansowych a systemy informatyczne*, Warszawa 2001, Wyd. Software Konferencje Sp. z o.o.
7. Welenc P. *Wymagania wobec księgowości prowadzonej za pomocą systemów informatycznych*, [w:] *Materiały z Polskiej Konferencji Audytorów PolKA 200, Audyt: standardy, planowanie, realizacja*, Bielsko-Biała 2002
8. Information Systems Control Journal 1/2002
9. „Rachunkowość” 1-12/2001, 1-2/2002
10. Międzynarodowe Standardy Rewizji Finansowej (MSRF) IFAC 1996
11. Międzynarodowe Stowarzyszenie Audytorów Systemów Informatycznych – <http://www.isaca.org>

mgr Piotr Welenc  
Zaoczne Studium Doktoranckie  
Instytut Badań Systemowych PAN, Warszawa,  
ul. Newelska 6, tel. 603 692276  
[p.welenc@ibspan.waw.pl](mailto:p.welenc@ibspan.waw.pl)





# PRZYROSTOWA BUDOWA APLIKACJI: ANALIZA BIZNESOWA I SYSTEMOWA

Marzena WINIARSKA

**Streszczenie:** Artykuł jest pierwszą z dwóch części opisu metody wytwarzania aplikacji internetowych opracowanej i wdrożonej w Dziale Projektów Internetowych firmy PROKOM. W artykule zaprezentowano początkowy etap procesu wytwórczego, jakim jest analiza biznesowa i systemowa. Prezentowana metoda prowadzenia analizy, jak i cały proces wytwórczy, którego jest elementem, została zastosowana między innymi w trakcie budowy **Internetowego systemu zarządzania SZ@FiF**, udostępnianego w modelu ASP. Metoda ta powstała na bazie metodologii „The Select Perspective” i jest praktyczną implementacją zaproponowanej przez firmę Microsoft ramy pracy MSF (Microsoft Solution Framework). Rezultatem opisanego w artykule pierwszego etapu procesu wytwórczego jest specyfikacja wymagań wobec systemu informatycznego, opracowana w formie modelu przypadków użycia utworzonego przez modelowanie i analizę typowych procesów biznesowych realizowanych w przedsiębiorstwie. Ponieważ usługi ASP są ciągle nowością na polskim rynku, w pierwszej części artykułu zaprezentowano ideę usługi wynajmu aplikacji, a także koncepcję samego systemu udostępnianego w modelu ASP.

## 1. Opis systemu

**SZ@FiF** jest systemem nowej generacji, łączącym w sobie nowatorstwo zastosowanych rozwiązań z nowym sposobem udostępniania aplikacji. Sposób ten, określany modelem ASP (ang. *Application Service Provider*), polega na wynajmowaniu oprogramowania za pośrednictwem Internetu.

System **SZ@FiF** adresowany jest do firm małych i średniej wielkości, dla których informatyka jest narzędziem wspomagającym procesy zarządzania. Pojawienie się na rynku usług ASP umożliwia takim firmom korzystanie z nowoczesnych systemów informatycznych, a jednocześnie pozwala im uniknąć dużych nakładów związanych z kupnem i uruchomieniem własnego oprogramowania na własnym sprzęcie oraz jego utrzymaniem. Korzystanie z usług ASP pozwala tym firmom szybciej reagować na zmiany rynkowe, dzięki czemu mogą one skuteczniej konkurować z większymi przedsiębiorstwami.

**SZ@FiF** jest rozwiązaniem standardowym, obsługującym podstawowe procesy biznesowe przedsiębiorstwa. W toku przeprowadzonych prac analitycznych został opracowany model biznesowy przedsiębiorstwa, na podstawie którego wyodrębniono trzy obszary funkcjonalne systemu:

- Finanse i rachunkowość
- Logistyka
- Zarządzanie zasobami ludzkimi.

Wyodrębnienie w systemie obszarów funkcjonalnych (podsystemów, modułów) umożliwia użytkownikowi korzystanie tylko z tych elementów systemu, które są mu niezbędne. Pomimo tego podziału zachowano pełną integrację procesów zachodzących pomiędzy poszczególnymi obszarami. Dzięki możliwości zmiany parametrów konfiguracyjnych system **SZ@fiP** może zostać przystosowany do potrzeb konkretnej branży czy odmiennego przebiegu procesu.

## 2. Cykl życia systemu

Budowa systemu informatycznego jest procesem skomplikowanym i złożonym. Bez względu na to, jaki proces tworzenia oprogramowania zostanie wybrany do realizacji konkretnego przedsięwzięcia, początkiem każdego z nich jest sformułowanie problemu, który ma zostać rozwiązany za pomocą systemu, a wynikiem – przekazanie działającego systemu do eksploatacji. Pomędzy tymi etapami występują zwykle fazy definiowania wymagań użytkownika, planowania przedsięwzięcia, projektowania, implementacji, testowania i dokumentowania systemu. Opis działań podejmowanych w każdej fazie, zbiór zaleceń i procedur, odpowiedzialność ról, kolejność faz, specyfikacja produktów będących rezultatem każdej z nich składają się na definicję procesu tworzenia oprogramowania.

Przystępując do realizacji systemu **SZ@fiP**, byliśmy świadomi, że czynniki decydujące o powodzeniu projektu systemu informatycznego, takie jak: szybki czas realizacji, niskie koszty czy pełna zgodność z wymaganiami użytkownika, zależą przede wszystkim od właściwej definicji procesu wytwórczego i wdrożenia postępowania zgodnego z tą definicją. Z drugiej zaś strony byliśmy świadomi mnogości modeli cyklu życia projektów wypracowanych w ostatnich latach i jednocześnie pewni, że nie znajdziemy takiego modelu, który będzie całkowicie przystawał do wymagań stawianych wobec naszego projektu. Decyzją było stopniowe tworzenie własnego procesu produkcji oprogramowania na bazie istniejących modeli, z założeniem traktowania tych opublikowanych bardziej jako zaleceń i rad niż standardów do bezwzględnego przestrzegania. Opieraliśmy się na cyklu życia projektu zaproponowanym w ramach metodologii „The Select Perspective” oraz na wskazówkach i procedurach zaproponowanych przez Microsoft w ramie pracy MSF (Microsoft Solution Framework).

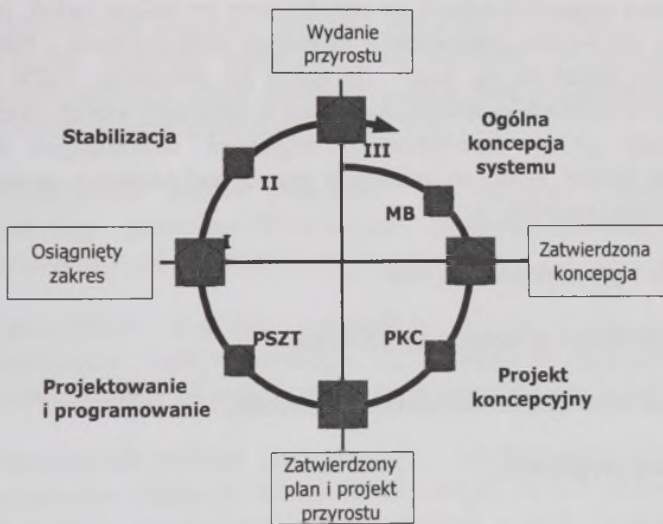
Podstawowymi cechami przyjętego przez nas procesu wytwarzania oprogramowania są:

- iteracyjny i przyrostowy charakter procesu,
- cykl życia systemu oparty na kamieniach milowych,

- zdefiniowanie ról i odpowiedzialności w zespole skorelowanych z przebiegiem procesu wytwórczego.

Iteracyjny i przyrostowy charakter procesu oznacza, że oprogramowanie nie jest dostarczane w całości w wersji ostatecznej na koniec projektu, ale w zamkniętych fragmentach – przyrostach. **Przyrost** stanowi pewną kompletną pod względem funkcjonalności część systemu, która może być dostarczona użytkownikowi jako samodzielna, gotowa do użycia całość. Dla każdego przyrostu realizowany jest pełny, pojedynczy cykl wytwórczy. Zachowanie spójności między przyrostami oraz wykorzystywanie wyników i doświadczeń z poprzednich etapów pracy zachodzi automatycznie, gdyż operuje się ciągle na tym samym modelu, do którego dodawane są szczegóły specyficzne dla danego przyrostu. W różnicy między przyrostami zawarte są zarówno nowe funkcje, jak i efekty korekty błędów wykrytych w produkcie finalnym poprzedniego przyrostu.

W przyjętym przez nas cyklu życia systemu wyróżnione zostały 4 fazy, z których każda zakończona jest kamieniem milowym.



Rys. 1. Cykl życia systemu

**Kamienie milowe** to punkty na krzywej rozwoju systemu. Osiągnięcie kamienia milowego oznacza, że zostały wykonane wszystkie prace zaplanowane w danej fazie rozwoju systemu i nadszedł czas na dokonanie przeglądu, któremu podlegają stworzone dokumenty projektowe oraz gotowa aplikacja (w zależności od rodzaju osiągniętego kamienia milowego). Przegląd pozwala na ocenę postępu prac, a także stwarza możliwość modyfikacji zakresu projektu w razie zmiany wymagań wobec systemu czy w wyniku reakcji na zidentyfikowane ryzyka. Kamień milowy reprezentuje osiągnięcia zespołu i zgodę na kontynuację prac.



Przy wytwarzaniu systemu **SZ@FIP** wyróżnione zostały następujące kamienie milowe:

- Zatwierdzona koncepcja – kończy prace nad ogólną koncepcją systemu - modelem biznesowym.
- Zatwierdzony plan i projekt koncepcyjny – kończy prace nad projektem koncepcyjnym oraz planowanie i harmonogramowanie prac dotyczących wybranego zakresu przyrostu.
- Osiągnięty zakres aplikacji – kończy prace projektowe i implementacyjne.
- Wydanie – kończy okres stabilizacji wydania.

Sprawne zarządzanie procesem wytwarzania systemu wymaga nie tylko zdefiniowania faz i etapów procesu, ale także klarownego określenia zadań i zakresu odpowiedzialności członków zespołu. Model zespołu zbudowany został na podstawie modelu proponowanego przez ramę pracy MSF. Składa się on z 6 równorzędnych ról, które ponoszą pełną odpowiedzialność za przydzielony im zakres zadań, przy czym odpowiedzialność nie wygasa całkowicie w momencie przechodzenia z fazy do fazy cyklu, co najwyżej zwiększa się bądź zmniejsza jej znaczenie. Takie podejście gwarantuje, że każdy z członków zespołu ma wkład w ostateczny kształt produktu oraz że zachodzi ciągły proces monitorowania zagadnień wchodzących w zakres odpowiedzialności każdej z ról, co zmniejsza prawdopodobieństwa ujawnienia się problemów pod koniec projektu.

W zespole reprezentowane są role:

- Odpowiedzialność za oczekiwania Klienta,
- Odpowiedzialność za oczekiwania Użytkownika,
- Zarządzanie programem,
- Implementacja,
- Testowanie,
- Logistyka,

przy czym analitycy ponoszą odpowiedzialność za oczekiwania klienta, użytkownika oraz testowanie.

### 3. Analiza biznesowa i systemowa

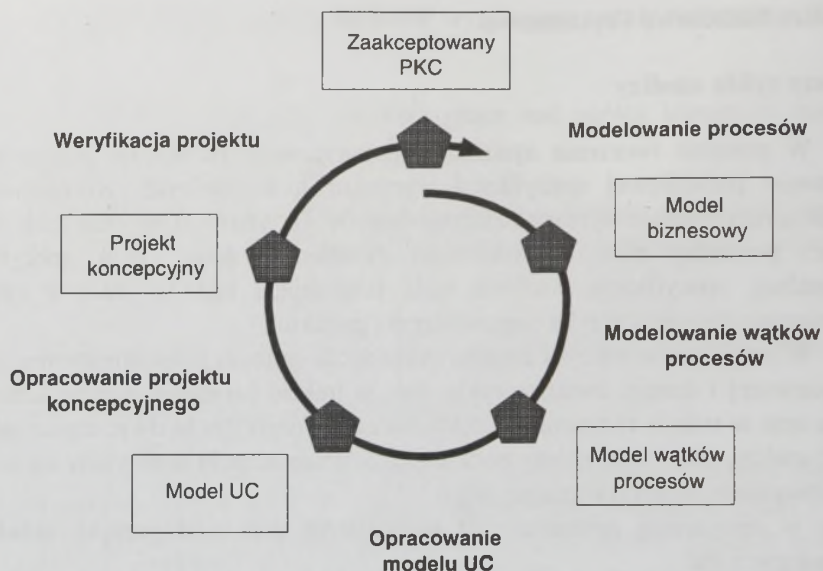
#### 3. 1. Fazy cyklu analizy

W procesie tworzenia systemu informatycznego szczególne znaczenie ma opracowanie prawidłowej specyfikacji wymagań funkcjonalnych. Konieczne jest dokładne sprecyzowanie wymagań zleceniodawców i użytkowników oraz zachowanie spójności pomiędzy nimi a tworzonym modelem systemu. Brak specyfikacji funkcjonalnej, specyfikacja wadliwa bądź posiadająca luki to jeden z częściowo wymienianych obszarów ryzyka niepowodzenia projektu.

W przyjętym w naszym zespole cyklu życia systemu prace analityczne lokują się w pierwszej i drugiej ćwiartce cyklu, tzn. w trakcie tworzenia ogólnej koncepcji systemu oraz w trakcie tworzenia projektu koncepcyjnego. Na te dwie części nakłada się cykl analizy, przy czym należy podkreślić, że prace zespołu analityków nie kończą się po utworzeniu projektu koncepcyjnego.

Przyjęty w opisywanej metodzie cykl prowadzenia prac analitycznych składa się z następujących faz:

- Opracowanie diagramu hierarchii procesów – efektem tej fazy analizy jest diagram hierarchii procesów-model biznesowy – ilustrujący i opisujący zidentyfikowane grupy procesów biznesowych przebiegających w typowych obszarach działalności przedsiębiorstwa.
- Modelowanie wątków procesów – dla wybranej z modelu biznesowego podgrupy procesów biznesowych następuje uszczegółowienie opisów przebiegów pojedynczych wątków procesów.
- Opracowanie diagramu przypadków użycia – na podstawie analizy przebiegów wątków procesów definiowane są wymagania funkcjonalne wobec systemu informatycznego – powstają diagramy przypadków użycia.
- Opracowanie projektu koncepcyjnego – rezultaty poprzednich faz w postaci wybranego fragmentu modelu biznesowego, modeli wątków procesów oraz diagramów przypadków użycia włączone zostają do dokumentu projektu koncepcyjnego.
- Weryfikacja projektu – dokonywana zarówno wewnątrz zespołu analizy, jak i na zewnątrz zespołu przez Zespół Projektowo-Implementacyjny i Zespół Testów Zewnętrznych.



Rys. 2. Cykl życia analizy

### 3. 2. Modelowanie procesów biznesowych podstawą analizy wymagań

Do niedawna metody prowadzenia analizy wymagań wobec systemu oparte były na analizie struktury funkcjonalnej przedsiębiorstwa. W praktyce wyglądało to tak, że zespół analityków przeprowadzał analizę działalności w konkretnym, wyodrębnionym funkcjonalnie dziale przedsiębiorstwa. Działania takie prowadziły do powstania systemów informatycznych doskonałych z punktu widzenia realizacji potrzeb konkretnego działu, niedoskonałych natomiast z punktu widzenia przedsiębiorstwa jako całości. Systemy takie funkcjonowały jako zamknięte całości i jakkolwiek często zestaw tego typu systemów eksploatowanych w przedsiębiorstwie nazywany był szumnie zintegrowanym systemem zarządzania, to jednak z integracją działań pomiędzy poszczególnymi działami przedsiębiorstwa nie było najlepiej. Systemy takie nie zapewniały przepływu informacji pomiędzy działami, nie dostarczały mechanizmów wspierających obieg informacji, nie zapewniały integracji zarówno na poziomie zasobów, jak i przepływów. Prowadziło to do powtarzania tych samych działań w różnych systemach, co prowadziło do podwójnej rejestracji, opóźnień w przepływie danych, redundancji danych oraz często braku ich spójności. Tak eksploatowane systemy informatyczne podnosiły wydajność pracy w ramach poszczególnych działów, natomiast utrwały wady typowe dla struktury funkcjonalnej firmy, takie jak powstawanie mechanizmów izolujących działy i pionów,



niekontrolowane opóźnienia czasowe występujące pomiędzy działaniami, wielokrotna realizacja tych samych zadań, brak odpowiedzialności za jakość obsługi klienta.

Wraz ze wzrostem konkurencji w przedsiębiorstwach coraz silniej uwidaczniają się słabe strony struktur funkcjonalnych. Obecnie w zarządzaniu organizacjami obserwuje się tendencje, w których większy nacisk kładzie się na znaczenie i wartość procesów biznesowych. Podział przedsiębiorstwa dokonywany przez pryzmat procesów biznesowych jest obecnie zjawiskiem powszechnym. Pojawiły się także nowe wymagania w stosunku do systemów informatycznych – powinny one dostarczać mechanizmów wspierających poziomą koordynację działań w ramach firmy. Wymusiło to zmianę podejścia do metody prowadzenia analizy. Analiza wymagań funkcjonalnych wobec systemu oparta jest na modelowaniu procesów biznesowych, a jej efektem jest nie tylko specyfikacja zadań systemu w poszczególnych ogniwach przedsiębiorstwa, ale także takie zaprojektowanie systemu, aby poprawiał jakość powiązań pomiędzy poszczególnymi ogniwami.

Nowe podejście do zagadnienia analizy wymagań funkcjonalnych wobec systemu znalazło odbicie w wyborze sposobu prowadzenia analizy podczas projektowania systemu **SZ@FiP**. Zrezygnowano z analizy struktury funkcjonalnej firmy, a skoncentrowano się na modelowaniu procesów występujących wewnątrz firmy, w szczególności procesów dostarczających wartości klientowi.

Wyniki przeprowadzonej analizy zapisywane są w postaci odpowiednich modeli. Tworzenie ich pozwala z jednej strony na precyzyjne określanie zakresu i obowiązków systemów biznesowego i informatycznego, z drugiej zaś na prowadzenie uzgodnień specyfikacji wymagań wobec systemu. Stosowanie formalnie zdefiniowanej metodyki modelowania ma ogromne znaczenie dla przebiegu wytwarzania systemu. Używanie wspólnego języka opisu z dobrze poznaną notacją i semantyką oraz większa precyzyjność języka formalnego w stosunku do języka naturalnego pozwalają uniknąć wieloznaczności specyfikacji. Stosowanie modeli pozwala na dokumentowanie zmian zarówno specyfikacji, jak i projektów. Uzyskiwany efekt to pełna dokumentacja wszystkich zmian oraz możliwość przeglądu rozwoju systemu przez kolejne wersje modeli, a także możliwość precyzyjnego planowania rozwoju systemu na poziomie wymagań biznesowych.

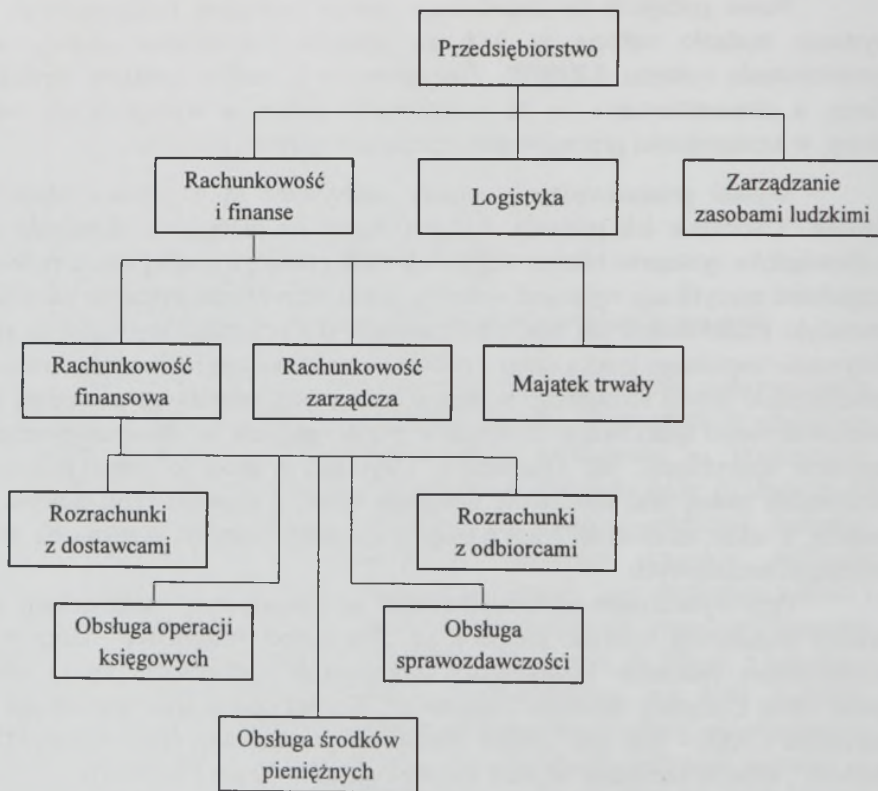
Przy wytwarzaniu systemu **SZ@FiP** jako praktyczną implementację metody analizy biznesowej wybrano metodologię „The Select Perspective”, która w części modelowania procesów biznesowych wykorzystuje metodologię Lynx, stworzoną przez firmę Computer Sciences Corporation. Konsekwencją tego wyboru jest wybór narzędzia CASE - jest nim „Select Enterprise” dostarczany przez firmę „Princeton Softech”, które to narzędzie wspiera metodologię „The Select Perspective”.

Modelowanie procesów zachodzących w przedsiębiorstwie jest przedsięwzięciem skomplikowanym i zwykle jeden rodzaj modelu nie wystarcza do pokazania pełnego obrazu tych procesów. Model powinien obrazować z jednej strony statyczną strukturę wewnętrzną organizacji, zaś z drugiej - aktywne zachowanie się przedsiębiorstwa w czasie oraz reakcje na zdarzenia wewnętrzne i zewnętrzne.

W metodologii Lynx do tworzenia modeli statycznych i dynamicznych procesów biznesowych wykorzystuje się dwa typy diagramów:

- diagramy hierarchiczne procesów,
- diagramy wątków procesów.

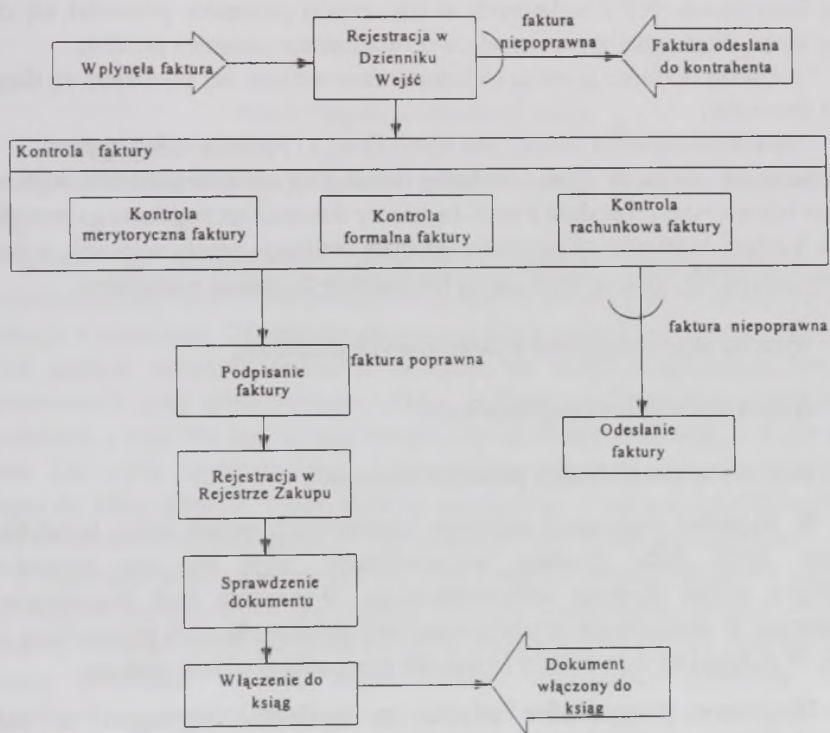
**Diagramy hierarchiczne procesów** przedstawiają procesy biznesowe w postaci drzewa hierarchii procesów. Obrazuje ono w formie graficznej statyczne powiązania zachodzące pomiędzy procesami oraz ich dekompozycję na różnych poziomach hierarchii. Liczba poziomów podziału procesów biznesowych w ramach diagramu jest nieokreślona - zależy ona od stopnia skomplikowania modelowanego obszaru.



Rys. 3. Diagram hierarchii procesów

Nadrzędnym elementem w diagramie hierarchii procesów jest modelowany obszar – całe przedsiębiorstwo lub jego wyodrębniona część. Dekompozycję procesów zwykle kończy się na zaprezentowaniu pojedynczych elementarnych procesów biznesowych w ramach grup biznesowych, jakkolwiek możliwe jest dokonanie dalszego podziału przez wyodrębnienie, w ramach elementarnych procesów biznesowych, kroków biznesowych.

W celu pokazania dynamiki procesów wykorzystuje się diagramy wątków procesów. Pojedynczymi ogniwami są tutaj elementarne procesy biznesowe. Diagramy wątków procesów prezentują ciąg elementarnych procesów biznesowych wykonywanych jako reakcja na określone zdarzenie zewnętrzne oraz wartość otrzymywaną jako rezultat końcowy procesu. Wartość otrzymana jako rezultat jednego procesu może stanowić zdarzenie inicjujące przebieg innego procesu biznesowego.



Rys. 4. Diagram wątku procesu



W pracach analitycznych prowadzonych zgodnie z opisaną metodą wykorzystujemy oba rodzaje diagramów. Jako pierwszy powstaje model biznesowy będący diagramem hierarchicznym procesów. Nadrzędnym elementem jest modelowane przedsiębiorstwo, a poniżej wyodrębniono trzy kluczowe obszary działalności - Rachunkowość i finanse, Logistykę oraz Zarządzanie zasobami ludzkimi. Liśćmi na diagramie drzewa hierarchii procesów są wątki procesów.

Wyniki tej fazy analizy zawarte są w dokumencie „Model biznesowy”. Oprócz diagramu hierarchii procesów w dokumencie tym opisane są, dla każdego obszaru działalności, cele i sposób działania w ramach grup biznesowych, a także przedstawione są strumienie danych wymienianych pomiędzy modelowanym obszarem a elementami otoczenia. Dokładne określenie celów, zakresów (granic) i elementów (z opisem na wysokim poziomie ogólności) składających się na kluczowy obszar działalności umożliwia odtworzenie pełnego obrazu przedsiębiorstwa (w zakresie procesów), wybranie procesów pierwszoplanowych lub niezbędnych. Na podstawie opisu zidentyfikowanych i wybranych w ten sposób procesów prowadzi się dalszą, bardziej szczegółową analizę, zmierzającą do dokładnego opisania procesu.

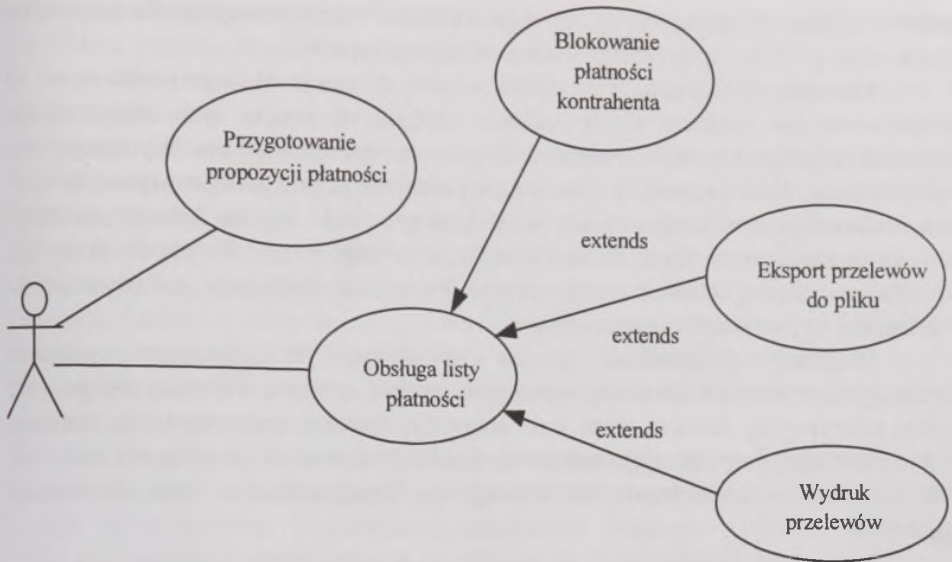
Dokładne opisanie procesu to kolejna faza analizy. Jej wynikiem są diagramy wątków procesów.

Następnym krokiem analizy jest identyfikacja i opisanie usług systemu informatycznego. Punktem wyjścia do takiej definicji są stworzone modele wątków procesów biznesowych. Zgodnie z metodą, należy dokonać szczegółowego przeglądu modeli i wskazać te fragmenty procesów, których realizację należy wspomóc systemem informatycznym. Kryteria wyboru takich fragmentów to przede wszystkim:

- wyraźna poprawa jakości wykonywanych procesów,
- skrócenie czasu realizacji procesów,
- możliwość automatyzacji pewnych czynności.

W wypadku większości procesów biznesowych można łatwo identyfikować fragmenty, które będą poddane informatyzacji. Stają się one wymaganiami użytkownika wobec systemu informatycznego. Wskazanie tych fragmentów jest równoznaczne ze sformułowaniem wymagań funkcjonalnych, czyli przypadków użycia systemu. Wynikami tej fazy analizy są modele przypadków użycia systemu.

**Diagramy przypadków użycia** są modelami wymagań w zakresie funkcjonalności systemu. Pokazują one powiązania pomiędzy aktorami a przypadkami użycia oraz pomiędzy różnymi przypadkami użycia.



Rys.5. Diagram przypadków użycia

**Modelowanie** należy rozpocząć od identyfikacji aktorów projektowanego systemu.

**Aktor** to abstrakcyjny użytkownik systemu, reprezentujący grupę rzeczywistych użytkowników lub partnerów systemu o podobnych funkcjach i sposobie komunikacji z systemem. Określenie aktorów systemu umożliwia zawężenie obszaru, który ma znaleźć odzwierciedlenie w systemie, do usług świadczonych na rzecz zainteresowanych grup użytkowników. Aktor to klasa użytkowników o określonych uprawnieniach, a więc nie jest on konkretnym użytkownikiem, ale rolą, w którą wciela się jeden lub wielu użytkowników. Jeden użytkownik może również pełnić role przypisane do kilku aktorów. Przez aktorów należy więc rozumieć wszystkie aktywne elementy środowiska informatycznego używające go i komunikujące się z nim.

**Przypadek użycia systemu** jest to ciąg interakcji między aktorem a systemem oraz transakcji (niepodzielnych operacji) w systemie, dostarczający aktorowi rezultatu o mierzalnej wartości. Opis przypadku użycia musi zawierać przede wszystkim opis scenariuszy przebiegu – czyli ciągu kroków opisujących interakcję między użytkownikiem a systemem. Scenariuszy może być kilka, ponieważ często – oprócz przebiegu podstawowego, będącego podstawowym sposobem przeprowadzenia przypadku użycia – dołącza się opisy przebiegów alternatywnych, obejmujących rzadziej występujące warianty przypadku użycia czy też sytuacje spowodowane przez błędy. Jest też wiele innych elementów, które można dodać do opisu przypadku użycia. W stosowanym w opisanym szablonie projektu koncepcyjnego do opisu przypadku użycia można dołączać także opis przeznaczenia przypadku użycia oraz warunki wejściowe, czyli opis tego, co ma być spełnione zanim rozpocznie się przypadek użycia. W przypadkach bardziej skomplikowanych pod względem ilości i

sposobu obsługi ekranów zaleca się wprowadzanie mapy nawigacji dla przypadku użycia, czyli graficznego przedstawienia przebiegu przypadku.

Elementem diagramu przypadków użycia są relacje. Relacje podstawowe to relacje pomiędzy aktorami a przypadkami użycia. Ale oprócz tych związków na diagramie pokazuje się często relacje pomiędzy przypadkami użycia. Występują dwa rodzaje relacji. Relacja „uses” pozwala na wyodrębnienie przypadków użycia, których funkcjonalność jest wykorzystywana przez inne przypadki użycia. Relacja „extends” pozwala na zaznaczenie faktu, że funkcjonalność jednego przypadku użycia może być rozszerzona o funkcjonalność innego przypadku użycia. Relacja ta jest używana do przedstawiania przebiegów opcjonalnych.

Diagramy przypadków użycia są kluczowym elementem projektu koncepcyjnego, bowiem ilustrują wymagania wobec systemu informatycznego. Ale projekt koncepcyjny zawiera także inne elementy. Podczas opracowywania szablonu projektu koncepcyjnego przyjęto założenie, że układ i zawartość projektu ma opisywać przebieg procesu dochodzenia do wymaganej funkcjonalności. Stąd elementami projektu są:

- Opis działalności biznesowej dla wybranej z modelu hierarchii procesów grupy biznesowej. W opisie tym zawarte są także podstawowe cele biznesowe, generalne wymagania funkcjonalne i pozafunkcjonalne.
- Model wstępnej hierarchii procesów, którego zadaniem jest przedstawienie miejsca omawianej podgrupy biznesowej w ogólnym modelu biznesowym.
- Sposób mapowania modelu biznesowego na systemy informatyczne, czyli odwzorowanie grupy wątków procesów w ramach analizowanej podgrupy biznesowej na moduły systemu informatycznego.
- Opisy modułów, które oprócz ogólnych założeń projektowych, diagramu przypadków użycia, zawierają także przebiegi wątków procesów, których analiza doprowadziła do określenia i przedstawienia wymagań funkcjonalnych wobec systemu w postaci diagramu przypadków użycia.

Opracowany i zweryfikowany projekt koncepcyjny zostaje przekazany do Zespołu Projektowo-Implementacyjnego, gdzie stanowi podstawę do planowania i, w dalszej kolejności, do podjęcia prac projektowych.

#### **4. Czynniki warunkujące powodzenie**

Na zakończenie artykułu chciałabym wymienić dwa czynniki, które, moim zdaniem, mają ogromny wpływ na jakość produktu końcowego, którym jest



wytwarzane oprogramowanie. Pierwszy z nich to ściśle powiązanie wyników analizy z modelem systemu, drugi to przyjęty sposób organizacji pracy.

Stosowana przez nas metoda zakłada, że modelowanie usług systemu jest podporządkowane wynikom analizy biznesowej i koncentruje się na wspieraniu przebiegu procesu biznesowego przy użyciu technologii informatycznej. Przy takim podejściu tworzy się modele systemu po to, aby stanowiły one pomost pomiędzy procesem, który ma być wspomagany, a tworzonym w tym celu oprogramowaniem. W przeciwieństwie do metod strukturalnych, powstające modele bardzo dobrze odzwierciedlają modele analityczne i są jednocześnie niezbędne w fazie implementacji. Ponadto modele, w których biznes jest ściśle powiązany ze specyfikacją funkcjonalną, niezbędne są w sytuacji, gdy konieczne staje się wprowadzenie zmian w systemie. Pozwalają one szybko zidentyfikować zakres systemu, w którym należy wprowadzić zmiany spowodowane zmianami w biznesie.

Drugim czynnikiem to przyjęty sposób organizacji pracy, który zakłada stałą, personalną reprezentację różnych punktów widzenia na wytwarzany system w całym cyklu życia systemu. Przykładowo, osiągnięcie drugiego kamienia milowego na krzywej rozwoju systemu oznacza co prawda, że został opracowany i zweryfikowany projekt koncepcyjny dla danego przyrostu, ale nie oznacza wcale, że wraz z przekazaniem projektu do zespołu projektowo–implementacyjnego kończy się udział zespołu analizy w realizacji danego zakresu funkcjonalnego. Ideą przyjętego sposobu organizacji pracy zespołu jest powołanie ról – ośrodków odpowiedzialności za wybrany aspekt systemu i przyjęcie założenia, że jakkolwiek odpowiedzialność za dany aspekt jest największa w czasie trwania jednej fazy, to nie wygasa w momencie wydania produktu tej fazy. W praktyce oznacza to, że członkowie zespołu analizy w trakcie całego cyklu życia systemu koncentrują się na przypisanych im zagadnieniach, takich jak:

- Właściwy zakres funkcjonalny systemu.
- Zgodność z oczekiwaniami użytkowników.
- Poprawność merytoryczna przyjętego rozwiązania.
- Zapewnienie odpowiedniego poziomu jakości oprogramowania.
- Zapewnienie komfortu pracy z systemem.
- Łatwość wdrożenia systemu.

Jest sprawą oczywistą, że aby wykształcić poczucie odpowiedzialności za dany aspekt wytwarzania systemu i tym samym zabezpieczyć projekt przed ryzykiem powstania zagrożeń w różnych obszarach, należy zaplanować i uwzględnić w harmonogramie konkretne, związane z tym zadania dla członków zespołu. Częstym błędem w trakcie budowy harmonogramu jest przydział większości czasu przede wszystkim na realizację

produktu podstawowej fazy – w przypadku zespołu analizy na tworzenie modelu biznesowego i projektu koncepcyjnego. Z mego doświadczenia wynika, że analitycy prawie w takim samym stopniu zaangażowani są w kolejnych fazach cyklu wytwórczego systemu jak w „przynależnej” im fazie. Najważniejsze zadania realizowane przez Zespół Analizy w dalszych fazach wytwarzania oprogramowania to uzgodnienia przyjętych przez projektantów rozwiązań konstrukcyjnych, weryfikacja prototypów pod względem zgodności z projektem koncepcyjnym i standardem interfejsu, opracowanie dokumentacji użytkownika oraz wykonywanie testów systemu.

Oba wymienione czynniki podkreślają związek analityków i efektów ich prac z produktami dalszych faz cyklu życia systemu. Zachowanie tego związku gwarantuje, że powstające oprogramowanie w pełni odzwierciedla wymagania biznesu, a tym samym w dużym stopniu decyduje o powodzeniu projektu informatycznego.

## Literatura

- 2.1. Martin Fowler, Kendall Scott: UML w kropelce, Oficyna Wydawnicza LTP, 2002,
- 3.2. Coad P., Yourdon E.: Analiza obiektowa, Oficyna Wydawnicza READ ME, Warszawa 1994.
- 4.3. Coad P., Yourdon E.: Projektowanie obiektowe, Oficyna Wydawnicza READ ME, Warszawa 1994.
- 5.4. Dumnicki R., Kasprzyk A., Kozłowski M., Analiza i projektowanie obiektowe, Wydawnictwo Helion, Gliwice 1998.

Mgr Marzena WINIARSKA  
-Prokom Software S.A. Gdynia,  
ul. Śląska 23/25 Tel.:(0-58) 628-69-63  
e-mail: [winiarskam@prokom.pl](mailto:winiarskam@prokom.pl)

# PRZYROSTOWA BUDOWA APLIKACJI: PROJEKTOWANIE I IMPLEMENTACJA

Robert ZYSKOWSKI

**Streszczenie:** Celem artykułu jest prezentacja metod i technik stosowanych w trakcie prac projektowo–implementacyjnych nad Internetowym systemem zarządzania **SZ@FiP** firmy Prokom Software SA. Przedstawiony jest w nim sposób postępowania prowadzący do wyprodukowania aplikacji internetowej na podstawie specyfikacji funkcjonalnej, opracowanej w formie modelu przypadków użycia. Model ten powstał w wyniku modelowania i analizy typowych procesów biznesowych realizowanych w przedsiębiorstwie. W artykule omówione zostały produkty powstające w wyniku kolejnych etapów projektowania i implementacji wraz z opisem ich przeznaczenia, zawartości oraz sposobu wytworzenia. W części poświęconej projektowaniu zaprezentowane zostały techniki modelowania statycznej i dynamicznej części systemu stosowane przy wytwarzaniu systemu **SZ@FiP**, oparte na diagramach oferowanych w ramach notacji UML (Unified Modeling Language). W części dotyczącej implementacji przedstawione zostały praktyczne korzyści wynikające z podziału aplikacji na warstwy: biznesową, dostępu do danych i bazy danych. Zaprezentowano także fragmenty kodu charakterystyczne dla poszczególnych warstw aplikacji.

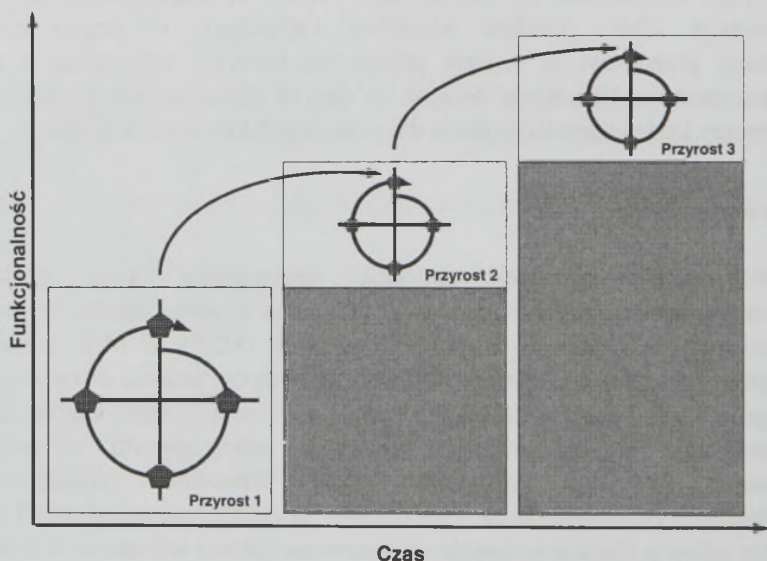
## 1. Proces wytwórczy

W opisaney poniżej metodzie prowadzenia prac projektowo–implementacyjnych konstrukcję procesu wytwarzania systemu oparto na propozycji procesu zawartej w Microsoft Solution Framework (MSF). Z MSF zostały m.in. zaczerpnięte, a następnie dookreślone definicje faz i etapów procesu wytwórczego oraz definicje produktów poszczególnych faz i etapów. Ponieważ MSF kładzie nacisk na ścisłe powiązanie między procesem a rolami występującymi w zespole, w prezentowanej metodzie występuje jawne wskazanie rozłączności ról odpowiedzialnych odpowiednio za wytworzenie produktów poszczególnych faz i ich odbiór. Przy takim podejściu szczególnego znaczenia nabiera wdrożenie techniki pracy polegającej na rozdzieleniu odpowiedzialności za specyfikowanie tego, co ma być zrobione, od odpowiedzialności za to, w jaki sposób zostanie to zrealizowane. Dwoistość: *specyfikacja – realizacja* przewija się przez całość procesu wytwarzania, zaczynając od modelu procesów biznesowych, a na implementacji metod i struktur danych kończąc. Zgodnie z tą zasadą, model biznesowy jest specyfikacją dla modelu przypadków użycia, który stanowi jego realizację. Model przypadków użycia jest z kolei specyfikacją dla modelu klas i obiektów, które stanowią jego realizację itd. Zasada dwoistości znajduje również swoje odzwierciedlenie w organizacji wszystkich zadań projektowo–implementacyjnych, dla których specyfikacją jest harmonogram powiązany ze specyfikacją funkcjonalną, a realizację stanowią gotowe projekty



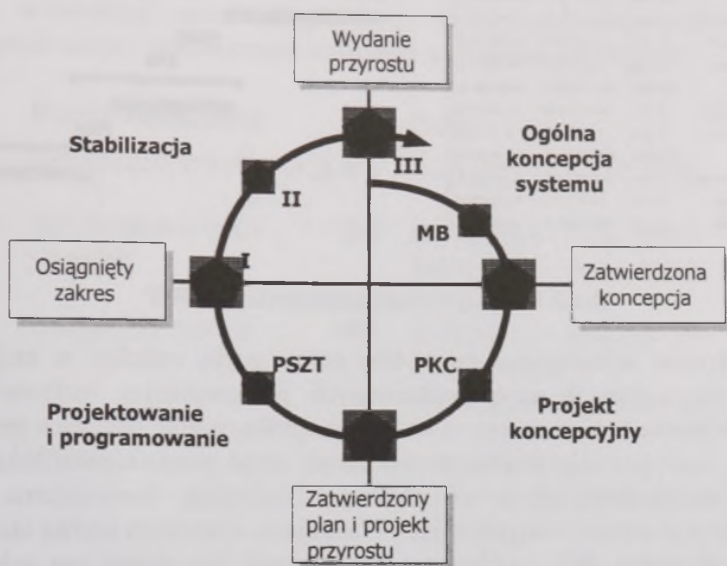
szczegółowe oraz działający kod. Aby tak zdefiniowana zasada organizacji procesu wytwórczego mogła być zastosowana w praktyce i przynosiła wymierne korzyści w postaci produktów dostarczanych na czas i zgodnych ze specyfikacją funkcjonalną, wymagane jest z jednej strony każdorazowe uzgadnianie specyfikacji i uzyskiwanie potwierdzenia dobrego jej zrozumienia, z drugiej zaś informacja zwrotna, czy realizacja faktycznie spełnia wymagania specyfikacji przy założonym poziomie akceptacji.

Intuicyjnie wydaje się, że zapewnienie prawidłowego przebiegu procesu zgodnie z podaną definicją wymaga zastosowania rozbudowanej kontroli formalnej, która znacznie zwiększa pracochłonność. Efekt ten faktycznie występuje, ale tylko w początkowej fazie wytwarzania systemu. Wraz z rozwojem systemu formalna kontrola stopniowo traci znaczenie. Powtarzane w kolejnych fazach i etapach czynności wytwórcze zaczynają mieć charakter rutynowy. Stosowanie zasady *specyfikacja – realizacja* staje się elementem samodyscypliny zespołu wytwórczego. Efekt autokorekty procesu wytwórczego jest wzmacniany przez stosowanie iteracyjnego, addytywnego cyklu życia systemu, zgodnie z którym system wytwarzany jest w kolejnych, kompletnych przyrostach funkcjonalności.



Rys.1. Przyrostowy cykl życia systemu

Celem pojedynczej iteracji jest zbudowanie modelu i zaimplementowanie kompletnego pod względem funkcjonalnym fragmentu systemu - przyrostu. Tak więc w ramach każdego przyrostu realizowany jest pełny cykl wytwórczy. Zachowanie spójności między przyrostami oraz wykorzystywanie wyników poprzedniej pracy zachodzi automatycznie, gdyż operuje się ciągle na tym samym modelu, do którego dodawane są szczegóły specyficzne dla danego przyrostu. W różnicy między przyrostami zawarte są zarówno nowe funkcje, jak i efekty korekty błędów wykrytych w produkcie finalnym poprzedniego przyrostu.



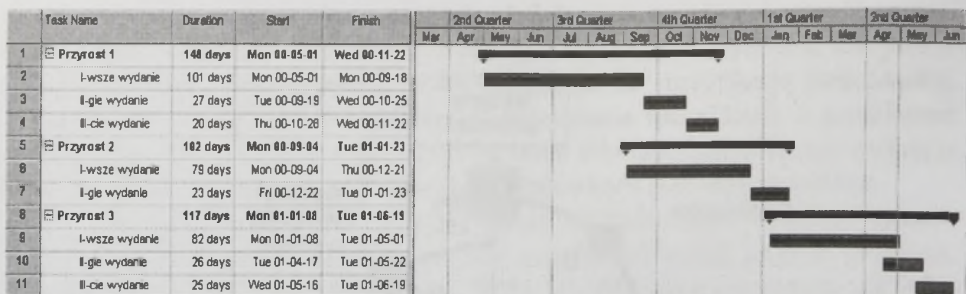
Rys.2. Przebieg pojedynczej iteracji<sup>1</sup>

Określenie zakresu przyrostu wymaga pewnej wprawy. W wypadku systemu **SZ@FiP** jest to ok. 40 przypadków użycia do zrealizowania w ciągu 4 miesięcy przez 20-osobowy zespół realizacyjny (kierownik zespołu analizy + 6 analityków, kierownik zespołu projektowo-implementacyjnego + 12 projektantów- programistów) przy korzystaniu z pomocy 5-osobowego zespołu dokumentacji zaangażowanego w ok. 50% oraz 6-osobowego zespołu testów zaangażowanego w ok. 10%.

Zdumiewająca jest zgodność proporcji czasu trwania fazy wytworzenia do czasu trwania fazy stabilizacji, wynosząca 2:1 zarówno w wypadku zastosowanego do produkcji systemu **SZ@FiP** języka programowania Visual Basic, jak i wg danych prezentowanych na kursie MSF. Dla przykładu, w wypadku stosowania C++ proporcja ta wynosi 1:2, ale osiągnięta funkcjonalność biznesowa jest dużo mniejsza w porównaniu z osiąganą z zastosowaniem języka Visual Basic, przy założeniu analogicznych zasobów realizacyjnych.

Generalnie, wytworzenie przyrostu trwa 4 miesiące, stabilizacja 1,5-2 miesiące, przy czym poszczególne przyrosty mogą się zalegać. Całość obejmuje ok. 6 miesięcy, co daje średnio wydania co 4-5 miesięcy.

<sup>1</sup> MB – model biznesowy, PKC – projekt koncepcyjny, PSZT – projekt szczegółowy techniczno-technologiczny.



Rys.3. Przebieg wytwarzania systemu **SZ@FiP**

Definicja procesu wytwórczego nie byłaby pełna, gdyby zabrakło w niej jawnego określenia ról i funkcji przyporządkowanych poszczególnym osobom/zespołom. Przyporządkowanie to daje szansę na skuteczne egzekwowanie podjętych przez zespół zobowiązań oraz pozwala zwykle na wdrożenie zasad przestrzegania zakresu praw i obowiązków wynikających z ról i funkcji. Tradycyjna, hierarchiczna struktura zespołu, z dobrze zdefiniowanymi rolami i funkcjami, umożliwia szybką identyfikację właściciela problemu, jeśli problem ten już wystąpił. Nie chroni ona jednak przed występowaniem sytuacji eskalacji każdej wątpliwości na poziom kierownika przedsiębiorstwa, co często stwarza zagrożenie powstania paraliżu decyzyjnego.

W procesie realizacji systemu **SZ@FiP** istotą modelu zarządzania jest symbioza równoważnych, choć często sprzecznych ze sobą ról odpowiedzialnych za poszczególne punkty widzenia na system. Dominacja poszczególnych ról w zespole jest konsekwentnie przenoszona wraz z następowaniem kolejnych faz i etapów pojedynczej iteracji.



Rys.4. Różne punkty widzenia na system reprezentowane w przedsięwzięciu od początku procesu wytwarzania



W strukturze skorelowanej z przebiegiem procesu wytwórczego, w której nastąpiło sprowadzenie poziomu decyzyjnego do ról odpowiedzialnych za poszczególne fazy procesu, w naturalny sposób kładzie się nacisk na podejmowanie decyzji i eliminowanie postaw nastawionych wyłącznie na szukanie problemów.

Punkt kontrolny	Rola
Zatwierdzona koncepcja ➡	Zarządzanie produktem
Zatwierdzony plan i projekt ➡	Zarządzanie produktem Zarządzanie programem
Osiągnięty zakres ➡	Zarządzanie programem Implementacja i Edukacja Użytkownika
Wydanie ➡	Implementacja Testowanie i Zarządzanie logistyką

Rys.5. Przechodzenie dominacji ról przez fazy iteracji

Podporządkowanie struktury zespołu modelowi procesu wytwórczego powoduje w efekcie powstanie i funkcjonowanie pewnego rodzaju centrów kompetencyjnych, skupiających się z jednej strony na produktach konkretnej fazy, z drugiej zaś reprezentujących istotny punkt widzenia na system od początku jego wytwarzania. Pozwala to w maksymalnym stopniu zabezpieczyć się przed ryzykiem powstania zagrożeń w różnych obszarach przez jednoznaczne określenie właścicieli ról odpowiadających poszczególnym obszarom.

## 2. Architektura i technologia

O ile MSF abstrahuje od kwestii technicznych wytwarzanego systemu, o tyle praktyka pokazuje, że im bardziej precyzyjna jest definicja procesu, tym ściślej odwołuje się ona do architektury i technologii wybranych do realizacji systemu. System **SZ@FiP**, realizowany przy użyciu opisywanej metody, został zaimplementowany w architekturze wielowarstwowej z wykorzystaniem COM+ [2] w połączeniu z Internet Information Server, w oparciu o serwery Microsoft .NET Platform [3].

**Warstwa prezentacji** to strony HTML, które obsługują całość interakcji z systemem. Za ich pośrednictwem odbywa się wprowadzanie danych, sprawdzanie ich poprawności oraz wyświetlanie żądanych informacji. Do wyświetlania stron HTML służy przeglądarka Internet Explorer w wersji 5.0 lub nowszej. Zainstalowanie przeglądarki IE 5.0 jest jedynym koniecznym warunkiem użytkowania systemu **SZ@FiP**. Strony HTML są generowane na serwerze na podstawie plików Active

Server Pages. Poniżej zaprezentowany jest przykładowy kod wygenerowanej strony

```
Public Function dajKol(sIn As String) As String
    On Error GoTo EH
    Dim xmlIn As MSXML.DOMDocument
    Dim oDB As Z_DziennikCz_DB.dziennikCzDB
    Set xmlIn = CreateObject("MSXML.DOMDocument")
    Call xmlIn.loadXML(sIn)
    Set oDB = CreateObject("Z_DziennikCz_DB.dziennikCzDB")
    sIn = oDB.dajKol(CLng(xmlIn.selectSingleNode("//@RokObrId").Text))
    dajKol = RetOK("<KolDziennikCz>" & sIn & "</KolDziennikCz>")
    Exit Function
EH:
    DisCommit
    Call Err.Raise(-1, ERR_HND, ObsBl())
End Function
```

aplikacji.

```
html>
<head>
<script src="DefDzienCz.js" defer></script>
</head>
<body onload="init()">
  <table id="Tab" style="display:none" onMouse="1" onRow="1" withMenu="1" doColSort="1">
    <thead>
      <tr>
        <td name="tbchbx"></td>
        <td name="smb" format="text" style="width: 11ex">Symbol dziennika</td>
        <td name="nazwa" format="text" style="width: 31ex">Nazwa</td>
        <td name="dataOtw" format="date" style="width: 11ex">Data otwarcia</td>
      </tr>
    </thead>
  </table>
</body>
</html>
```

Wygenerowane strony zawierają jedynie opis sposobu wyświetlania oraz sprawdzania poprawności wprowadzanych danych, natomiast nie zawierają samych danych. Dane te są ściągane z serwera po wyświetleniu strony w przeglądarce. Wymiana danych pomiędzy stronami HTML i serwerem odbywa się wyłącznie w formacie XML [4]. Poniżej pokazany jest przykład wywołania usługi systemu i wymiany danych pomiędzy stroną HTML a serwerem WWW.

```
<function init() {
  var xmlOut

  xmlOut = LIB_Serwis("Z_DziennikCz_Bus.DziennikCz", "dajKol", xmlIn)
  if(blad(xmlOut, true))
    return
  drawTab(Tab, xmlOut, "KolDziennikCz")
}
```

Językiem programowania warstwy prezentacji systemu jest Java Script oraz DHTML.

**Warstwa biznesowa** systemu powstaje jako oprogramowanie komponentowe w oparciu o specyfikację COM+ dla środowiska Microsoft Windows. Komponenty te są umieszczone w plikach .dll na serwerze. Zadaniem warstwy biznesowej jest powtórne sprawdzenie poprawności otrzymanych danych, sprawdzenie uprawnień

użytkownika, zamiana danych z formatu XML do postaci akceptowanej przez komponenty warstwy dostępu do danych, przekazanie danych do komponentów warstwy dostępu do danych, odebranie przetworzonych danych i odesłanie ich do warstwy prezentacji. Wymiana danych między warstwą prezentacyjną i biznesową następuje w formacie XML. Językiem programowania warstwy biznesowej systemu jest Visual Basic. Poniżej przedstawiony jest przykład implementacji usługi systemu w warstwie biznesowej.

Na **warstwę dostępu do danych**, podobnie jak w wypadku warstwy biznesowej, składają się komponenty umieszczone w plikach .dll na serwerze. Zadaniem tych komponentów jest przekazywanie danych pomiędzy warstwą biznesową i bazą danych.

```
Public Function dajKol(id As Long) As String
    On Error GoTo EH
    Dim oDB As ZSZ2Db.ZSZDb20
    Set oDB = CreateObject("ZSZ2Db.ZSZDb20")
    dajKol = oDB.RunSPReturnXML("Z_DziennikCz_dajKol", oDB.pInt(id))
    Exit Function
EH:
    DisCommit
    Call Err.Raise(-1, ERR_HND, ObsBl())
End Function
```

Komponenty te zajmują się wywoływaniem procedur składowanych, obsługą błędów specyficznych dla bazy danych (zamianą komunikatów o błędach bazy danych na komunikaty zrozumiałe dla użytkownika) oraz przekazywaniem przetworzonych danych do warstwy biznesowej. Poniżej zaprezentowany jest przykładowy kod implementujący usługę systemu w warstwie dostępu do danych.

**Warstwę bazy danych** stanowi motor bazy danych. Dostęp do danych umieszczonych w tablicach odbywa się wyłącznie za pośrednictwem procedur składowanych w oparciu o standard ADO. System wykorzystuje motor baz danych MS SQL Server 2000. Poniżej przedstawiony jest przykład implementacji usługi systemu realizowanej

```
Create Procedure Z_DziennikCz_dajKol(@id Integer)
As
    Select smb, nazwa, dataOtw From rf_DziennikCz As DziennikCz
    Where RokObr_id = @id
    Order By smb
    For Xml Auto
```

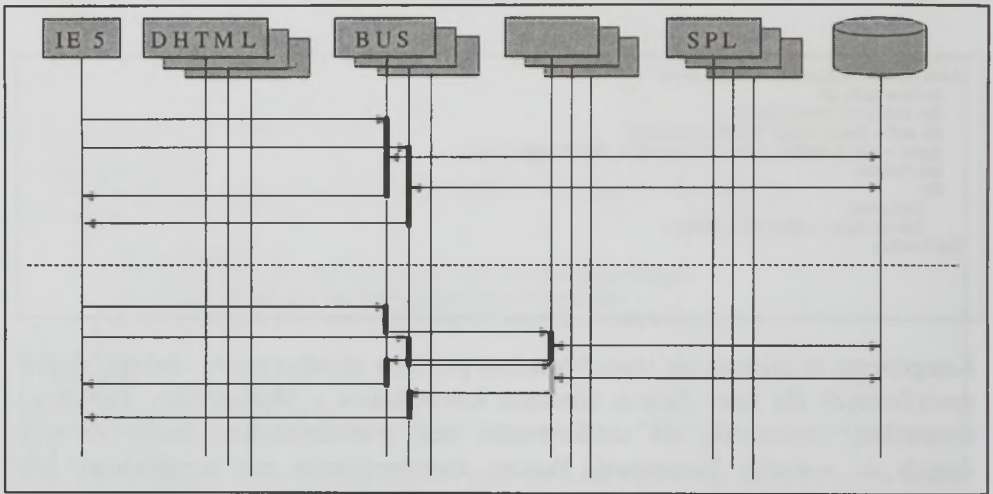
w warstwie bazy danych.

Zastosowanie wielowarstwowej architektury aplikacji daje szereg korzyści. Należą do nich m.in.:

- lepsza skalowalność aplikacji,
- lepsze wykorzystanie zasobów sprzętowych,
- możliwość rozpraszania obciążenia.



Na poniższym rysunku przedstawiono porównanie wykorzystania zasobów sprzętowych w wariancie bez dodatkowej warstwy dostępu do danych i z warstwą dodatkową.



Rys.6. Lepsze wykorzystanie zasobów

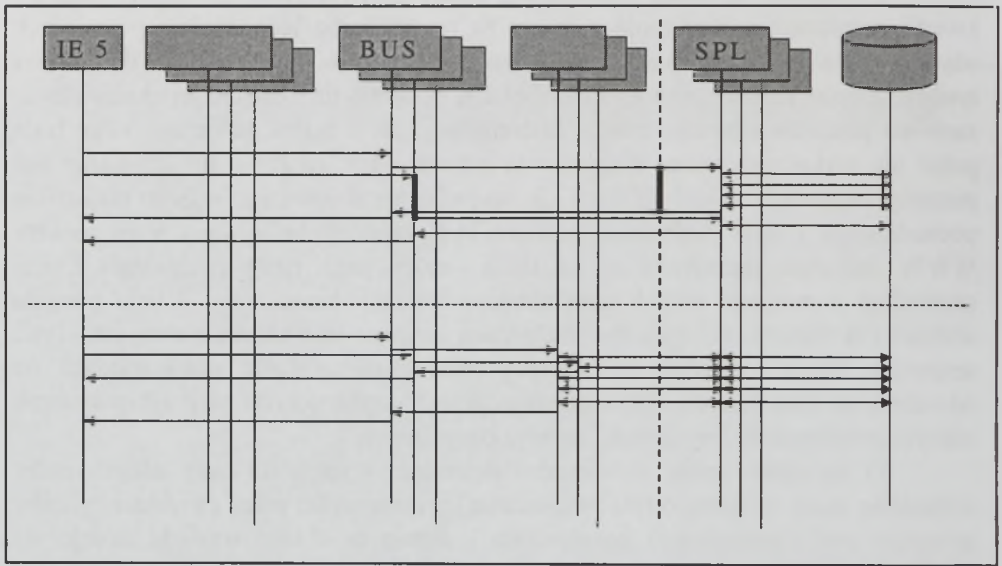
Jeśli założymy, że czas pojedynczego wykonania usługi systemu w pierwszym wypadku wynosi 3s oraz że wykonujący ją komponent alokuje 1MB pamięci operacyjnej, to łatwo zauważyć, że drugie żądanie wykonania tej samej usługi, następujące z opóźnieniem 1s, powoduje alokację kolejnego 1MB pamięci operacyjnej na kolejne 3s. Łączne jednoczesne zapotrzebowanie na pamięć operacyjną wynosi więc 1MB przez pierwszą sekundę, następnie 2 MB przez 2s i 1MB przez okres 1s. W drugim wypadku pierwsze wykonanie usługi rozpoczęte w warstwie biznesowej systemu przekazuje sterowanie do warstwy dostępu do danych. Komponent warstwy dostępu do danych alokuje 1MB. Następnie rozpoczyna się drugie wywołanie tej samej usługi, które, tak jak w poprzednim wypadku, przekazuje sterowanie do warstwy dostępu do danych. Przetwarzanie w warstwie dostępu do danych powoduje tym razem wykorzystanie wcześniej zaalokowanego już 1MB pamięci. Następnie wyniki przetwarzania są przekazywane zwrótnie do warstwy biznesowej i następuje zwolnienie zaalokowanych zasobów. W porównaniu z pierwszym wariantem łączne zapotrzebowanie na pamięć operacyjną jest mniejsze. Komponenty warstwy biznesowej przekazują sterowanie do warstwy dostępu do danych, a tym samym nie

alokują tyłu zasobów przez cały okres wykonywania usługi systemu, jak to ma miejsce w poprzednim wypadku. Z omówionego powyżej przykładu wynika, że przy jednoczesnym występowaniu setek czy nawet tysięcy żądań wykonania tych samych usług systemu wprowadzenie dodatkowej warstwy odpowiedzialnej za dostęp do danych może powodować dużo mniejsze chwilowe zapotrzebowanie na pamięć operacyjną, co w praktyce oznacza efektywniejsze wykorzystanie zasobów sprzętowych i w rezultacie obniżenie kosztów infrastruktury przy zachowaniu założonej wydajności.

Projektując aplikację internetową, należy przywiązywać szczególną wagę do kwestii rozpraszania obciążenia z uwagi na to, że liczba jednocześnie pracujących użytkowników może zmieniać się od kilku pojedynczych użytkowników do kilkuset tysięcy. Z tego powodu architektura aplikacji powinna umożliwiać jej skalowalność zarówno przez zwiększanie mocy obliczeniowej, jak i liczby serwerów, które będą przez nią wykorzystywane. Wszystko to powinno być możliwe do uzyskania bez potrzeby ingerencji w kod aplikacji. W wypadku spodziewanego dużego obciążenia pochodzącego z sieci skalowalna powinna być warstwa obsługiwana przez serwery WWW. Jednakże, niezależnie od ich ilości i mocy, część ruchu pochodząca z sieci spowoduje wywołanie metod komponentów warstwy biznesowej. Z tego powodu również i ta warstwa powinna być skalowalna zarówno jeśli chodzi o moc, jak i ilość serwerów. Część odwołań do warstwy biznesowej zostanie przekształcona na odwołania do bazy danych. Tak więc, ostatecznie, o szybkości realizacji usługi systemu zdecyduje dostępność i wydajność serwera bazy danych.

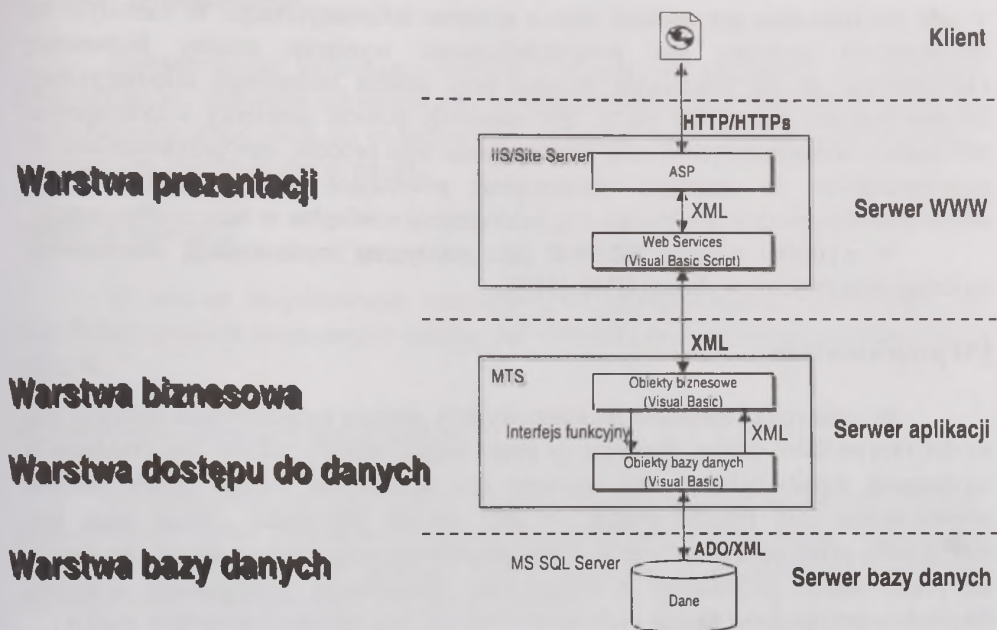
O ile łatwo sobie wyobrazić zwiększenie wydajności bazy danych przez rozbudowę mocy serwera, o tyle zwiększenie jej dostępności przez zwiększanie liczby serwerów jest rozwiązaniem kosztownym i złożonym. Z tego względu stosuje się rozwiązanie rozpraszające ruch kierowany do bazy danych, polegające na zastosowaniu dodatkowej warstwy aplikacji, tzw. warstwy dostępu do danych. Na rysunku 7. zilustrowano ideę rozpraszania obciążenia zrealizowaną w oparciu o zastosowanie tej warstwy. W pierwszym wariantcie usługa systemu skierowana do warstwy biznesowej powoduje zestawienie połączenia z bazą danych i wykonanie serii zapytań w ramach pojedynczego połączenia. Po wykonaniu wszystkich zapytań następuje zwrócenie wyników zapytań do warstwy biznesowej i zamknięcie połączenia z bazą danych. Załóżmy, że czas połączenia z bazą danych trwa 4s. Pojawiające się po 1s drugie wywołanie innej usługi systemu, zanim będzie mogło być zrealizowane w ramach skończonej, dostępnej liczby połączeń z bazą danych, musi w skrajnym wypadku oczekiwać na zwolnienie połączenia przez poprzednie wywołanie usługi systemu. W podanym przykładzie trwa to 3s (4s – 1s). Wprowadzenie dodatkowej warstwy dostępu, odpowiedzialnej za wykonywanie zapytań bazodanowych, powoduje, że połączenie z bazą jest nawiązywane za każdym razem w celu wykonania pojedynczego zapytania. Przy tym, po jego wykonaniu, połączenie z bazą danych jest zamykane. Dzięki takiemu rozwiązaniu żądanie połączenia z bazą danych pochodzące z wywołania drugiej usługi systemu nie musi oczekiwać na zakończenie obsługi wywołania pierwszej usługi, ale może być zaspokojone już po wykonaniu pierwszego zapytania z serii wymaganych przez pierwszą usługę. Powoduje to, co prawda, że

całkowity czas wykonania pierwszej usługi systemu nieznacznie się wydłuży, niemniej system jako całość działa wydajniej ze względu na minimalizację oczekiwań dostępu do bazy danych.



Rys.7. Rozpraszanie obciążenia bazy danych





Rys.8. Architektura systemu SZ@FiP

### 3. Modelowanie

Krytyczne znaczenia dla prawidłowego przebiegu wytwarzania systemu ma stosowanie formalnie zdefiniowanej metodyki jego modelowania. Tworzenie modeli pozwala z jednej strony na precyzyjne wyrażanie zakresu i obowiązków systemów biznesowego i informatycznego, z drugiej zaś na prowadzenie uzgodnień specyfikacji wymagań wobec systemu. Stosowanie wspólnego języka opisu z dobrze poznaną notacją i semantyką oraz większa precyzja języka formalnego w porównaniu z językiem naturalnym pozwalają uniknąć wieloznaczności specyfikacji. Stosowanie modeli pozwala na dokumentowanie zmian zarówno specyfikacji, jak i projektów. Uzyskiwany efekt to kompletna dokumentacja wszystkich zmian oraz możliwość przeglądu rozwoju systemu poprzez kolejne wersje modeli, a także możliwość precyzyjnego planowania rozwoju systemu na poziomie wymagań biznesowych.

Przy wytwarzaniu systemów zarządzania firmą wybór metody analizy procesów biznesowych do opisanie organizacji związany jest z koniecznością patrzenia na firmę przez pryzmat jej konkurencyjności. Dlatego w wypadku systemu SZ@FiP zrezygnowano z analizy struktury funkcyjnej firmy, a skoncentrowano się na modelowaniu procesów występujących wewnątrz firmy, w szczególności procesów dostarczających wartości klientowi. Jako praktyczną implementację metody analizy biznesowej wybrano Lynx 95 5.

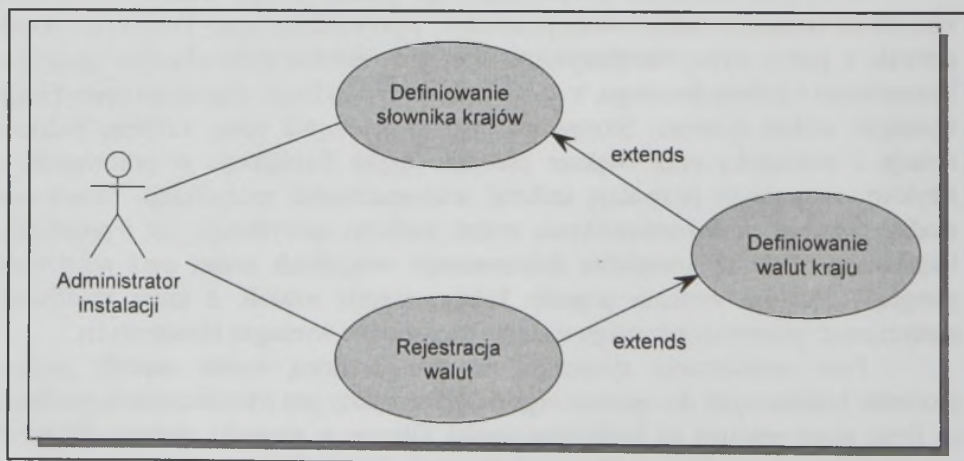
Wybór obiektowej metodyki projektowania [6] wynika w dużym stopniu z przyjętej metodyki analizy biznesowej [5], zakładającej dekompozycję procesów

w celu modelowania przypadków użycia systemu informatycznego. W metodyce tej modelowanie systemu jest podporządkowane wynikom analizy biznesowej i koncentruje się na wspieraniu procesu przy użyciu technologii informatycznej. Modele systemu tworzy się po to, aby stanowiły pomost pomiędzy wspomaganym procesem a wytworzonym, w celu wspomagania tego procesu, oprogramowaniem. W przeciwieństwie do metodyki strukturalnej powstające modele bardzo dobrze odzwierciedlają modele analityczne i są jednocześnie niezbędne w fazie implementacji.

W wypadku systemu **SZ@fir** jako praktyczną implementację obiektowego modelowania systemu wybrano UML [7][8].

## Cel projektowania

W omawianej metodzie punktem wyjścia procesu projektowania aplikacji jest model przypadków użycia dostarczony przez zespół analizy. Model ten przedstawia wymagania funkcjonalne wobec systemu (co system ma robić). Celem procesu projektowania jest przedstawienie, w jaki sposób przypadki użycia mają być realizowane przez oprogramowanie. Prace projektowe rozpoczynają się od zapoznania się przez zespół projektowy z dostarczoną specyfikacją funkcjonalną w postaci diagramów przypadków użycia i ich opisów oraz jej uzgodnienia z zespołem analizy.



Rys. 9. Przykładowy diagram przypadków użycia

## **Techniki projektowania**

Podstawowymi technikami, którymi posługuje się projektant, są: diagramy klas i obiektów, diagramy sekwencji oraz diagramy tablic. Diagramy klas i obiektów oraz diagramy tablic służą do modelowania statycznej części systemu, zaś diagramy sekwencji do modelowania dynamiki.

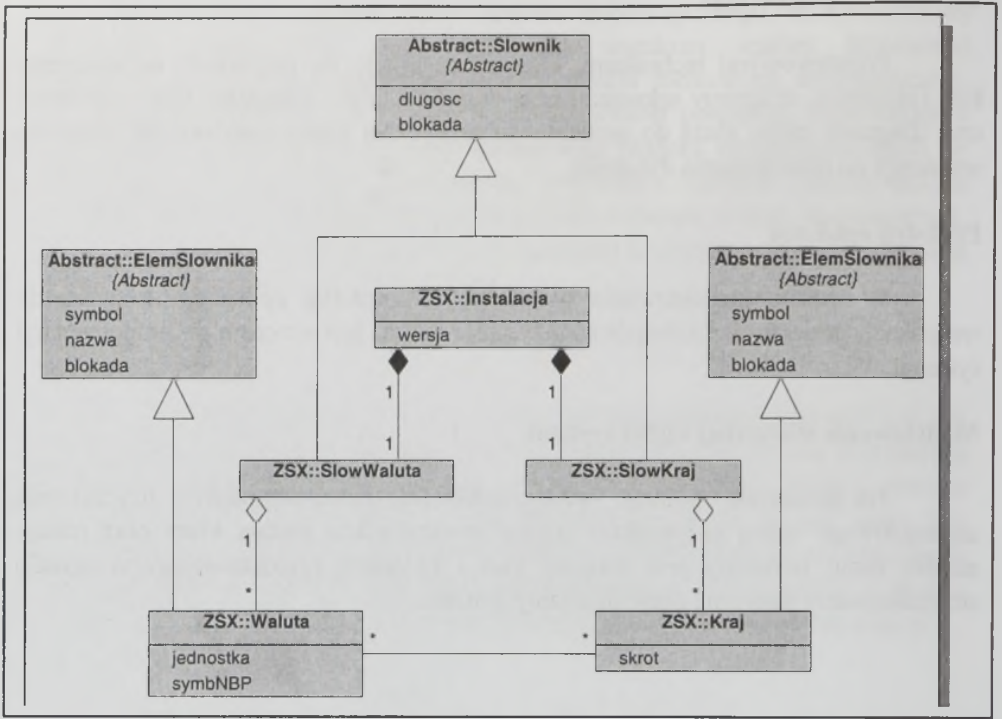
### **Prototyp aplikacji**

W trakcie projektowania tworzony jest prototyp aplikacji. Służy on do weryfikacji projektu przez zespół analizy, jak również jest wstępem do implementacji systemu.

### **Modelowanie statycznej części systemu**

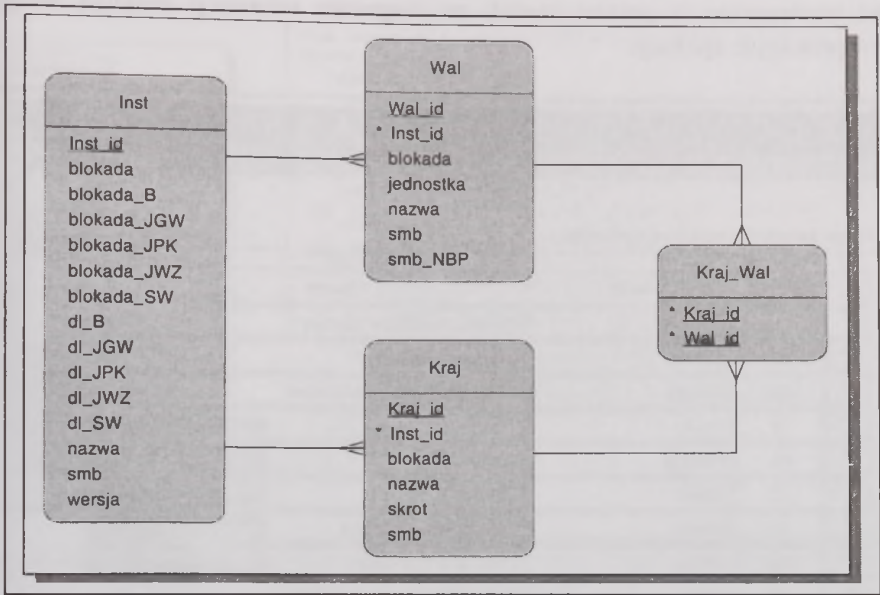
Na podstawie ogólnego opisu modułu (diagramu przypadków użycia) oraz szczegółowego opisu przypadków użycia wyodrębniane zostają klasy oraz relacje między nimi; tworzony jest diagram klas i obiektów przedstawiający w sposób sformalizowany statyczną część dziedziny systemu.





Rys. 10. Przykładowy diagram klas i obiektów

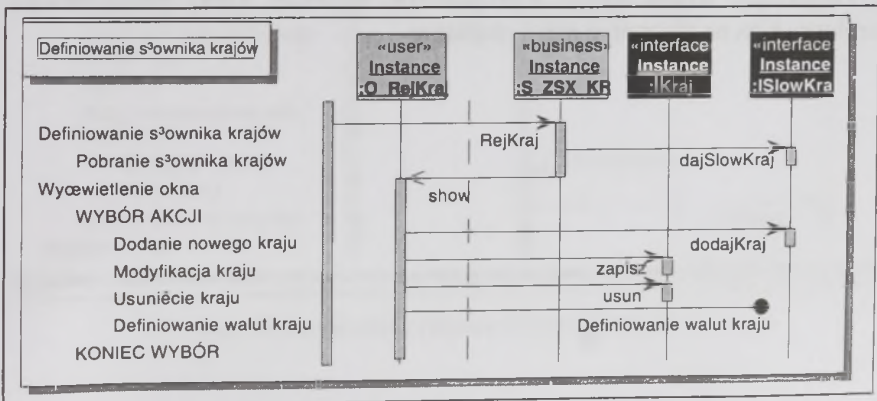
Diagram ten służy jako podstawa do stworzenia relacyjnego modelu danych, przedstawianego za pomocą diagramów tablic. Ze względu na optymalizację dostępu do danych model ten jest denormalizowany. W trakcie implementacji może podlegać dalszej denormalizacji - w ten sposób zawsze odzwierciedla fizyczną strukturę bazy danych.



Rys. 11. Przykładowy diagram tabel i relacji

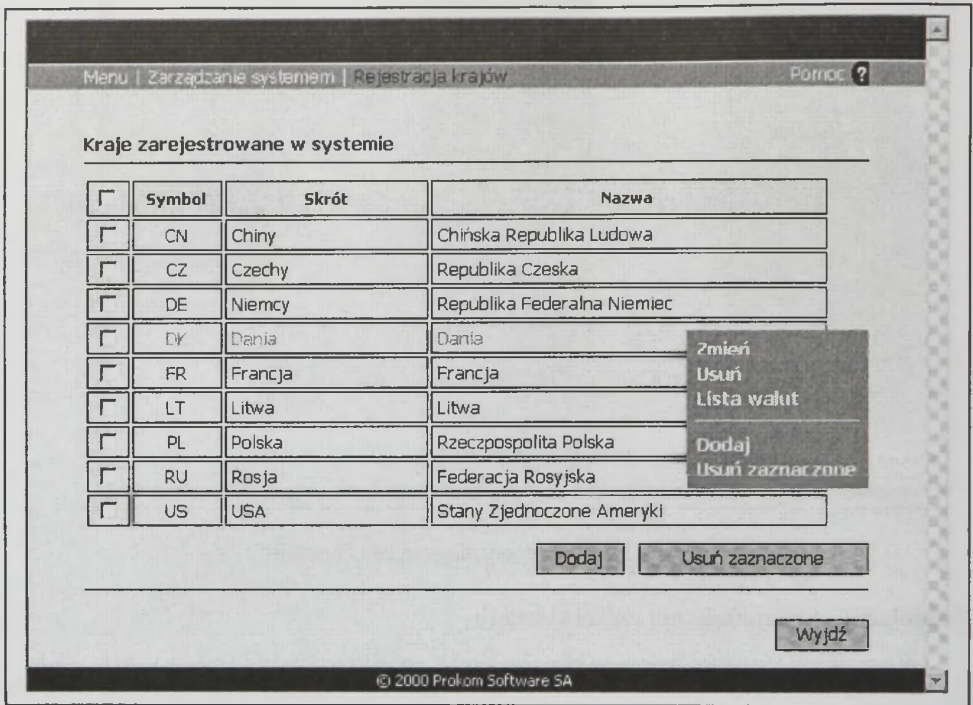
### Modelowanie dynamicznej części systemu

Dynamika systemu modelowana jest za pomocą diagramów sekwencji. Każdy przypadek użycia posiada swój diagram sekwencji będący jego uszczegółowieniem. Diagram ten pokazuje sposób interakcji użytkownika z systemem, określa zakres czynności, które użytkownik może wykonać. Elementy znajdujące się na diagramie sekwencji to: obiekty użytkownika (formatki), obiekty sterujące (odpowiedzialne za powoływanie formatek) oraz interfejsy (stanowiące zbiór serwisów udostępnionych na zewnątrz przez system).



Rys. 12. Przykładowy diagram sekwencji

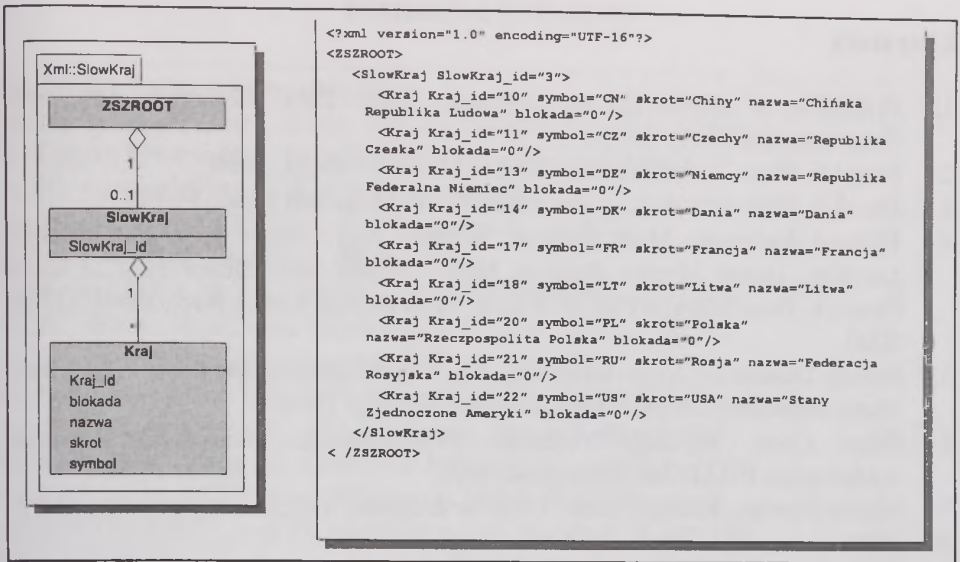
Formatki występujące w postaci modeli na diagramie sekwencji znajdują swoje odbicie w prototypie aplikacji.



Rys. 13. Prototyp aplikacji

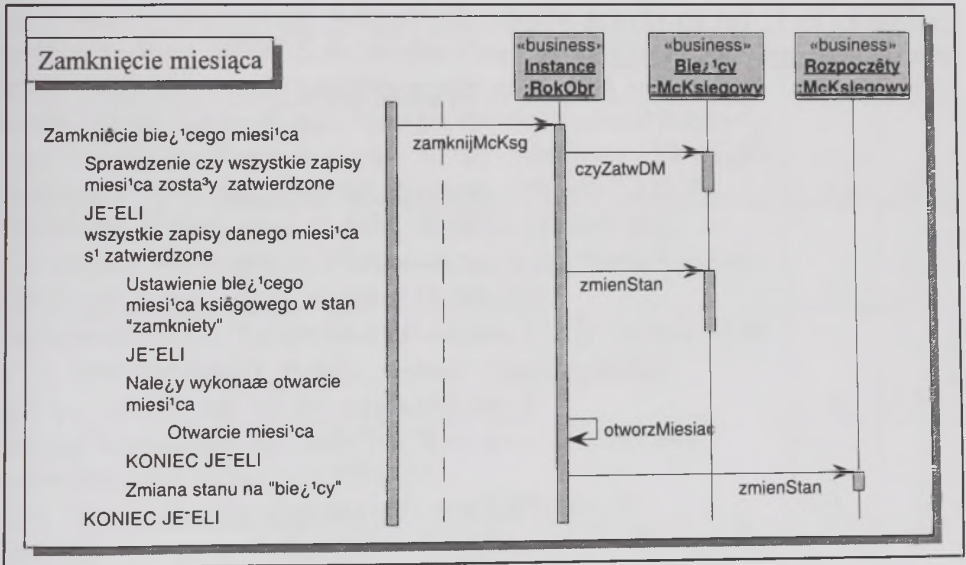
Szczególną wagę przywiązuje się do projektowania serwisów określających, w jaki sposób aplikacja użytkownika komunikuje się z serwerem aplikacji. Założono, że zarówno dane wejściowe, jak i wyjściowe będą przedstawiane jako dokumenty XML. Projektując serwis, należy zaprojektować struktury XML, przedstawiając je jako struktury klas na diagramie klas i obiektów.





Rys. 14. Przykładowe dane w postaci dokumentu XML

W wypadku złożonych serwisów niezbędne jest ich uszczegóławianie za pomocą kolejnych diagramów sekwencji.



Rys. 15. Przykładowy diagram sekwencji serwisu

## Literatura

1. Principles of application development, Class Pack 1516ACP. Microsoft Corporation, 1999.
2. David S. Platt: Understanding COM+, Microsoft Press, 1999.
3. David S. Platt: Introducing Microsoft .NET, Microsoft Press, 1999.
4. Richard Anderson, Mark Birbeck, Michael Kay, Steven Livingstone, Brian Loesgen, Didier Martin, Stephen Mohr, Nikola Ozu, Bruce Peat, Jonatan Pinnock, Peter Stark, Kevin Williams: Professional XML, Wrox Press Ltd.®, 2000.
5. Robert Dumnicki, Artur Kasprzyk, Mariusz Kozłowski: Analiza i projektowanie obiektowe, Wydawnictwo Helion, Gliwice 1998.
6. Peter Coad, Edward Yourdon: Projektowanie obiektowe, Oficyna wydawnicza READ ME, Warszawa 1994.
7. Martin Fowler, Kendall Scott: UML w kropelce, Oficyna Wydawnicza LTP, 2002.
8. UML Semantics, UML Notation. Rational Software Corporation, 1997.

Robert ZYSKOWSKI

Prokom Software SA, Gdynia,

ul. Śląska 35/37 Tel.:(0-58) 628-66-66

e-mail: [zyskowskir@prokom.pl](mailto:zyskowskir@prokom.pl)

## INDEKS AUTORÓW

Nazwisko Imię, adres autora	Rozdział	Tom
<b>Adamski</b> Włodzimierz, dr inż., Polskie Zakłady Lotnicze Sp. z o.o., ul. Wojska Polskiego 3, 39-300 Mielec Tel.:(0-17)7887188 e-mail: <a href="mailto:adamski@ptc.pl">adamski@ptc.pl</a> , <b>Artykuł 13, str. 167</b>	5	II
<b>Baczyński</b> Dariusz, Politechnika Warszawska, Instytut Elektroenergetyki, <b>Artykuł 35, str. 459</b>	7	II
<b>Baranowski</b> Jan, dr inż., THOMSON MULTIMEDIA POLSKA 05-500 Piaseczno ul. Gen. Okulickiego 7/9, Tel.:(0-22) 7571500 e-mail: <a href="mailto:JanBar@thmulti.com">JanBar@thmulti.com</a> , <b>Artykuł 14, str. 183</b>	5	II
<b>Bartnik</b> Ewa, prof. dr hab., Zakład Genetyki Uniwersytetu Warszawskiego, ul. Pawińskiego 5a 02-106 Warszawa, <b>Artykuł 36, str. 469</b>	7	II
<b>Biernacki</b> Piotr T., mgr inż., Corel EPM Representative, czł. Zarz. PRO i KA PiliT, wykł. PJWSTK ul. Wilanowska 14a/10, 00-422 Warszawa, tel.:(0-22) 71-111-71, e-mail: <a href="mailto:piotrek@pol.pl">piotrek@pol.pl</a> , <b>Artykuł 13, str. 175</b>	2	I
<b>Bliziński</b> Grzegorz, dr inż., Instytut Systemów Informatycznych, Wydział Cybernetyki WAT, Warszawa, <b>Artykuł 1, str. 5</b>	4	II
<b>Bobkowska</b> Anna, dr inż., Katedra Zastosowań Informatyki, Politechnika Gdańska, ul. Narutowicza 11/12, 80-952 Gdańsk, e-mail: <a href="mailto:annab@eti.pg.gda.pl">annab@eti.pg.gda.pl</a> , <b>Artykuł 21, str. 321</b>	3	I
<b>Boryka</b> Andrzej, BPSC S.A., 41-506 Chorzów, Ul. Gałęzki 61 tel.0-32-3493 500, e-mail: <a href="mailto:bpsc@bpsc.com.pl">bpsc@bpsc.com.pl</a> <b>Artykuł 28, str. 401</b>	6	II
<b>Buczek</b> Marek, Senior Project Manager Hewlett-Packard Polska Sp. z o.o. University Business Center II, ul. Szturmowa 2A 02-678 Warszawa, e-mail: <a href="mailto:marek_buczek@hp.com">marek_buczek@hp.com</a> , <b>Artykuł 7, str. 89</b>	4	II
<b>Chmielarz</b> Witold, prof. dr hab., Wydział Zarządzania, Uniwersytet Warszawski, Warszawa, ul. Szturmowa 3, e-mail: <a href="mailto:witek@mail.wz.uw.edu.pl">witek@mail.wz.uw.edu.pl</a> , <b>Artykuł 22, str. 331</b>	3	I
<b>Ciarkowski</b> Jacek, Politechnika Gdańska, Ul. G. Narutowicza 11/12, 80-952 Gdańsk, Polska, e-mail: <a href="mailto:ciara@pg.gda.pl">ciara@pg.gda.pl</a> , Tel/Fax: +48 58 347 27 27, <b>Artykuł 1, str. 5</b>	1	I
<b>Cierpisz</b> Tomasz mgr inż., MINTECH sp. z o.o. ul. Polarna 8 40-698 Katowice tel/fax (32) 2525442, email: <a href="mailto:Tomasz.Cierpisz@mintech.pl">Tomasz.Cierpisz@mintech.pl</a> , <b>Artykuł 2, str. 35</b>	4	II
<b>Cysewski</b> Grzegorz, Politechnika Gdańska, ul. G. Narutowicza 11/12, 80-952 Gdańsk, e-mail: <a href="mailto:grechut@eti.pg.gda.pl">grechut@eti.pg.gda.pl</a> , tel: +48 58 347 13 18, Fax: +48 58 347 27 27, <b>Artykuł 2 str. 13</b>	1	I
<b>Czarnacka-Chrobot</b> Beata, dr, Katedra Informatyki Gospodarczej, Szkoła Główna Handlowa, Al. Niepodległości 162, 02-554 Warszawa, tel. (22) 849 53 95, e-mail: <a href="mailto:bczarn@sgh.waw.pl">bczarn@sgh.waw.pl</a> , <b>Artykuł 23, str. 357</b>	3	I



<b>Dalkowski Bogumił Tomasz, Stowarzyszenie Project Management Polska, Artykuł 3, str. 41</b>	4	II
<b>Deminet Jarosław, dr, ComputerLand S.A., 02-486 Warszawa, Al.Jerozolimskie 180, tel 0-22-571 1000, fax 0-22-571 1101 e-mail: <a href="mailto:jdeminet@computerland.pl">jdeminet@computerland.pl</a>. Artykuł 4, str. 53</b>	4	II
<b>Dyczkowski Mirosław, dr, Instytut Informatyki Ekonomicznej, Akademia Ekonomiczna we Wrocławiu, 53-345 Wrocław, ul. Komandorska 118/120, e-mail: <a href="mailto:dyczkows@manager.ae.wroc.pl">dyczkows@manager.ae.wroc.pl</a>. Artykuł 5, str. 61</b>	4	II
<b>Dziduszko Marek, mgr, MTD – Technologie Rynkowe, 05-806 Komorów, ul. Mieczysława 9, tel. 0-prefix-22 759 10 11, e-mail:<a href="mailto:as_mtd@poczta.onet.pl">as_mtd@poczta.onet.pl</a>. Artykuł 6, str. 77</b>	4	II
<b>Gawel Edward, mgr, MSWiA – Departament Rozwoju Informatyki i Rejestrów Państwowych, e-mail: <a href="mailto:egawel@dli.mswia.gov.pl">egawel@dli.mswia.gov.pl</a>. Artykuł 14, str. 189</b>	2	I
<b>Goczyła Krzysztof, Politechnika Gdańska, ul. G. Narutowicza 11/12, 80-952 Gdańsk, e-mail: <a href="mailto:kris@eti.pg.gda.pl">kris@eti.pg.gda.pl</a>, tel.: +48 58 347 13 18, fax: +48 58 347 27 27, Artykuł 2, str. 13</b>	1	I
<b>Grabara Iwona, mgr inż., Instytut Informatyki i Ekonometrii, Wydział Zarządzania Politechniki Częstochowskiej, Al. Armii Krajowej 19b, 42-200 Częstochowa, tel. (0-34) 3250-242, e-mail: <a href="mailto:grabara@czestochowa.net.pl">grabara@czestochowa.net.pl</a> , Artykuł 15, 24, str. 191, 379</b>	3 5	I II
<b>Grabara Janusz Krzysztof, dr inż., Instytut Informatyki i Ekonometrii, Wydział Zarządzania Politechniki Częstochowskiej, Al. Armii Krajowej 19b, 42-200 Częstochowa, tel. (0-34) 3250-240, e-mail: <a href="mailto:grabara@czestochowa.net.pl">grabara@czestochowa.net.pl</a> , Artykuł 15, 37 str. 207, 475</b>	2 7	I II
<b>Grocholski Tadeusz, Przedsiębiorstwo Państwowe Porty Lotnicze – Agencja Ruchu Lotniczego, Al. Żwirki i Wigury 1, 00-906 Warszawa. e-mail: <a href="mailto:t.grocholski@polish-airports.com">t.grocholski@polish-airports.com</a>. Artykuł 16, str. 201</b>	5	II
<b>Gruber Jacek, Politechnika Wrocławska, Wydział Informatyki i Zarządzania, Wydziałowy Zakład Informatyki, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław e-mail: <a href="mailto:gruber@ci.pwr.wroc.pl">gruber@ci.pwr.wroc.pl</a>. Artykuł 3, str. 23</b>	1	I
<b>Helt Piotr, dr inż., Instytut Elektroenergetyki, Politechniki Warszawskiej ul. Koszykowa 75, 00-662 Warszawa, <a href="mailto:piotr.helt@ien.pw.edu.pl">piotr.helt@ien.pw.edu.pl</a>, Artykuł 17, 35, str. 213, 459</b>	5 7	II II
<b>Iskierka Iwona, dr, Instytut Informatyki i Ekonometrii, Wydział Zarządzania Politechniki Częstochowskiej, Al. Armii Krajowej 19b, 42-200 Częstochowa, tel. (0-34) 3250-242, Artykuł 4, str. 77</b>	1	I
<b>Iskierka Sławomir, dr hab. inż. Prof. P.Cz, Politechnika Częstochowska Instytut Informatyki, Wydział Elektryczny, ul Armii Krajowej 17, 42-200 Częstochowa Artykuł 18, str. 225</b>	5	II

<b>Jelonek Dorota</b> , dr, Katedra Informatyki Ekonomicznej, Wydział Zarządzania Politechniki Częstochowskiej, Al. Armii Krajowej 19b, 42-200 Częstochowa, tel. (0-34) 325-03-91, e-mail: <a href="mailto:jelonek@zim.pcz.czest.pl">jelonek@zim.pcz.czest.pl</a> , <b>Artykuł 38, str. 483</b>	7	II
<b>Karasek Jan</b> , mgr inż., Xtrade SA Warszawa, ul. Wynalazek 4 Tel.:(0-22)640-08-41, e-mail: <a href="mailto:jan.karasek@xtrade.com.pl">jan.karasek@xtrade.com.pl</a> , <b>Artykuł 27, str. 373</b>	5	II
<b>Kaczarewski Tadeusz</b> – Nacz. Inżynier Górniczy KWB „Turów” S.A., 59-916 Bogatynia 3, tel. 0-75-7735679, fax 0-75-7735117, <b>Artykuł 19, str. 231</b>	5	II
<b>Kędzierski Stanisław</b> , dr hab., AE Katowice, 1-go Maja 50 e-mail: <a href="mailto:kedziers@ae.katowice.pl">kedziers@ae.katowice.pl</a> , <b>Artykuł 5, str. 81</b>	1	I
<b>Kocięcki Leszek Maria</b> , Warszawa ul. Pożarowa 9/11 tel. 0 601 223 777, e-mail: <a href="mailto:leko@crm.com.pl">leko@crm.com.pl</a> <b>Artykuł 20, str. 255</b>	5	II
<b>Komorowski Krzysztof</b> , mgr inż., ComputerLand SA, 02-486 Warszawa, Al.Jerozolimskie 180, tel. 0-22-571 1410, e-mail: <a href="mailto:kkomorowski@computerland.pl">kkomorowski@computerland.pl</a> , <b>Artykuł 29, str. 409</b>	6	II
<b>Kontinuum Sp. z o.o.</b> ul. Przemysłowa 30, 00-450 Warszawa, tel. 622-38-41, fax 622-39-41, <b>Artykuł 30, str. 413</b>	6	II
<b>Kosieradzki Wiesław</b> , Dr inż., Centrum Rozwiązań Menedżerskich S.A. Warszawa, ul. Pożarowa 9/11 Tel. (0-22) 811-44-40, e-mail: <a href="mailto:w.kosieradzki@crm.com.pl">w.kosieradzki@crm.com.pl</a> , <b>Artykuł 16, str. 213</b>	2	I
<b>Kot Sebastian</b> , mgr inż., Politechnika Częstochowska, Wydział Zarządzania, <b>Artykuł 37, str. 475</b>	7	II
<b>Krupa Kazimierz</b> , dr inż., Uniwersytet Rzeszowski, e-mail: <a href="mailto:kkrupa@pf.pl">kkrupa@pf.pl</a> , <b>Artykuł 17, str. 235</b>	2	I
<b>Krzemiński Janusz</b> , dr inż., Politechnika Częstochowska Instytut Informatyki, Wydział Elektryczny, ul Armii Krajowej 17, 42-200 Częstochowa, <b>Artykuł 18, str. 225</b>	5	II
<b>Kuczyński Łukasz</b> , Politechnika Częstochowska ,Instytut Matematyki i Informatyki, tel. 604559300, ul. Dąbrowskiego 73, e-mail: <a href="mailto:hpiech@adm.pcz.czest.pl">hpiech@adm.pcz.czest.pl</a> , , <b>Artykuł 8,40 str. 119, 517</b>	1 7	I II
<b>Kulik Wojciech</b> , mgr, Katedra Zarządzania, Politechnika Lubelska, ul. Nadbystrzycka 38; 20-618 Lublin, e-mail: <a href="mailto:kulik@antenor.pl.lublin.pl">kulik@antenor.pl.lublin.pl</a> , <b>Artykuł 25, str. 385</b>	3	I
<b>Kulisiewicz Tomasz</b> , mgr inż. e-mail: <a href="mailto:tomkul@idea.net.pl">tomkul@idea.net.pl</a> , <b>Artykuł 39, str. 493</b>	7	II
<b>Kurzydłowska Anna</b> , dr, Hewlett-Packard Polska Sp. z o.o.University Business Center II, ul, Szturmowa 2A 02-678 Warszawa, e-mail: <a href="mailto:anna_kurzydłowska@hp.com">anna_kurzydłowska@hp.com</a> , <b>Artykuł 7, str. 89</b>	4	II
<b>Leks Dariusz</b> , Politechnika Częstochowska ,Instytut Matematyki i Informatyki, tel. 604559300, ul. Dąbrowskiego 73, e-mail: <a href="mailto:hpiech@adm.pcz.czest.pl">hpiech@adm.pcz.czest.pl</a> , , <b>Artykuł 8, 40 str. 119, 517</b>	1 7	I II

<b>Lis Tomasz</b> , mgr inż., Instytut Informatyki i Ekonometrii, Wydział Zarządzania Politechniki Częstochowskiej, Al. Armii Krajowej 19b, 42-200 Częstochowa, tel. (0-34) 325 03 78, <b>Artykuł 26, str. 397</b>	3	I
<b>Ładyga Jarosław</b> , niezależny konsultant, współpracownik firmy TORN Sp.z o.o., ul. Sosnkowskiego 17, 02-495 Warszawa, e-mail: <a href="mailto:iladyga@torn.com.pl">iladyga@torn.com.pl</a> , <b>Artykuł 18, str. 247</b>	2	I
<b>Łapeta Jarosław</b> , mgr inż. Instytut Informatyki i Ekonometrii, Wydział Zarządzania Politechniki Częstochowskiej, Al. Armii Krajowej 19b, 42-200 Częstochowa, tel. (0-34) 325 03 78, e-mail: <a href="mailto:ilapeta@zim.pcz.czest.pl">ilapeta@zim.pcz.czest.pl</a> , <b>Artykuł 15, str. 191</b>	5	II
<b>Łukasik-Makowska Barbara</b> , dr inż. Instytut Informatyki Ekonomicznej, Akademia Ekonomiczna im. O. Langego, ul. Komandorska 118/120, 53-345 Wrocław, e-mail: <a href="mailto:blm@manager.ae.wroc.pl">blm@manager.ae.wroc.pl</a> , <b>Artykuł 8, str. 103</b>	4	II
<b>Machala Wojciech</b> , mgr inż., Instytut Systemów Informatycznych, Wydział Cybernetyki WAT, 00-908 Warszawa, ul. Kaliskiego 2, tel. 0-22 68-37-410, e-mail: <a href="mailto:wmachala@isi.wat.waw.pl">wmachala@isi.wat.waw.pl</a> , <b>Artykuł 9, str. 115</b>	4	II
<b>Mazur Zygmunt</b> , dr hab. inż., Prof. PWr., Politechnika Wrocławska, Wydział Informatyki i Zarządzania, Wydziałowy Zakład Informatyki, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, e-mail: <a href="mailto:mazur@ci.pwr.wroc.pl">mazur@ci.pwr.wroc.pl</a> , <b>Artykuł 3, str. 23</b>	1	I
<b>Mazurkiewicz Wojciech</b> , FileNET Polska Sp. z o.o. 00-867 Warszawa, Al. Jana Pawła II nr 29, tel.0-22-5289207 e-mail: <a href="mailto:wmazurkiewicz@filenet.com">wmazurkiewicz@filenet.com</a> <b>Artykuł 31, str. 431</b>	6	II
<b>Mączyńska-Spiewak Grażyna</b> , mgr, Politechnika Częstochowska, ul. Dąbrowskiego 69, 42-200 Częstochowa, <b>Artykuł 41, str. 527</b>	7	II
<b>Miłosz Marek</b> , dr inż., Katedra Informatyki Politechnika Lubelska, ul. Nadbystrzycka 36 b, 20-618 Lublin, tel. (0 81) 525 20 46, fax. (0 81) 532 30 96 w.17, e-mail: <a href="mailto:marekm@pluton.pol.lublin.pl">marekm@pluton.pol.lublin.pl</a> , <b>Artykuł 10, 21, str. 126, 283</b>	4 5	II II
<b>Morzy Mikołaj</b> , mgr inż., Instytut Informatyki Politechniki Poznańskiej, Poznań ul. Piotrowo 3A tel.: (0-61) 665-21-27, email: <a href="mailto:Mikolaj.Morzy@cs.put.poznan.pl">Mikolaj.Morzy@cs.put.poznan.pl</a> , <b>Artykuł 6, str. 93</b>	1	I
<b>Niemiec Andrzej</b> , dr, PRIM sp z o.o. Wrocław, ul M Skłodowskiej Curie 43/3, 50-369 Wrocław, tel. (071) 328 30 94, e-mail: <a href="mailto:prim@prim.com.pl">prim@prim.com.pl</a> , <b>Artykuł 19, 22, str. 261, 293</b>	2 5	I II
<b>Nikodem Ryszard</b> , mgr, Instytut Informatyki Ekonomicznej, Akademia Ekonomiczna we Wrocławiu, 53-345 Wrocław, ul. Komandorska 118/120, tel/fax (0-71) 368 03 76, e-mail: <a href="mailto:nikodem@manager.ae.wroc.pl">nikodem@manager.ae.wroc.pl</a> , <b>Artykuł 7, str. 109</b>	1	I



<b>Nowak Jerzy Stanisław</b> , mgr inż., ComputerLand S.A. – 40-955 Katowice, ul. Bytkowska 1B, tel. (0-32) 789 96 01, e-mail: <a href="mailto:jnowak@computerland.pl">jnowak@computerland.pl</a> ; Polskie Towarzystwo Informatyczne, 40-012 Katowice, ul. Św.Jana 10, tel. 0-32-2539678 <a href="mailto:Jerzy.Nowak@pti.org.pl">Jerzy.Nowak@pti.org.pl</a> , <b>Artykuł 27, str. 403</b>	3	I
<b>Nowak Stefan</b> , mgr inż., Instytut Informatyki i Ekonometrii, Wydział Zarządzania Politechniki Częstochowskiej, Al. Armii Krajowej 19b, 42-200 Częstochowa, tel. (0-34) 325 03 78, e-mail: <a href="mailto:snowak@zim.pcz.czest.pl">snowak@zim.pcz.czest.pl</a> , <b>Artykuł 26, str. 397</b>	3	I
<b>Pawlowski Piotr</b> , Polskie Centrum Teleinformatyki – Wrocław, e-mail: <a href="mailto:Piotr.Pawlowski@pct.pl">Piotr.Pawlowski@pct.pl</a> , <b>Artykuł 23, str. 325</b>	5	II
<b>Piech Henryk</b> , prof. dr hab. inż., Wydział Inżynierii Mechanicznej i Informatyki, Politechnika Częstochowska, ul. Dąbrowskiego 73, 42-200 Częstochowa, (0-34) 325-03-31, e-mail: <a href="mailto:Hpiech@adm.pcz.czest.pl">Hpiech@adm.pcz.czest.pl</a> , <b>Artykuł 8, 40, str. 119, 517</b>	1 7	I II
<b>Piotrowski Paweł</b> , Politechnika Warszawska, Instytut Elektroenergetyki, <b>Artykuł 35, str. 459</b>	7	II
<b>Ptak Aleksandra</b> , Politechnika Częstochowska, Instytut Matematyki i Informatyki, tel. 604559300, ul. Dąbrowskiego 73, e-mail: <a href="mailto:hpiech@adm.pcz.czest.pl">hpiech@adm.pcz.czest.pl</a> , <b>Artykuł 40, str. 517</b>	7	II
<b>Premium Technology</b> <b>Artykuł 32, str. 441</b>	6	II
<b>Rekosz Dariusz</b> , Południowy Koncern Energetyczny S.A. Elektrownia „Lagisza”, ul. Pokoju 14, 42-504 Będzin e-mail: <a href="mailto:lagisza@elektrownia-lagisza.com.pl">lagisza@elektrownia-lagisza.com.pl</a> <b>Artykuł 24, str. 331</b>	5	II
<b>Siemieński Piotr</b> , dr inż., Management Consulting, Deloitte & Touche Sp. z o.o., ul. Fredry 6, 00-097 Warszawa. <b>Artykuł 32, str. 441</b>	6	II
<b>Sołowczuk Marek</b> , mgr, - St.Specjalista ds. GSI, KWB „Turów” S.A. 59-916 Bogatynia 3; e-mail: <a href="mailto:geologia@kwbturow.com.pl">geologia@kwbturow.com.pl</a> , <b>Artykuł 19, str. 231</b>	5	II
<b>Spalek Seweryn</b> , mgr inż., Politechnika Śląska, Wydział Organizacji i Zarządzania Katedra Informatyki i Ekonometrii, e-mail: <a href="mailto:spalek@polsl.gliwice.pl">spalek@polsl.gliwice.pl</a> , <b>Artykuł 11, str. 139</b>	4	II
<b>Sukiennik Jerzy</b> , Zarząd Portu Gdańsk SA, Gdańsk e-mail: <a href="mailto:Jerzy.Sukiennik@portgdansk.pl">Jerzy.Sukiennik@portgdansk.pl</a> , <b>Artykuł 25, str. 343</b>	5	II
<b>Szkic-Czech Ewa</b> , dr, Elektrownia „Opole” SA – 0-77-4235145 0-603 777 501, e-mail: <a href="mailto:eszkc@elopole.pol.pl">eszkc@elopole.pol.pl</a> <b>Artykuł 41, str. 533</b>	7	II
<b>Szopa Janusz</b> , prof. dr hab., Instytut Informatyki i Ekonometrii, Wydział Zarządzania Politechniki Częstochowskiej, Al. Armii Krajowej 19b, 42-200 Częstochowa, tel. (0-34) 325 04 98, e-mail: <a href="mailto:szopa@matinf.pcz.czest.pl">szopa@matinf.pcz.czest.pl</a> , <b>Artykuł 41, str. 527</b>	7	II

<b>Szyjewski Zdzisław</b> , prof. dr hab., Instytut Informatyki w Zarządzaniu, Uniwersytet Szczeciński, ul. Mickiewicza 66, 71-101 Szczecin, e-mail: <a href="mailto:zszyjew@uoo.univ.szczecin.pl">zszyjew@uoo.univ.szczecin.pl</a> , <b>Artykuł 25, str. 385</b>	3	I
<b>Tatarska Barbara</b> , mgr, Progress Software Sp. z o.o., 04-501 Warszawa, ul. Płowiecka, tel 0-22-673 1044, fax 0-22-610 9483, e-mail: <a href="mailto:btarsk@progress.com">btarsk@progress.com</a> , <b>Artykuł 33, str. 445</b>	6	II
<b>Traczyk Tomasz</b> , dr inż., Politechnika Warszawska, Instytut Automatyki i Informatyki Stosowanej, ul. Nowowiejska 15/19, 00-665 Warszawa e-mail: <a href="mailto:ttraczyk@ia.pw.edu.pl">ttraczyk@ia.pw.edu.pl</a> , URL: <a href="http://www.ia.pw.edu.pl/~ttraczyk">http://www.ia.pw.edu.pl/~ttraczyk</a> , <b>Artykuł 9, str. 129</b>	1	I
<b>Tyrowicz Andrzej</b> , Dyrektor Departamentu Informatyki Głównego Urzędu Ceł w likwidacji, 00-960 Warszawa, Świętokrzyska 12, tel: 22 - 5817 185, fax: 5817 187 email: <a href="mailto:atyrowicz@qdnnet.pl">atyrowicz@qdnnet.pl</a> , <a href="mailto:Dyrektor-DI@guc.ac.gov.pl">Dyrektor-DI@guc.ac.gov.pl</a> , <b>Artykuł 26, str. 357</b>	5	II
<b>Valenta Marek A.</b> , dr inż., Katedra Informatyki AGH Kraków, Al. Mickiewicza 30, e-mail: <a href="mailto:valenta@agh.edu.pl">valenta@agh.edu.pl</a> , <b>Artykuł 12, str. 163</b>	1	I
<b>Welenc Piotr</b> , mgr, Zaoczne Studium Doktoranckie, Instytut Badań Systemowych PAN, Warszawa, ul. Newelska 6, tel. 603 692276, e-mail: <a href="mailto:p.welenc@ibspan.waw.pl">p.welenc@ibspan.waw.pl</a> , <b>Artykuł 12, str. 149</b>	4	II
<b>Weźgowiec Zbigniew</b> , dr inż., Politechnika Częstochowska Instytut Informatyki, Wydział Elektryczny, ul Armii Krajowej 17, 42-200 Częstochowa, <b>Artykuł 18, str. 225</b>	5	II
<b>Wierchoń Lilianna</b> , mgr inż., ComputerLand S.A. 02-486 Warszawa, Al. Jerozolimskie 180, tel. +48 22 571 11 32, e-mail: <a href="mailto:lwierchon@computerland.pl">lwierchon@computerland.pl</a> , <b>Artykuł 20, str. 291</b>	2	I
<b>Winiarska Marzena</b> , mgr, Prokom Software S.A. Gdynia, ul. Śląska 23/25 Tel.:(0-58) 628-69-63, e-mail: <a href="mailto:winiarskam@prokom.pl">winiarskam@prokom.pl</a> , <b>Artykuł 29, str. 433</b>	3	I
<b>Wiśniewska Elżbieta</b> , dr, BRE Bank SA Warszawa, ul. Długa 44/55 Tel.:(0-22) 829-17-70, e-mail: <a href="mailto:elzbieta.wisniewska@brebank.com.pl">elzbieta.wisniewska@brebank.com.pl</a> , <b>Artykuł 27, str. 373</b>	5	II
<b>Wojciechowski Marek</b> , Politechnika Poznańska, Instytut Informatyki, ul. Piotrowo 3a, 60-965 Poznań, e-mail: <a href="mailto:marek,mzakrz@cs.put.poznan.pl">marek,mzakrz@cs.put.poznan.pl</a> , <b>Artykuł 6, 10, str. 93, 141</b>	1	I
<b>Zakrzewicz Maciej</b> , Politechnika Poznańska, Instytut Informatyki ul. Piotrowo 3a, 60-965 Poznań, e-mail: <a href="mailto:marek,mzakrz@cs.put.poznan.pl">marek,mzakrz@cs.put.poznan.pl</a> , <b>Artykuł 10, 11, str. 141, 153</b>	1	I
<b>Zygmunt Anna</b> , dr inż., Katedra Informatyki AGH Kraków, Al. Mickiewicza 30, e-mail: <a href="mailto:azygmunt@agh.edu.pl">azygmunt@agh.edu.pl</a> , <b>Artykuł 12, str. 163</b>	1	I

<b>Zyskowski Robert</b> , Prokom Software SA, Gdynia, ul. Śląska 35/37 Tel.:(0-58) 628-66-66, e-mail: <a href="mailto:zyskowskir@prokom.pl">zyskowskir@prokom.pl</a> , <b>Artykuł 30, str. 447</b>	3	I
<b>Żak Paweł</b> , Siemens Sp. z o.o., Dyr. Siemens Business Services 03-821 Warszawa, ul. Żupnicza 11, tel. 0-22-870 9700 <a href="mailto:sbs@siemens.pl">sbs@siemens.pl</a> <b>Artykuł 34, str. 453</b>	6	II





