Polskie Towarzystwo Informatyczne

PTI

Dylematy zarządzania projektem informatycznym

Redakcja naukowa:

Marek Miłosz

Janusz K. Grabara

Dylematy zarządzania projektem informatycznym

Dylematy
zarządzania
projektem
informatycznym

Dylematy zarządzania projektem informatycznym

Redakcja naukowa:

Marek Miłosz
Janusz K. Grabara

Polskie Towarzystwo Informatyczne – Oddział Górnośląski
Katowice 2006

Recenzenci:
Prof. P.Cz. dr hab. Nadia Gubarieni
Prof. dr hab. Maria Nowicka-Skowron
Prof. P.Cz. dr hab. Henryk Piech
Prof. PW, dr hab. inż. Kazimierz Waćkowski

Copyright © 2006 Polskie Towarzystwo Informatyczne
Oddział Górnośląski

ISBN 10: 83-60810-06-0
ISBN 13: 978-83-60810-06-4

Redakcja techniczna mgr inż. Tomasz Lis, mgr inż. Jerzy S. Nowak
Projekt okładki Marek J. Piwko

Utwór w całości ani we fragmentach nie może być powielany ani rozpowszechniany za pomocą urządzeń elektronicznych, mechanicznych, kopiujących, nagrywających i innych, w tym również nie może być umieszczany ani rozpowszechniany w postaci cyfrowej zarówno w Internecie, jak i w sieciach lokalnych bez pisemnej zgody posiadacza praw autorskich.

Polskie Towarzystwo Informatyczne
Oddział Górnośląski
40-074 Katowice, ul. Raciborska 3
tel. (0 32 251 9811) , e-mail: Katowice@pti.org.pl
www.pti.katowice.pl

Fotokopie, druk i oprawę
Wykonano w Zakładzie Graficznym Politechniki Śląskiej w Gliwicach
zam.402/06

SPIS TREŚCI

CZĘŚĆ 1

I	Zarządzanie projektami informatycznymi – pomiędzy formalizmem a elastycznością *Marek Miłosz*	9
II	Zarządzanie wymaganiami w procesie wytwórczym oprogramowania *Grzegorz Bliźniuk, Anna Żak*	23
III	Zarządzanie projektami IT – model mapowania kompetencji i aktywności *Kazimierz Frączkowski, Aleksandra Sidorczyk*	39
IV	Zagadnienia wdrażania zintegrowanych systemów zarządzania projektami w firmie informatycznej *Jan Werewka*	61
V	Typowe czynniki niepowodzenia w realizacji informatycznych przedsięwzięć projektowych – spojrzenie Standish Group *Beata Czarnacka-Chrobot*	77
VI	Jakość jako podstawa efektywnego wdrożenia systemu informatycznego *Anna Kaczorowska, Anna Pamuła*	105
VII	Zarządzanie projektem „MOVEX-ISO-FAMEG” *Krzysztof Jasiorowski, Jerzy Stanisław Nowak, Remigiusz Jasiorowski*	121
VIII	Efektywne zarządzanie ryzykiem w projektach wdrożeń zintegrowanych systemów informatycznych *Anna Kaczorowska*	141
IX	Szczypta psychologii dla kierowników projektów *Ewa Mizerska*	159

CZĘŚĆ 2

X Project servant – zintegrowane środowisko wspomagania wytwarzania oprogramowania w firmie informatycznej 169
Marek Sujdak, Jan Werewka, Dawid Pytka

XI Metody zapewnienia użyteczności oprogramowania 185
Beata Hysa

XII Wpływ kultury organizacyjnej na wdrożenie systemu informatycznego w Abanku 193
Justyna Maria Bugaj

XIII Wspomaganie działalności przedsiębiorstwa przy pomocy technologii informacyjnej 207
Janusz K. Grabara

CZĘŚĆ 3

XIV Technologia informacyjna w procesach nieliniowego montażu filmowego i telewizyjnego 215
Marcin Dudziński

CZĘŚĆ 1

ROZDZIAŁ I

ZARZĄDZANIE PROJEKTAMI INFORMATYCZNYMI – POMIĘDZY FORMALIZMEM A ELASTYCZNOŚCIĄ

Marek MIŁOSZ

Wstęp

W latach 70-80 XX wieku w obszarze wytwarzania oprogramowania uwidocznił się kryzys. Kryzys przejawił się dużym odsetkiem nieukończonych projektów informatycznych (w tym również projektów ukończonych, ale które trudno było uznać za sukces) i zwiększonymi kosztami utrzymania systemów informatycznych, spowodowanymi zastraszająca niską ich jakością. Badanie tego zjawiska rozpoczęto w 1987 roku, ale bardziej kompleksowe analizy zostały prowadzone od 1994 roku przez Standish Group[1]. Pozwoliły one zwymiarować zjawisko (np. okazało się, że tylko 16% projektów można uznać za udane) oraz zdiagnozować jego przyczyny. Problemy z realizacją projektów objawiły się w dwóch podstawowych obszarach: ilościowym oraz jakościowym.

Obszar ilościowy to przede wszystkim podstawowe parametry projektu: czas i budżet, a jakościowy: zgodność wytworzonego produktu końcowego ze specyfikacją wymagań. Niska jakość projektów objawiała się zatem niedotrzymywaniem terminów, zwiększonym w stosunku do planowanego budżetem oraz niezgodnością z wymaganiami produktu końcowego. Udział projektów zakończonych niepowodzeniem ilościowym waha się wokół 70% i powoli maleje[2]. Procent zrealizowanych wymagań w produktach końcowych w latach 1994-2002 niestety też stale malał z ok. 60% do niewiele ponad 50%[3]. Zjawisko kryzysu w obszarze wytwarzania systemów informatycznych nazwano „chaosem".

Reakcją na „chaos" w realizacji projektów informatycznych było naturalne wymaganie wprowadzenia porządku. Bardzo szybko rozwinęły się metodyki inżynierii oprogramowania, formalizujące (a więc porządkujące) artefakty i procesy projektowo-wytwórcze. Początkowo były one tworzone chaotycznie w różnych, rozproszonych ośrodkach by z czasem przejść do fazy standaryzacji (w ramach różnych systemów, takich jak ANSI, ISO czy IEEE, a także ISO/IEC). Oprogramowanie zaczęło być postrzegane jako zwykły produkt inżynierski, taki

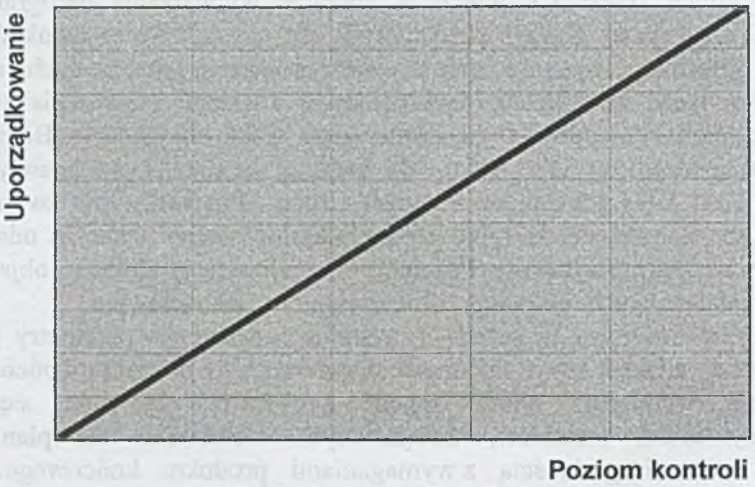
---

[1] *The CHAOS Report (1994).* Standish Group, West Yarmouth, MA, 1995

[2] Czarnacka-Chrobot B.: *Z najnowszych „Kronik Chaosu" Standish Group, czyli czy uczymy się na błędach?* W: *Efektywność zastosowań systemów informatycznych 2004.* Pod red.: Grabary J.K., Nowaka J.S., WNT, Warszawa, 2004

[3] tamże

jak dom czy droga[4]. Równocześnie zaczęto porządkować procesy wytwórcze, wprowadzając systemowe metody zarządzania projektami do procesów wytwarzania oprogramowania. Wierzono, że uporządkowanie procesów (a więc ograniczenie zidentyfikowanego „chaosu" w projektach informatycznych) można osiągnąć poprzez wprowadzenie rygorystycznego i niezależnego nadzoru-kontroli. Zależność (rys. 1) wydaje się prosta: im większy poziom kontroli (formalizmu, sterowania, nadzoru, sprawdzania, śledzenia, reagowania na bieżąco) tym większy poziom uporządkowania (ograniczenia „chaosu"), a w konsekwencji większe prawdopodobieństwo osiągnięcia sukcesu.



Rys. 1. Wyidealizowana zależność pomiędzy poziomem kontroli a uporządkowaniem projektów

1. Taksonomia projektów informatycznych

Pojęcie projekt informatyczny jest pojęciem złożonym i różnie rozumianym. W wąskim jego znaczeniu jest on rozumiany jako wytworzenie oprogramowania, a w szerszym również jego wdrożenie czy nawet np. budowa ogólnopolskiej sieci optycznej.

Do klasyfikacji projektów informatycznych można wykorzystać następujące ich cechy:

- produkt końcowy,
- obszar,
- klient.

[4] *Inżynieria oprogramowania w projekcie informatycznym.* Pod red. Górskiego J., MIKOM, Warszawa, 1999

Produkt końcowy dzieli projekty na:

- projekty dostarczające nowego produktu (wytwórcze) – np. produkcja nowego oprogramowania czy też nowego telefonu komórkowego;
- projekty wdrożeniowe, w których gotowy produkt informatyczny przekazywany jest do eksploatacji – np. wdrożenie systemu ERP.

W praktyce występują także projekty kompleksowe, polegające na wytworzeniu i wdrożeniu systemu informatycznego przez jedną firmę. Zwykle jednak takie projekty są wyraźnie rozdzielane na dwa podprojekty: wytwórczy i wdrożeniowy.

Obszar projektów informatycznych pozwala podzielić je na następujące klasy projektów:

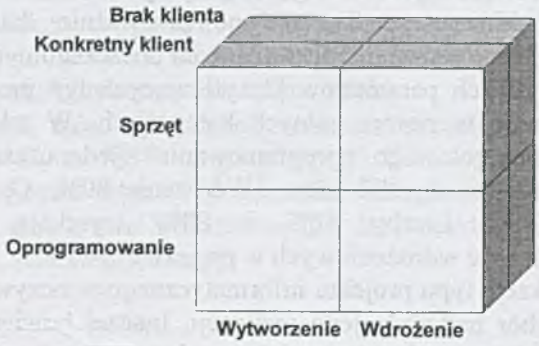
- sprzętowych, w których główna część pracy związana jest z dostarczeniem infrastruktury technicznej (np. budowa węzła sieci LAN) wraz z niezbędnym oprogramowaniem systemowym i pracami konfiguracyjnymi;
- programistycznych, w których część sprzętowa nie jest brana pod uwagę (np. produkcja nowej gry komputerowej czy wdrożenie nowej wersji oprogramowania).

W tym przypadku dopuszczalne są także projekty kompleksowe polegające np. na wdrożeniu oprogramowania przez integratora systemów informatycznych, który odpowiedzialny jest także za budowę infrastruktury technicznej.

Projekty informatyczne mogą być realizowane w różnych warunkach jeśli chodzi o fakt posiadania zdefiniowanego i jasno określonego zewnętrznego klienta, który określa wymagania do produktu końcowego, a także weryfikuje ich realizowalność. W płaszczyźnie posiadania klienta można wydzielić projekty:

- posiadające klienta,
- o niezdefiniowanym kliencie.

Fakt braku zewnętrznego klienta zmienia warunki (a przede wszystkim podmiot) definiowania wymagań. Brak klienta nie jest równoznaczny oczywiście z brakiem sponsora, kierownictwa czy wreszcie zamawiającego projekt.



Rys. 2. Klasyfikacja projektów informatycznych

Na te trzy wymiary klasyfikujące projekty informatyczne (rys. 2) nakłada się jeszcze jeden bardzo ważny: wielkość projektu. Projekty duże (niezależnie od miary użytej do określenia wielkości: wielkość zespołu, pracochłonność czy budżet) są bowiem bardzo odmienne od małych, jeśli je rozpatrywać w aspektach wymagań procesu realizacji i zarządzania tym procesem.

W konsekwencji typ projektu informatycznego (w wyidealizowanym binarnym podejściu) można opisać w następujący sposób:

Typ projektu ← (Produkt, Obszar, Klient, Wielkość)

gdzie:

Produkt = {Wytworzenie, Wdrożenie},
Obszar = {Oprogramowanie, Sprzęt},
Klient = {Konkretny, Brak},
Wielkość = {Duży, Mały}.

Wykorzystując zarysowaną taksonomię projektów informatycznych można zdefiniować najczęściej występujące w praktyce projekty, jako:

- projekty wytwarzania dużego lub małego oprogramowania na zamówienie (tzw. oprogramowanie unikalne) ← (Wytworzenie, Oprogramowanie, Konkretny, *<do zdefiniowania>*);
- projekty wytwarzania dużego lub małego oprogramowania na półkę (tzw. programowanie powielarne lub komercyjne) ← (Wytworzenie, Oprogramowanie, Brak, *< do zdefiniowania >*);
- projekty wdrożenia oprogramowania ← (Wdrożenie, Oprogramowanie, Konkretny, *<do zdefiniowania>*);
- projekty budowy infrastruktury technicznej ← (Wytworzenie, Sprzęt, Konkretny, *<do zdefiniowania>*).

Można oczywiście wykorzystać zaproponowana taksonomię do innej typologii projektów, dzielących je na[5]: projekty MIS (ang. *Management Information Systems*), systemy outsourcingowe, oprogramowanie systemowe, oprogramowanie komercyjne, militarne i jednostkowe, a także projekty webowe.

Projekty informatyczne mogą dotyczyć jednocześnie wielu obszarów czy produktów. W takim przypadku są one przeważnie dzielone na oddzielne podprojekty. Można również definiować udział poszczególnych części w projekcie i z wartości binarnych parametrów klasyfikujących typ projektu przejść na ich udziały procentowe w poszczególnych kategoriach. W takim przypadku duży projekt wdrożenia gotowego oprogramowania z jednoczesną budową centrum przetwarzania można określić jako: (Wdrożenie: 80%, Oprogramowanie: 50%, Konkretny, Duży). Liczby 50% i 80% wynikają z oceny udziału oprogramowaniai prac wdrożeniowych w projekcie.

Identyfikacja typu projektu informatycznego w oczywisty sposób powinna wpływać na wybór metodyki jego realizacji. Inaczej bowiem wygląda struktura organizacyjna oraz zarządzanie projektem dużym i małym, z lub bez klienta,

---

[5] *The Story of Managing Project: A Global, Cross-Disciplinary Collection of Perspectives.* Carayannis E.G., Kwark Y.H. (ed.). Greenwood Press, Quorum Books, 2002.

a także wytwórczym czy wdrożeniowym. Cechy te wpływają także na różnorodność podejścia do stopnia formalizacji procedur planowania, realizacji i rozliczania przedsięwzięć.

2. Klasyczne metodyki zarządzania projektami informatycznymi

Klasyczne metodyki zarządzania projektami informatycznymi realizują podstawowe zadanie: zwiększenie przewidywalności i zarządzalności procesu wytwarzania produktów informatycznych. Większość z nich posiada cechę restrykcyjności, formalizmu i dbania o kontrolę procesów. Kontroli podlega większość procesów zarówno planistycznych, realizacyjnych i rozliczeniowych. Metodyki ukierunkowane są zatem na kontrolowanie (a w zasadzie restrykcyjne zarządzanie) zakresu, czasu, zasobów, ludzi, budżetu, zmian, komunikacji, jakości i ryzyka[6].

Klasyczne metodyki zarządzania projektami historycznie rozwijały się w różnych ośrodkach oraz kierunkach. W konsekwencji można je sklasyfikować jako[7]:

- Metodyki ogólne, tj. takie które znajdują zastosowanie w wielu obszarach aktywności, nie tylko dla projektów informatycznych. Do nich zalicza się przede wszystkim amerykańską metodykę PMI (ang. *Project Management Institute*), brytyjską - APM *(ang. Association for Project Management)*, japońską - P2M (ang. *Project & Program Management for Enterprise Innovation),* a także podejście Kepner-Tregoe'a[8].
- Metodyki firmowe, tj. takie które powstały w laboratoriach poszczególnych firm informatycznych i zostały upublicznione a nawet przyjęte jako standardy. Są to metodyki firmy IBM pod nazwą PMM (ang. *Project Management Methodology*), brytyjska metodyka PRINCE2 (ang. ***PR**ojects IN Controlled Environments*) pierwotnie opracowana dla Central Computer and Telecommunication Agency (CCTA) oraz RUP (ang. *Rational Unified Process*) opracowana przez firmę Rational Software (obecnie IBM). Są one ukierunkowane na zarządzanie projektami informatycznymi, realizowanymi dla konkretnego klienta w pełnym cyklu wytwarzania (przy wykorzystaniu różnych modeli cyklów życia: od kaskadowego do iteracyjnego) i wdrożenia, ze szczególnym uwzględnieniem dużych projektów.
- Metodyki specjalne, które są ukierunkowane na specyficzne typy projektów informatycznych. Są to MSF (ang. *Microsoft Solution Framework*) firmy Microsoft dla projektów typu: (Wytworzenie, Oprogramowanie, Brak, Duży) oraz ASAP (ang. *Accelerated SAP*) firmy SAP dla projektów typu (Wdrożenie, Oprogramowanie, Konkretny, Duży).

---

[6] Szyjewski Z.: *Metodyki zarządzania projektami informatycznymi.* PLACET, Warszawa, 2004

[7] tamże

[8] *Kepner-Tregoe Project Management. Process Guide.* Kepner-Tregoe Inc., 1998

Bardzo duża liczba metodyk zarządzania projektami powstała dla projektów wdrożeniowych[9]. Są one bardzo specyficzne i silnie związane z wdrażanym produktem oraz z końcowym klientem. Są to (przy czym daleko nie wszystkie) następujące metodyki[10]:

- ASAP firmy SAP (system R3 czy mySAP);
- BASIS organizacji SSA;
- AIM firmy IFS wdrażania IFS Applications;
- Implex - Intentia-Vimex (system Movex) ;
- JBA Advantage - JBA (system JBA System 21);
- Q-Managment firmy QAD (system MFG/Pro);
- MEGA - Quantum (system MEGA);
- Epicor i Kewill - Qumak (system Platinum/MMRP);
- Signature firmy Scala (system Scala Global Series);
- Target - TCH Systems (system Baan IV);
- Extract II - Exact Software (system Exact);
- REP - Normax (system JDEdwards);
- Fourth Shift firmy Hogart (system Fourth Shift);
- Microsoft Business Solutions Partner Methodology firmy Microsoft (system MBS Axapta).

Do projektów wdrożeniowych stosowane są także metodyki uniwersalne (ale bardzo rzadko).

Klasyczne metodyki zarządzania projektami charakteryzują się zwykle formalnymi strukturami organizacyjnymi (w większości występują wielopoziomowe, hierarchiczne struktury: od komitetu Sterującego, poprzez Kierownika Projektu do Zespołów Wykonawczych, zwykle uwzględniające specjalne komórki typu Zarządzania Jakością czy Biuro Projektu). Formalizują również procedury związane ze wszystkimi procesami na wszystkich typowych etapach realizacji projektów (rys. 3).

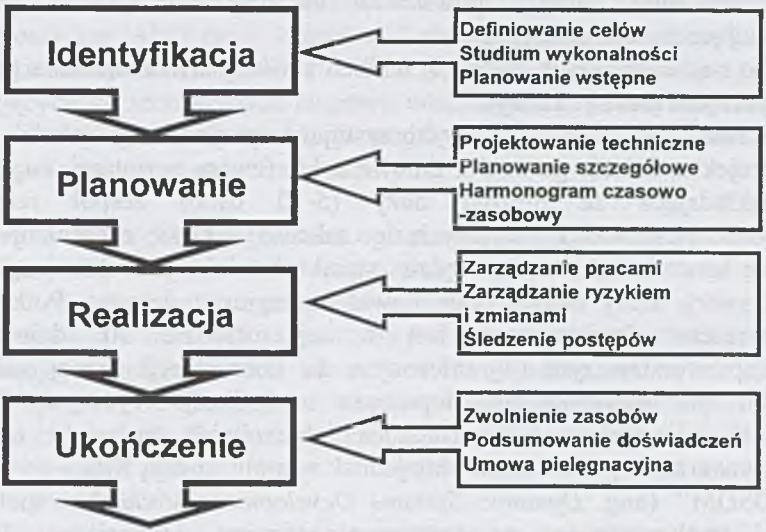
Formalizacja procesów zarządzania projektami, wprowadzona w klasycznych metodykach by ograniczyć chaos poprzez wzmożenie kontroli (rys. 1), postrzegana jest bardzo często jako nadmierna (i zbędna) praca papierkowa (dokumentacyjna), zmniejszenie elastyczności organizacyjno-decyzyjnej (jako naturalny rezultat zbyt precyzyjnego zdefiniowania procesów) oraz ograniczenie zdolności do wprowadzania zmian w trakcie trwania projektu (również jako wynik zbytniej formalizacji procesów). Poza tym, wynikające z formalizmów narzuty na pracochłonność realizacji projektów metodykami

---

[9] *Przedsięwzięcia wdrożeniowe – od teorii do praktyki.* Pod red. Miłosza M. MIKOM, Warszawa, 2003

[10] Miłosz M.: *Wdrażanie systemów informatycznych – problemy i metodyki.* Materiały konferencji: „Systemy informatyczne w ciepłownictwie", Słok k/Bełchatowa, 29-30.11.2004, Izba Gospodarcza Ciepłownictwo Polskie, str. 3-17.

klasycznymi osiąga znaczne wielkości – 10%-30%[11]. Jako reakcja na zbyt restrykcyjne procesy oraz „niepotrzebne" tracenie sił i środków powstały metodyki lekkie w informatyce. Ich istotę wyraża manifest zwinności (ang. *Manifesto for Agile Software Development*)[12].



Rys. 3. Typowe etapy i procesy projektu[13]

3. Lekkie metodyki wytwarzania oprogramowania

Lekkie (niekiedy zwane zwinnymi, ang. *Agile*) metodyki wytwarzania oprogramowania stanowią zaprzeczenie stosowanych uprzednio metod formalnych. Wspomniany manifest[14] wyraża ich istotę i sprowadza się do czterech podstawowych „wytycznych" w procesie produkcji oprogramowania:

- preferowanie indywidualności twórców i bezpośredniej (tj. ustnej, bez dokumentów) komunikacji pomiędzy nimi z jednoczesnym zaprzeczeniem podejścia procesowego i wykorzystania narzędzi go porządkujących;
- pracujące oprogramowanie jest najważniejsze a dokumentacja – nie;
- najważniejsza jest współpraca z klientem w przeciwieństwie do zobowiązań kontraktowych;

[11] Meredith J.R., Mantel S.: *Project Management - A Managerial Approach.* John Wiley&Sons, 2000

[12] http://agilemanifesto.org

[13] *Przedsięwzięcia wdrożeniowe – od teorii do praktyki.* Pod red. Miłosza M. MIKOM, Warszawa, 2003

[14] tamże

- ważna jest szybkość reakcji na zmiany a nie plany i terminowość ich realizacji.

Manifest zwinności był ukierunkowany na projekty typowo programistyczne dla konkretnego klienta, a więc typu: (Wytworzenie, Oprogramowanie, Konkretny, *<do zdefiniowania>*). W wyniku rozprzestrzenienia jego idei powstały metodyki wytwarzania typu Programowanie Ekstremalne (ang. *Extreme Programming, XP*). W ślad za nimi zaczęły powstawać metodyki zarządzania projektami wykorzystujące zwinne podejście.

Do najważniejszych metodyk lekkich (zwinnych) zarządzania projektami informatycznymi można zaliczyć:

- Scrum[15] – metodyka, wykorzystująca przyrostowy model rozwoju projektu, dzieląc go na 30. dniowe cykle (zwane sprintami, ang. *Sprints*), zakładająca że niezbyt duży (5-10 osób) zespół realizatorów (odpowiedzialnych i dążących do sukcesu) sam się zorganizuje i będzie się kontrolował oraz, że będzie istniał właściciel produktu (ang. *Produkt Owner*), który będzie podejmował strategiczne decyzje. Podstawowym okresem planistycznym jest w tej metodzie 30. dniowy cykl, a sprawozdawczym (ograniczonym do komunikacji ustnej zakresie) – 1 dzień. Metodyka nie dopuszcza modyfikacji wymagań w trakcie realizacji sprintu i nie narzuca jakiekolwiek metodyki technicznej wytwarzania produktu (zwykle jednak zespoły stosują XP).
- DSDM[16] (ang. *Dynamic Systems Development Method*) – metodyka ta ukierunkowana jest na dotrzymanie terminu realizacji projektu (przy przyjętym z góry założeniu możliwości płynności wymagań i zakresu). Zakłada ona również aktywny udział klienta oraz pełnię władz decyzyjnych zespołu realizującego (ang. *DSDM Team*), a także aktywne wykorzystanie metod prototypowania, iteracyjności i przyrostowości (główne iteracje dotyczą: modelowania funkcji, budowy rozwiązania oraz jego wdrożenia) procesu dostarczania produktu. Metodyka przewiduje stosowanie zasady „*good enough*" (a właściwie zasady Pareto: 80/20), wyrażającej się w fundamentalnej dla metodyki regule, że dostarczenie produktu na czas jest ważniejsze od jego kompletności (a więc zakresu i jakości).
- Crystal Clear[17] - metodyka, której podstawowym credo jest, że „opracowanie oprogramowania jest zespołową grą innowacyjności i komunikacji". Gra ta wyraża się częstymi dostawami produktu do użytkownika (dostawy bazują na cyklu miesięcznym, tj. 30. dniowym), wykorzystaniu dotychczasowych doświadczeń (podejście adaptacyjne, wykorzystujące refleksję nad dotychczasowym przebiegiem prac) przy

[15] Schwaber K.: *Sprawne zarządzanie projektami metodą Scrum.* Microsoft, Warszawa, 2005.

[16] http://www.dsdm.org/

[17] Cockburn A.: *Crystal Clear : A Human-Powered Methodology for Small Teams.* Addison-Wesley, 2004

produkcji następnych przyrostów, permanentnej nieskrępowanej komunikacji w zespole (notabene niezbyt licznym) przy zapewnieniu bezpieczeństwa osobistego członkom zespołu oraz swobodnego dostępu do ekspertów. Metodyka zakłada również aktywne wykorzystanie narzędzi CASE, takich jak automatyczne testowanie. Wprowadza zarządzanie konfiguracją powstającego oprogramowania oraz częste iteracje.

Inne, mniej znane metodyki lekkie, takie jak FDD (ang. *Future-Driven Development*) czy ASD (ang. *Adaptive Software Development*), mają podobne zasady (ukierunkowanie na produkt oraz jego zmianę), modele iteracyjne (wiele powtarzających się identycznych etapów), adaptacyjne (wykorzystanie na bieżąco zdobytej wiedzy w trakcie realizacji projektu) oraz ukierunkowanie na małe, dynamiczne i samoorganizujące się zespoły.

Twórcy wielu metodyk zwinnych sami ograniczają ich stosowalność do projektów:

- elastycznych w obszarze procesowym,
- realizowanych przy pomocy niewielkiego zespołu,
- o znanych w ogólnych zarysach na początku wymaganiach do produktu.

Słabe strony metodyk lekkich są oczywiste. Ograniczone są do projektów o niewielkiej skali, realizowanych bardzo dynamicznie (a więc w małym horyzoncie czasowym). Dostarczony produkt nie posiada dokumentacji a jedynie spełnia wymagania klienta, który to zresztą musi być silnie zaangażowany w proces wytwórczy.

Niewielka skala projektu, zarządzanego metodyką lekką, wynika z założenia realizacji go przez niewielkie zespoły (zwykle do 10 osób) w dość krótkich cyklach wykonawczych. Trudno bowiem zorganizować elastyczne i dynamicznie, samoorganizujące się bardzo wieloosobowe zespoły. Poza tym dynamika procesu oraz jego adaptacyjność silnie ogranicza horyzont planowania projektu zwinnego. W niektórych przypadkach brak długookresowego planowania jest niedopuszczalny lub nieakceptowalny przez klienta, który mimo wszystko (tj. pomimo stałej współpracy z twórcami systemu oraz bieżących konsultacji) chciałby wiedzieć na kiedy i co będzie zrealizowane.

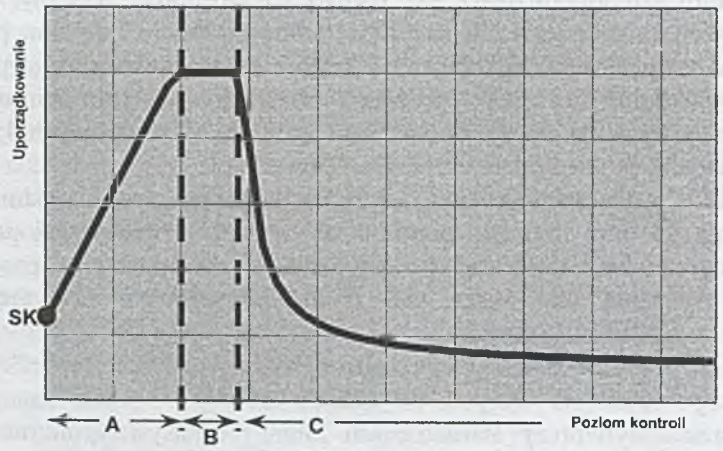
Brak dokumentacji produktu może w znacznym stopniu podnieść koszty serwisowania go oraz rozwoju, pomimo stosowanej w większości przypadków refaktoryzacji kodu. W niektórych przypadkach dokumentacja produktu ma kolosalne znaczenie lub wręcz jest wymagana prawem czy też dobrymi praktykami.

Istotnym problemem występującym w wielu projektach jest brak spełnienia podstawowego założenie wszystkich lekkich metodyk – duże zaangażowanie klienta w proces wytwórczy. Bardzo często klient jest pasywny, nie ma czasu czy jest wręcz przeciwnikiem realizacji projektu. Ustawiczne konsultowanie dostarczanie do testowania coraz to nowych przyrostów w takich warunkach nie jest możliwa. Klient nie kontroluje procesu wytwórczego, nie uczestniczy w nim i w konsekwencji projekt żyje własnym życiem, niekoniecznie spełniając wymagania klienta (które pozostają ukryte, nierozpoznane lub też błędnie zidentyfikowane).

4. Pomiędzy kontrolą a jej brakiem

Metodyki lekkie zakładają, że zespoły realizujące projekt pod okiem bardziej mentora niż kierownika (np. Scrum master'a lub Trenera z XP, ang. *Coach*) same zorganizują sobie pracę i będą kontrolować jej wykonanie bez procesów formalnej kontroli. To założenie jest zgodne z doświadczeniem i oczekiwaniami – zdeterminowani i odpowiednio zdopingowani członkowie zespołów, realizujących ciekawe prace w krótkich cyklach nie muszą być w ogóle kontrolowani. Samokontrola powoduje, że w projektach, pomimo braku kontroli zewnętrznej, jest pewien poziom uporządkowania (wartość SK na rys. 4). Poziom ten zwiększa się poprzez podkreślenie (często stosowane w metodykach lekkich) indywidualności członka zespołu i jego unikalności, a także przypisanie mu (lub parze) „własności" realizowanego fragmentu kodu.

Możliwe jest podniesienie poziomu uporządkowania w projekcie ponad wartość SK, poprzez wprowadzanie elementów kontroli (procesów: planowania, wykonania, rozliczenia) ale wydaje się, że tylko do pewnego momentu – obszar A na rys. 4. Po przekroczeniu pewnego progu wprowadzanie kolejnych procedur kontroli nie powoduje wzrostu uporządkowania (obszar B na rys. 4). Dalsze wzmaganie procesów kontroli powoduje wręcz spadek porządku w projekcie (obszar C). Nadmiar procedur kontrolnych doprowadza bowiem do sytuacji, że osoby je realizujące mają zbyt mało czasu, dane kontrolne są niedokładne, pojawiają się przekłamania lub wręcz fałszerstwa. Projekt na „papierze" wygląda całkiem inaczej (zwykle „idealnie", zgodnie z oczekiwaniami służb kontrolnych) niźli rzeczywistość.



Rys. 4. Realna zależność pomiędzy poziomem kontroli a uporządkowaniem projektów[18]

[18] na podstawie: *Agile Project Management*. CC Pace Systems, 2003, (http://www.ccpace.com/Resources/documents/AgileProjectManagement.pdf)

We wszystkich projektach, w których do utrzymania porządku nie wystarcza samokontrola zespołu (np. gdy niezbędny jest większy zakres dokumentowania prac niźli notatki czy komentarze robione samoistnie przez twórców) niezbędne jest zwiększenie kontroli. Służą zwykle do tego metodyki łączące lekkie podejście i formalne. Są to:

- AUP (ang. *Agile Unified Process*)[19] – metodyka wzorowana na RUP (ang. *Rational Unified Process*), w której zrezygnowano z części artefaktów projektowych, dyscyplin i procesów, a także ról członków zespołu zachowując przy tym iteracyjność procesu, jego 4. etapowość oraz zapewnienie kontroli jakości dostarczanego produktu poprzez testowanie (TDD, ang. *Test Driven Development*);
- XPrince (ang. *Extreme Programming in Controlled Environments*)[20] – metodyka opracowana przez prof. J.Nawrockiego z Politechniki Poznańskiej, powstała w wyniku twórczego połączenia XP i PRINCE2. Jest ona metodyką lekką ale zawiera także elementy kontroli przedsięwzięcia (jakość, ryzyko, praca) oraz niezbędny poziom wymaganej dokumentacji (z wykorzystaniem niektórych diagramów UML). W XPrince struktura organizacyjna projektu łączy zalety obu metodyk, zapewniając nadzór nad projektem przy minimalnych kosztach.

Metodyki te znajdują zastosowanie w projektach nie tylko wytwarzania oprogramowania.

5. Wdrożenie lekkich metodyk zarządzania projektami

Wiele firm komputerowych próbuje wykorzystać lekkie metodyki zarządzania projektami informatycznymi w swojej działalności modyfikując tradycyjny system zarządzania projektami informatycznymi. Proces zmiany metodyki w skali całej firmy jest dużym wyzwaniem oraz sprzyja popełnianiu całego szeregu pomyłek – błędów, zwiększających koszty adaptacji oraz doprowadzających do zwiększonych trudności na etapie przejściowym. Zidentyfikowano 10 takich typowych błędów na etapie wdrożenia metodyk lekkich[21]:

1. Jednorazowe przejście całej firmy na nową metodykę.
2. Szybkie przejście by szybko uzyskać rezultaty.
3. Zapominanie o kulturze korporacyjnej.
4. Błędna identyfikacja Sponsora projektu.
5. Złe przypisanie ról w zespole.
6. Brak planowania.
7. Zbyt szybkie tworzenie zespołów.

---

[19] Si Alhir S.: The Agile Unified Process (AUP). 2005 (http://home.comcast.net/~salhir/TheAgileUnifiedProcess.PDF)

[20] http://xprince.net/xprince

[21] Gurses L.: *10 Mistakes In Transitioning to Agile. Slow down the transition in order to go fast.* Dr. Dobb's Portal, 2006 (http://www.ddj.com/dept/architect/193402902)

8. Natychmiastowe pozbycie się wszelkich narzędzi wspomagających zarządzanie projektami i projektowanie systemów.
9. Zły wybór kluczowych osób w zespołach.
10. Ewangelizacja nowych metod.

Powyższe błędy wynikają z trzech przyczyn:

A. Proces i sposób wprowadzania zmiany.
B. Właściwość metodyk zarządzania projektami.
C. Właściwości metodyk lekkich.

Typowe problemy, związane z samym procesem wprowadzaniem zmiany (A) w organizacji pracy w firmie programistycznej, to:

- przejście jednorazowe (masowe) we wszystkich projektach realizowanych w firmie, bez pilotażu, bez uczenia się na błędach, bez możliwości przyzwyczajenia pracowników do nowych metod pracy (błąd nr 1);
- natychmiastowe przejście do nowych metod bez uwzględnienia stanu zaawansowania dotychczas realizowanych projektów, zbyt szybkie i niewłaściwie przygotowane, bez szczegółowego definiowania ról i zadań, bez szkoleń pracowników, bez wymaganych zmian w strukturze organizacyjnej firmy (błąd nr 2);
- łamanie dotychczasowej kultury korporacyjnej firmy, tj. istniejących metod, a także działów je realizujących – np. oddział planowania, rozwoju czy logistyki wraz z historycznie ustanowionymi barierami pomiędzy oddziałami, ich wewnętrzną polityką oraz interesami kierownictwa (błąd nr 3);
- szybkie tworzenie zespołów (swoiste dla metodyk lekkich) w sytuacji braku możliwości właściwego wykorzystania ich członków – podejście takie rodzi frustracje, brak zadań dla poszczególnych członków zespołu i w konsekwencji stworzenie złej opinii dla metodyk lekkich (błąd nr 7);
- szybkie zarzucenie dotychczasowych technik, narzędzi wspomagania zarządzania projektami i projektowania systemów bez dostatecznie długiego czasu na wprowadzenie tej zmiany (błąd nr 8);
- totalne zanegowanie wartości dotychczas stosowanych (przez wiele lat, zwykle z częściowym sukcesem) metod i ewangelizacja nowości BEZ widocznych jeszcze pozytywnych efektów stosowania nowych metod (błąd nr 10).

Do błędów, związanych z właściwościami (B) i wdrażaniem w praktyce metodyk zarządzania projektami (niezależnie do jakiej grupy się je zalicza: lekkich czy formalnych), można zaliczyć błędną identyfikację Sponsora (błąd nr 4), zły podział ról w zespole (błąd nr 5) oraz zły wybór osób kluczowych (błąd nr 9). Są to wielokrotnie popełniane, zdiagnozowane „typowe" błędy realizacji projektów[22].

Z powyższych rozważań wynika, że właściwie jedynym problemem związanym z wdrożeniem lekkich metodyk zarządzania projektami

[22] Szyjewski Z.: *Metodyki zarządzania projektami informatycznymi.* PLACET, Warszawa, 2004

w organizacjach są problemy wynikające z zarzuceniem długofalowego planowania projektu. Jest to dość istotny obszar zarządzania projektem, przy czym często związany z podmiotem zamawiającym projekt, a więc nie wynikającym tylko z działań wdrożeniowych.

Niezależnie od zidentyfikowanych problemów, wdrożenie lekkich (lub mieszanych) metodyk zarządzania projektami informatycznymi jest opłacalne. Jak pokazują badania[23] (przeprowadzone niestety wśród małych i średnich), ale dla ponad 700 firm z całego świata) zastosowanie metodyk lekkich doprowadza do:

- wzrostu odsetka projektów zakończonych sukcesem: metody lekkie – 81% *versus* formalne – 29%;
- przyśpieszenia wejścia na rynek z oprogramowaniem (w 86% przypadków), zwiększenia produkcyjności (87%), zmniejszenia liczby błędów (86%) oraz zmniejszenia kosztów wytworzenia (63%).

Wg tych samych badań najczęściej stosowana jest metodyka Scrum (w 40% przypadków), XP (23%), własne mieszane metody (14%) oraz DSDM (8%). Przy czym średni czas wykorzystywania metodyk lekkich przez badane firmy wynosił niespełna 2 lata.

Podsumowanie

Zwiększenie formalizacji metodyk zarządzania projektami informatycznych, które miało być panaceum na kryzys oprogramowania i „chaos" w procesach realizacji systemów informatycznych, nie przyniosło spektakularnych sukcesów. Formalizacja metodyk (w tym i wymóg szczegółowego dokumentowania oraz kontroli procesów) wywołała reakcję: zbyt ciężko, za dużo, niepotrzebnie (szczególnie dla twórców). W konsekwencji pojawiła się idea zwinności – *agile*.

Idea zwinności początkowo propagowana wśród programistów zaczęła rozwijać się na obszary związane z zarządzaniem projektami. Rozwój ten doprowadził do powstania całego szeregu lekkich metodyk zarządzania projektami, coraz szerzej i coraz bardziej efektywnie wdrażanych w praktykę (nie bez typowych problemów wdrożeniowych).

Należy zwrócić uwagę, że „stopień" lekkości metodyki powinien być dostosowany do typu projektu, realizowanego przy jej pomocy. Niektóre projekty niedopuszczają bowiem do nadmiernego „odciążenia". Wymagają bowiem dostarczenia odpowiedniej dokumentacji, zaplanowania działań, ich realizacji zgodnie z planem etc. W sukurs przychodzą w tym zakresie metodyki mieszane, łączące w sobie zalety metodyk formalnych i lekkich.

---

[23] *The State of Agile Development.* VersionOne LCC, 2006 (http://www.versionone.net/surveyresults.asp)

Literatura

1. *Agile Project Management*. CC Pace Systems, 2003 (http://www.ccpace.com/Resources/documents/AgileProjectManagement.pdf)
2. Cockburn A.: *Crystal Clear : A Human-Powered Methodology for Small Teams*. Addison-Wesley, 2004
3. Czarnacka-Chrobot B.: *Z najnowszych „Kronik Chaosu" Standish Group, czyli czy uczymy się na błędach?* W: Efektywność zastosowań systemów informatycznych 2004. Pod red.: Grabary J.K., Nowaka J.S., WNT, Warszawa, 2004
4. Gurses L.: *10 Mistakes In Transitioning to Agile. Slow down the transition in order to go fast*. Dr. Dobb's Portal, 2006 (http://www.ddj.com/dept/architect/193402902)
5. http://agilemanifesto.org
6. http://www.dsdm.org/
7. http://xprince.net/xprince
8. *Inżynieria oprogramowania w projekcie informatycznym*. Pod red. Górskiego J., MIKOM, Warszawa, 1999
9. *Kepner-Tregoe Project Management. Process Guide*. Kepner-Tregoe Inc., 1998
10. Meredith J.R., Mantel S.: *Project Management - A Managerial Approach*. John Wiley&Sons, 2000
11. Miłosz M.: *Wdrażanie systemów informatycznych – problemy i metodyki*. Materiały konferencji: „Systemy informatyczne w ciepłownictwie", Słok k/Bełchatowa, 29-30.11.2004, Izba Gospodarcza Ciepłownictwo Polskie, str. 3-17.
12. *Przedsięwzięcia wdrożeniowe – od teorii do praktyki*. Pod red. Miłosza M., MIKOM, Warszawa, 2003
13. Schwaber K.: *Sprawne zarządzanie projektami metodą Scrum*. Microsoft, Warszawa, 2005.
14. Si Alhir S.: *The Agile Unified Process (AUP)*. 2005 (http://home.comcast.net/~salhir/TheAgileUnifiedProcess.PDF)
15. Szyjewski Z.: *Metodyki zarządzania projektami informatycznymi*. PLACET, Warszawa, 2004
16. *The CHAOS Report (1994)*. Standish Group, West Yarmouth, MA, 1995
17. *The State of Agile Development*. VersionOne LCC, 2006 (http://www.versionone.net/surveyresults.asp)
18. *The Story of Managing Project: A Global, Cross-Disciplinary Collection of Perspectives*. Carayannis E.G., Kwark Y.H. (ed.). Greenwood Press, Quorum Books, 2002.

ROZDZIAŁ II

ZARZĄDZANIE WYMAGANIAMI W PROCESIE WYTWÓRCZYM OPROGRAMOWANIA

Grzegorz BLIŹNIUK, Anna ŻAK

Wstęp

Gwałtowny postęp technologiczny wiąże się z coraz większym zapotrzebowaniem na nowe, efektywne i spełniające określone funkcje oprogramowanie. Obecnie większość firm na rynku inwestuje w rozwiązania informatyczne w celu zwiększenia swojej wydajności, a co za tym idzie powiększenia zysków. Głównym celem wytwórców oprogramowania jest tworzenie takich produktów, który w pełni usatysfakcjonują klienta. Zadowolenie zleceniodawcy jest ściśle powiązane z funkcjonalnością sytemu oraz z czasem i kosztami wykonania projektu. Dlatego też zarządzanie wymaganiami jest tak istotnym elementem. Można nawet pokusić się o stwierdzenie, że jest to najważniejszy element cyklu wytwórczego oprogramowania, ponieważ wszelkiego rodzaju błędy popełnione w tym zakresie będą miały swoje surowe konsekwencje w późniejszych stadiach realizacji projektu.

Zgodnie z raportem *The Standish Group*[1] liczba projektów zakończonych sukcesem wzrasta w ostatnich latach, ale jest to nadal ok. 30% wszystkich rozpoczętych przedsięwzięć. Nadal wiele projektów zostaje zakończonych dużo później niż planowano, a inne upadają z powodu przekroczenia budżetu. Dlatego, aby uniknąć takich problemów trzeba dobrze zdefiniować wymagania na tworzony system. Definicja wymagań ma kluczowy wpływ na to, jakie funkcje będzie realizować przyszła aplikacja. Największym wyzwaniem jest udoskonalanie zdefiniowanych wymagań i dążenie do osiągnięcia pełnej funkcjonalności systemu. Kompletne wymagania są dla zespołu gwarancją zapewniającą zbudowanie systemu dobrej jakości w możliwie jak najkrótszym czasie.

Obecnie powstało wiele standardów wspomagających zarządzanie wymaganiami. Pojawiły się narzędzia zapewniające lepsza komunikację i redukujące ryzyko projektu, pozwalające dokumentować wymagania i zarządzać nimi. Należy jednak nadmienić, że nadal stosunkowo często ten element projektowania systemu jest wykonywany niedokładnie, w pośpiechu, a to pociąga za sobą konsekwencje w postaci niekompletnych, błędnych wymagań, braku znajomości dziedziny rozwiązywanego problemu i trudności wprowadzania jakichkolwiek zmian. Koszty naprawy takich błędów wzrastają w miarę postępowania projektu w kolejnych fazach realizacji przedsięwzięcia i dlatego lepiej jest, w miarę możliwości, wydobyć je wszystkie już na początku – działa

[1] The Standish Group "Chaos Report" (2000)

tutaj bowiem zjawisko piramidy propagacji błędów[2]. Jeśli się to nie uda to istnieje zagrożenie, że czas poświęcony pracom wdrożeniowym, implementacji, czy też testom zostanie zmarnowany, gdyż będzie on poświęcony błędnie zdefiniowanym elementom.

Wykrywanie błędów w oprogramowaniu i konieczność ich usuwania może również prowadzić do konieczności modyfikacji wymagań na oprogramowanie. W takiej sytuacji ważne jest dopilnowanie, aby każdy członek zespołu wytwórczego był należycie informowany o wszelkich korektach założeń, aby nie wprowadzano w życie produktu w oparciu o nieaktualne specyfikacje, co wiąże się z wysokimi kosztami i opóźnieniami w realizacji. Przy dobrym zarządzaniu wymaganiami taka sytuacja nie będzie miała miejsca, a powodzenie projektu nie będzie zagrożone.

Istota wymagań na system informatyczny

Jakość zdefiniowanych wymagań decyduje de'facto o tym, jakie funkcje system będzie realizował i czy będzie to rozwiązanie zgodne z oczekiwaniami klienta. Konieczne jest zatem poświęcenie należytej uwagi procesowi definiowania wymagań. Można z dużym prawdopodobieństwem przypuszczać, że im bardziej wymagania będą kompletne, a później zarządzane, tym lepsze będzie ostateczne rozwiązanie dostarczone klientowi.

Punktem wyjściowym w projektowaniu w pełni funkcjonalnej aplikacji są trzy podstawowe elementy związane z inżynierią wymagań, czyli:

- zebranie wymagań użytkowników,
- zarządzanie tymi wymaganiami, czyli zestawienie powstałej specyfikacji z oczekiwaniami klienta,
- weryfikowanie tych wymagań poprzez stałą komunikację całego zespołu, aby upewnić się, że powstaje odpowiednia aplikacja.

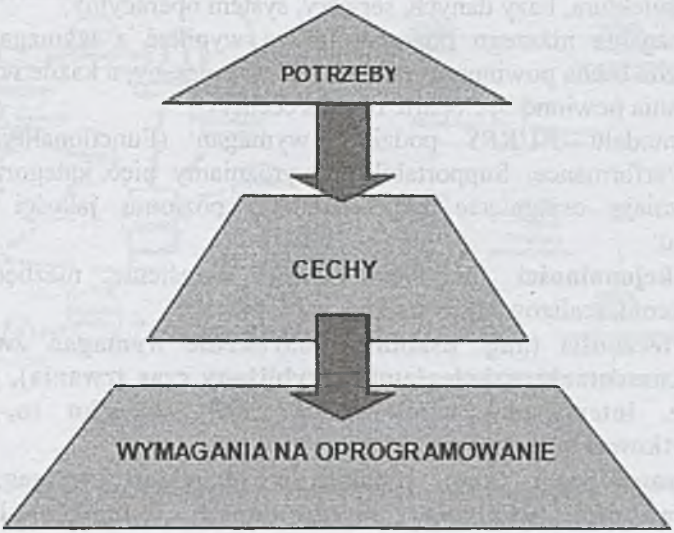
W literaturze można spotkać się z wieloma definicjami wymagań[3]. Jednak wszystkie dają pewną wizję, czym zajmuje się ta dziedzina inżynierii oprogramowania.

Def.1: *Wymaganie na oprogramowanie jest to zlokalizowana możliwość rozwiązania konkretnego problemu i osiągnięcia określonego celu, oczekiwana przez użytkownika, co wiąże się z możliwością spełnienia umowy, normy, specyfikacji lub innej narzuconej dokumentacji systemu lub jego komponentu.*

Def.2: *Zarządzanie wymaganiami jest to systematyczne podejście do uzyskiwania, organizowania i dokumentowania wymagań na system, co jest powiązane z procesem spełnienia umowy pomiędzy klientem i*

[2] Bliźniuk G. , „Badanie jakości programów współbieżnych", Warszawa, 2004
[3] Leffingwell Dean, Widrig Don „Zarządzanie wymaganiami" (WNT Warszawa, 2003)

zespołem realizującym przedsięwzięcie w zależności od zmieniających się wymagań na system".



Rys. 1. Rodzaje wymagań

Projekt powinien odnosić się do trzech poziomów wymagań pochodzących z różnych etapów i różnych źródeł[4].

1) **Wymagania biznesowe (potrzeby)** opisujące powód, dla którego tworzony jest system i pokazujące zalety płynące z realizacji przedsięwzięcia zarówno dla klienta, jak i dla otoczenia biznesowego. Wymagania te definiowane są na samym początku realizacji przedsięwzięcia.
2) **Wymagania użytkownika (cechy)** opisują zadania lub procesy biznesowe, które użytkownik będzie mógł realizować przy użyciu tworzonego produktu. Przedstawione one są za pomocą prostych opisów i służą do komunikacji z użytkownikiem, w celu ustalenia sposobów realizacji zadań przez system.
3) **Wymagania systemowe** (wymagania stawiane oprogramowaniu)
   a. **Funkcjonalne:** opis zachowania systemu, które muszą być zaimplementowane. Zaliczamy tu realizowane funkcje, ograniczenia, poprawność danych, informacje o błędach, wymagania instalacyjne (ilość pamięci, katalog instalacji itd.), powiązania pomiędzy poszczególnymi obiektami. Zgodnie z definicją spotykaną w literaturze wymagani funkcjonalne to *„cała funkcjonalność, którą produkt musi*

[4] Wiegers Karl E. "When Telepathy Won't Do: Requirements Engineering Key Practices" (2000)

realizować, jeśli jest ona użyteczna z punktu widzenia klienta"[5]. Wymagania te muszą być zgodne z zakresem projektu.

b. **Niefunkcjonalne**: opis warunków, w jakich system będzie działał, tzn. architektura, bazy danych, serwery, system operacyjny.

Wymagania niższego poziomu muszą wynikać z wymagań wyższego poziomu. Każda cecha powinna wynikać z jakiejś potrzeby, a każde wymagania na oprogramowania powinno być oparte o jakąś cechę.

W modelu ***FURPS*** podziału wymagań (Functionality, Usability, Reliability, Performance, Supportability) wyróżniamy pięć kategorii wymagań, które zapewniają osiągnięcie odpowiedniego poziomu jakości tworzonego systemu. Są to:

- **funkcjonalności** (ang.functionality): określenie niezbędnych cech systemu, realizowanych funkcji, zabezpieczeń
- **użyteczności (ang. usability): określenie wymagań związanych z dokumentacją, szkoleniami (przybliżony czas trwania), pomocą on-line, interfejsami użytkownika, czyli wszystko to, co ułatwi użytkowanie systemu**
- **niezawodności (ang. reliability): określenie wymagań, co do sprawności działania, niezawodności, odzyskiwania danych, dostepności**
- **wydajności (ang. performance): określenie wymagań, co do czasu wykonywania zadań, szybkości odpowiedzi**
- **utrzymywalności, pielęgnacyjności (ang. supportability): określenie wymagań** dotyczących testowania, ewentualnego rozbudowywania, konserwacji, konfiguracji.

W rozszerzeniu modelu FURPS zwanym ***FURPS+*** dodano:

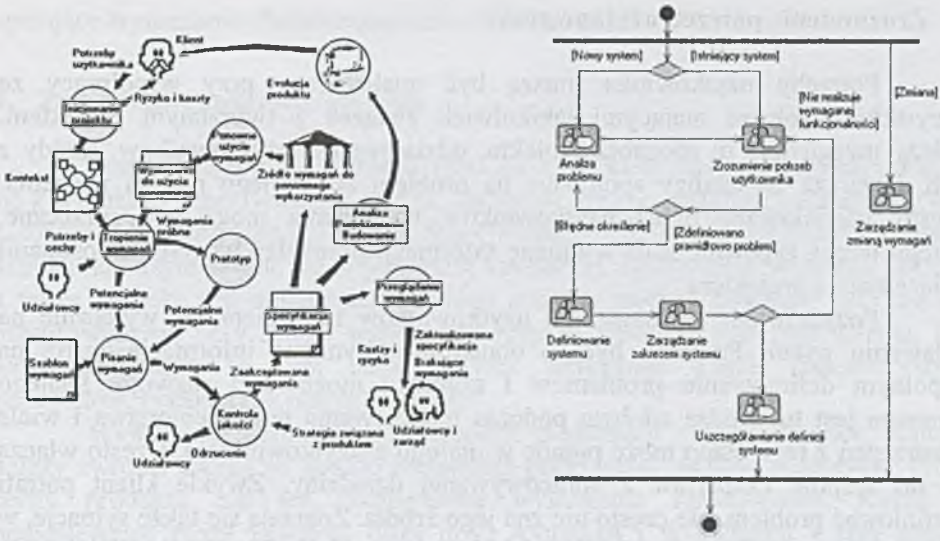
- **wymaganie implementacyjne:** wymagane standardy, język implementacji, bazy danych, system operacyjny i inny sprzęt.
- **wymagania fizyczne:** np. wymagane wolumeny danych.
- **wymagania interfejsu:** zewnętrzne elementy, z którymi system musi współpracować, limity czasowe, formaty.

Przykłady narzędzi do wspomagania zarządzania wymaganiami

Każdy tworzony projekt jest uzależniony od wymagań, których liczba w przypadku większych przedsięwzięć może być znacząca. W takiej sytuacji potrzebne są techniki i określone procedury postępowania, aby wszystkie te elementy tworzyły spójną i logiczną całość. W zastosowaniu są różnego rodzaju podejścia takie jak metodyka RUP, czy też podejście Volere.

---

[5] Robertson Suzanne, Robertson James "Mastering the requirements process" (ACM Press, 1999)



Rys. 2. Etapy pracy z wymaganiami w Volere[6](lewa część rysunku) i RUP[7] (część prawa)

1. Analiza problemu

Pierwszym krokiem w analizie problemu jest jego identyfikacja. Należy zweryfikować czy rzeczywiście on istnieje i czy może on być w jakiś sposób rozwiązany. Ważnym aspektem jest tu punkt widzenia użytkownika i jego postrzeganie problemu. Rozwiązanie problemu jest bezpośrednio związane z jego przyczyną, dlatego też trzeba znaleźć jego źródło. Do tego celu można wykorzystać tzw. diagram szkieletowy, na którym każda z „ości" stanowi potencjalne źródło problemu. Oczywiście nie rozwiązuje się wszystkich problemów wynikłych w czasie takiej analizy. Niezbędne jest więc nadanie priorytetów poszczególnym elementom. Uzupełnieniem może być tutaj diagram Paretto pokazujący zależności pomiędzy problemem, a jego znaczeniem. Na podstawie takich analiz można określić, czy rozwiązanie takiego problemu jest opłacalne. Kolejnym krokiem na tym etapie jest wyznaczenie udziałowców i użytkowników systemu. Oprócz zleceniodawcy musimy uwzględnić wszystkich ludzi, którzy będą w przyszłości korzystać z systemu lub, na których system będzie miał wpływ. Uwzględnia się tutaj nie tylko osoby, ale także inne obiekty, które wymieniają dane z systemem np. inne systemy, urządzenia. Im dokładniej określi się obiekty związane z tworzonym systemem, tym lepiej zostaną zdefiniowane wymagania i wynik końcowy będzie bardziej zgodny z oczekiwaniami klienta.

[6] Robertson Suzanne, Robertson James "Mastering the requirements process" (ACM Press, 1999)

[7] IBM Corporation "Rational Unified Process" (RUP, 2003)

2. Zrozumienie potrzeb użytkownika

Potrzeby użytkownika muszą być analizowane przy współpracy ze wszystkimi osobami mającymi jakikolwiek związek z tworzonym projektem. Należy uwzględnić tu sponsora projektu, udziałowców, użytkowników. Każdy z nich dostarcza do analizy spojrzenie na problem ze swojego punktu widzenia. Często zdefiniowane przez użytkowników wymagania mogą być sprzeczne, dlatego trzeba zapewnić stałą wymianę informacji pomiędzy wszystkimi osobami związanymi z projektem.

Pozyskiwanie wymagań od użytkowników nie opiera się wyłącznie na zadawaniu pytań. Powinna być to obustronna wymiana informacji oparta na wspólnym definiowaniu problemów i szukaniu możliwości rozwoju. Dlatego pomocna jest tu wiedza zdobyta podczas modelowania przedsiębiorstwa i wiele spostrzeżeń z tego etapu może pomóc w dialogu z użytkownikiem. Często włącza się do zespołu ekspertów z opracowywanej dziedziny. Zwykle klient potrafi zdefiniować problem, ale często nie zna jego źródła. Zdarzają się także sytuacje, w których klient sam nie wie dokładnie jaki końcowy wynik chce uzyskać lub też nie potrafi tego wyrazić. Pomocnym elementem na tym etapie są różnego rodzaju modele w postaci diagramów, tabel, przypadków użycia, które umożliwiają komunikację z użytkownikami i odzwierciedlają zrozumienie problemu przez analityka. Na ich podstawie użytkownik wskazuje elementy, których jeszcze nie uwzględniono lub źle zinterpretowano. Dlatego też stosuje się różnego rodzaju metody wspomagające ten proces takie jak wywiady, burze mózgów, warsztaty czy definiowanie przypadków użycia.

3. Definiowanie systemu

Duża liczba wymagań zebranych podczas prac z użytkownikiem wymaga dobrego zorganizowania, aby potem było łatwiej nimi zarządzać. Podejście takie ułatwia wychwytywanie ewentualnych błędów oraz pozwala lepiej widzieć zależności pomiędzy poszczególnymi wymaganiami. Wymagania powinny być zebrane w taki sposób, aby łatwo było znaleźć określoną informację w dokumentacji, dlatego powinno się je pogrupować według różnych poziomów szczegółowości. Często budowana aplikacja składa się z wielu modułów, podsystemów i z reguły dla każdego z nich definiuje się oddzielnie wymagania uwzględniając przy tym powiązania pomiędzy poszczególnymi elementami. Wymagania zbiera się w różnego rodzaju dokumentach, których ilość zależy od charakteru tworzonego systemu i jego złożoności. Ogólne potrzeby dotyczące tworzonego systemu, czyli wyjście do bardziej szczegółowych wymagań opisuje dokument wizji. W każdym z przypadków powstaje także specyfikacja wymagań systemu zawierająca wszystkie zebrane informacje odnośnie wymagań, co do osób, sprzętu i realizowanych funkcji. Z reguły w dużych przedsięwzięciach taki dokument tworzy się oddzielnie dla każdego z modułów. W przypadku tworzenia aplikacji z pewnej rodziny produktów dodatkowo definiuje się dokumenty

opisujące wymagania charakterystyczne dla wszystkich systemów.

4. Zarządzanie zakresem systemu

Zespół projektowy w czasie prac nad projektem musi najczęściej zmagać się z wysokimi oczekiwaniami klienta, napiętym harmonogramem i ograniczonymi zasobami[8]. Dostarczenie określonej funkcjonalności w takich warunkach wymaga decyzji nad czym skupić się w pierwszej kolejności, co z punktu widzenia klienta, a także zespołu jest najważniejsze. Tego rodzaju analizy powinny być zawsze przeprowadzane przy ścisłej współpracy zespołu projektowego oraz klienta. Do głównych zadań na tym etapie należy określenie trzech charakterystyk: priorytetów, ryzyka oraz pracochłonności.

Wymagania definiuje się na wielu poziomach szczegółowości, dlatego należy zwrócić uwagę na sposób nadawania priorytetów. Można analizować tylko cechy produktu lub też szczegółowe wymagania funkcjonalne. Każde z przedsięwzięć może zawierać dużą ilość wymagań i w związku z tym często priorytety nadaje się tylko na poziomie cech, a następnie przyporządkowuje się odpowiednie priorytety zależnym od nich wymaganiom. W przypadku większych przedsięwzięć niezbędne są metody szacowania, które ułatwiają określenie istotności danego elementu.

Kolejnym krokiem jest określenie nakładów pracy, jaki należy ponieść w realizacji każdego z wymagań. Zespół projektowy powinien umieć oszacować trudność zaimplementowania każdego elementu. Jest to dość trudne zadanie na tym etapie, ponieważ nie ma jeszcze zdefiniowanych szczegółowych wymagań. Dlatego też istotną rolę odgrywa tu doświadczenie zespołu. Podobnie jak w przypadku priorytetów wysiłek określa się na jednym z trzech poziomów: wysoki, średni, niski.

Requirements	Priorytet	Status	Zmienialnosc	Ryzyko	Author	Location
FEAT1: Rejestracja uzytkownika	Sredni	Zatwierdzone	Mala	Duze	mania	Vision
FEAT2: Przegląd listy połączeń	Wysoki	Propozycja	Mala	Srednie	mania	Vision
FEAT3: Codzienna aktualizacja lokalnej bazy danych	Wysoki	Propozycja	Duza	Duze	mania	Vision
FEAT4: Uwierzytelnianie użytkowników systemu	Wysoki	Zatwierdzone	Mala	Duze	mania	Vision
FEAT5: Zakup biletu	Wysoki	Zatwierdzone	Duza	Duze	mania	Vision
FEAT6: Anulowanie zakupionego biletu	Sredni	Zatwierdzone	Srednia	Srednie	mania	Vision
FEAT7: Generowanie listy rezerwacji dla danego połączenia	Sredni	Propozycja	Srednia	Srednie	mania	Vision
FEAT8: Zarządzanie użytkownikami	Sredni	Propozycja	Srednia	Srednie	mania	Vision
FEAT9: Generowanie statystyk	Sredni	Propozycja	Srednia	Srednie	mania	Vision
FEAT10: Określanie zniżek dla klienta	Sredni	Propozycja	Srednia	Srednie	mania	Vision
FEAT11: Wysyłka ofert marketingowych	Sredni	Propozycja	Srednia	Srednie	mania	Vision

Rys. 3. Nadawanie atrybutów w RequsitePro

Ostatnim elementem jest oszacowanie ryzyka. Błędne wykonanie lub niewykonanie danego elementu może pociągać za sobą ogromne konsekwencje finansowe oraz czasowe, dlatego trzeba określić zagrożenia związane z każdym z wymagań. Według RUP kluczem w zarządzaniu ryzykiem jest przewidywanie

[8] Wiegers Karl E. "First Thing First: Prioritizing Requirements" (1999) http://www.processimpact.com/articles/prioritizing.pdf

konsekwencji zanim pojawi się dany problem[9]. Można tu wykorzystać podobną skalę, jak w obu przypadkach powyżej.

Po dokonaniu klasyfikacji najważniejszych wymagań należy przejść do określenia linii bazowej. Pomiędzy charakterystykami opisanymi powyżej istnieje niewielka korelacja[10]. Przykładowo: cecha, która posiada priorytet „krytyczny", wymaga średniego wysiłku oraz poziom ryzyka ma niski, może być kandydatem do linii bazowej. Podobnie będzie w przypadku, gdy wszystkie elementy będą miały znacznik najwyższy. Inna sytuacja będzie dla cechy, która pomimo priorytetu „krytycznego" będzie miała niskie wartości wysiłku i ryzyka. Taki element można odłożyć do realizacji w późniejszych wersjach projektu. Wybranie wymagań, które stworzą linie bazową projektu jest bardzo trudne. Według Leffingwell'a należy uwzględnić wszystkie elementy krytyczne, kilka trudnych, a sprawę pozostałych pozostawia się zespołowi projektowemu. Nie jest to może podejście naukowe, ale jednak skuteczne. Należy także pamiętać, że wymagania rzadko są niezależne, dlatego też przy włączaniu danego wymagania do linii bazowej trzeba uwzględnić także cechy z nim powiązane.

Prace na tym etapie powinny odbywać się przy ścisłej współpracy z klientem. Oczywiście będzie on chciał uwzględnić jak najwięcej cech systemu w pierwszej wersji, ale dzięki różnego rodzaju umiejętnościom negocjacyjnym można osiągnąć kompromis i zawrzeć elementy tylko najbardziej potrzebne, lecz dostarczające określonej funkcjonalności. Oczywiście zdefiniowane tu informacje dotyczące zawartości każdej z wersji mogą w późniejszych etapach ulec zmianie.

5. Uszczegóławianie definicji systemu

Wcześniej zdefiniowane wymagania stanowią pewnego rodzaju zarys, jednak są to w wielu przypadkach bardzo ogólne charakterystyki, które na tym etapie musza być rozbudowane. Proces uszczegóławiania w RUP opiera się na rozbudowywaniu przypadków użycia, które na tym etapie powinny zawierać, krótki opis stanowiący punkt wyjściowy do dalszej pracy. Opisywanie każdego z przypadków odbywa się według poniższego scenariusza:

- Wyznaczenie warunków początkowych inicjujących dany przypadek
- Identyfikacja wszelkich akcji, które mogą przerwać jego działanie
- Uwzględnienie jego interakcji z aktorami i wymiany informacji

Rezultatem powyższych działań jest opis poszczególnych kroków realizowanych przez dany przypadek. Gdy opisy stają się zbyt skomplikowane, należy podzielić je na części. Niestety nie zawsze zachodzący ciąg zdarzeń da się przedstawić za pomocą opisu słownego. Taki opis można zastąpić wtedy inną formą. W literaturze przedmiotu można spotkać się z kilkoma przykładami takich

[9] IBM Corporation "Rational Unified Process" (RUP, 2003)

[10] Leffingwell Dean, Widrig Don „Zarządzanie wymaganiami" (WNT Warszawa, 2003)

opisów[11]. Jednym z nich jest pseudokod, który przedstawia dany ciąg zdarzeń i towarzyszące temu uwarunkowania przy prostych instrukcji programistycznych. Inną metodą są drzewa decyzyjne, które w zależności od sytuacji pokazują odpowiednie czynności począwszy od korzenia aż do liścia. Takiego rodzaju dynamikę odzwierciedlają też diagramy w notacji UML. Można także zastosować diagramy przepływu danych, które podlegają dekompozycji i z ogólnego diagramu przechodzą do bardziej szczegółowych. Ostatnia z przywołanych metod jest wykorzystywana w metodyce Volere wspomagającej zarządzanie wymaganiami na oprogramowanie[12].

Na etapie uszczegóławiania definicji systemu niezbędne jest także zebranie wszystkich informacji o wymaganiach w jednym miejscu. Elementem scalającym wszystkie zebrane informacje dotyczące wymagań jest specyfikacja wymagań na oprogramowanie (*Software Requirements Specification*). Rola takiego dokumentu jest następująca[13]:

1. Pomaga osiągnąć wspólne spojrzenie klienta i twórców systemu na funkcjonalność, jaka musi być spełniona. Zawarte tam informacje pomogą potencjalnemu użytkownikowi stwierdzić, czy tworzony system sprostał oczekiwaniom lub jak go zmodyfikować, aby uzyskać zadawalający efekt.
2. Stanowi podstawę do oszacowania ryzyka i kosztów związanych z projektem
3. Zmniejsza ilość pracy nad projektem, ponieważ dokładne jego przeanalizowanie pozwala wykluczyć zidentyfikowane nieścisłości już w początkowych fazach życia projektu.

W przypadku złożonych projektów takich dokumentów definicyjnych może być wiele. W związku z dużą liczbą narzędzi wspomagającymi proces pracy z wymaganiami informacje o definicji wymagań mogą być rozrzucone pomiędzy liczne artefakty opisujące system. Taki zbiór artefaktów jest w metodyce RUP nazywany pakietem (*Software Requirements Specification Package*), gdzie nie mamy wyłącznie dokumentów, ale także zbiory danych przechowujące wymagania, modele itp. Dokument ten stanowi kolekcję wszystkich wymagań związanych z projektem, który według powszechnie uznawanego standardu powinien być[14]:

1. Poprawny - każde zawarte w nim wymaganie jest rzeczywiście wymaganiem, jakie system powinien spełnić.
2. Jednoznaczny - zrozumiałe dla wszystkich osób związanych w jakiś sposób z projektem. Każde wymaganie powinno mieć tylko jedną

---

[11] Leffingwell Dean, Widrig Don „Zarządzanie wymaganiami" (WNT Warszawa, 2003)

[12] Robertson Suzanne, Robertson James "Mastering the requirements process" (ACM Press, 1999)

[13] Software Engineering Standards Committee of the IEEE Computer Society "IEEE Recommended Practice for Software Requirements Specification" (1998)

[14] Software Engineering Standards Committee of the IEEE Computer Society "IEEE Recommended Practice for Software Requirements Specification" (1998)

interpretację.

3. Kompletny – żadne wymaganie ani niezbędne informacje nie mogą być pominięte
4. Spójny – zgodny z wymaganiami wyższego poziomu oraz z innymi dokumentami.
5. Modyfikowalny – każde wymaganie musi być unikalne, aby łatwo można było śledzić jego historię i wprowadzać ewentualne zmiany. Działanie takie wspiera odpowiednie organizowanie wymagań oraz pilnowanie, aby każde wymaganie występowało w specyfikacji tylko jeden raz.
6. Weryfikowalny – powinien zawierać tylko wymagania weryfikowalne tzn. umożliwiające przeprowadzenie testów lub zastosowanie innych technik weryfikacji.
7. Stopniowalny – powinien zawierać tylko wymagania stopniowalne tzn. takie, które można klasyfikować, grupować w hierarchie na zasadzie relacji nadrzędny – podrzędny
8. Możliwy do śledzenia – umożliwia dostęp do informacji o źródle pochodzenia wymagania i jego powiązania z innymi

6. Zarządzanie zmianą wymagań i weryfikacja zmian

W trakcie pracy nad projektem potrzeba wprowadzenia zmian może pojawić się wielokrotnie. Gdyby było inaczej już dawno powstałyby szablony wymagań, które pasowałyby do każdego projektu i etap zwany zarządzaniem wymaganiami miałby o wiele mniejsze znaczenie w całym cyklu życia systemu. Dlatego też istotną rzeczą jest odpowiednie pogrupowanie wymagań, aby można było sprawnie zarządzać nimi, wprowadzać korekty, zbierać historię zmian, pilnować powiązań pomiędzy poszczególnymi wymaganiami itd. Wiadomo, że nieodpowiednio zarządzana zmiana pociąga za sobą niebezpieczeństwo niepowodzenia projektu, natomiast wszelkiego rodzaju kontrolowane aktualizacje nie tutaj zagrożenia[15]. Powody pojawiających się zmian mogą być różne. Możemy wyróżnić tu cztery podstawowe kategorie tych powodów[16]:

- Środowiskowe – wymagania, które ewoluują z powodu zmian, zachodzących w otoczeniu biznesowym firmy takie jak reorganizacja, zmiana przepisów, regulacji – nie są one związane bezpośrednio z projektem, ale maja na niego bezpośredni wpływ.
- Pojawiające się – wymagania, które stają się zrozumiałe dopiero podczas pracy z systemem, wynikające z braku zrozumienia klienta i analityka
- Wynikowe – wymagania, które pojawiają się dopiero po wdrożeniu gotowego produktu

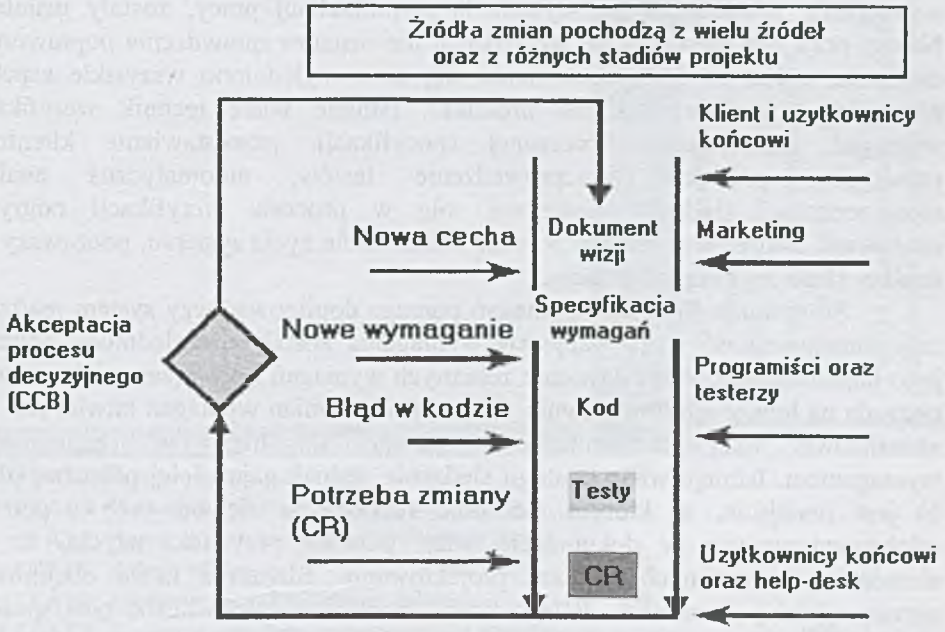
[15] IBM Corporation "Rational RequisitePro - User Guide" (2002) http://www306.ibm.com/software/rational/

[16] Sommerville Ian "Software Engineering 7th edition" (Pearson Education, 2000)

- Wymagania zgodności – wymagania związane z systemami i procesami działającymi wewnątrz firmy klienckiej, które mogą zostać zmienione przez klienta

Potrzeba wprowadzenia zmian może wywodzić się z nowej cechy, którą zdefiniował klient lub też może być wynikiem błędu wykrytego podczas testowania lub kodowania (Rys. 4). Jednak bez względu na miejsce odkrycia zmian należy uwzględnić jej wpływ na wymagania, informacje zawarte w dokumencie wizji, napisany kod, przypadki testowe. Aby nie dopuścić do niezgodności i pracy nad nieaktualnymi wymaganiami tego typu zmiany wprowadza się od „góry do dołu", tj.: najpierw zmienia się informacje zawarte w dokumencie wizji, potem w specyfikacji wymagań, a dopiero potem elementy projektu i przypadki testowe.

W przypadku pojawienia się zmiany nie należy od razu przechodzić do jej wprowadzenia. Najpierw należy dokładnie ją przeanalizować, a dopiero później podejmować jakieś działania. Z reguły powoływany jest specjalny zespół do zarządzania zmianą *(CCB – Change Control Board)*, który zajmuje się weryfikacją takich ewentualności[17]. Dopiero po wydaniu przez nich decyzji związanej z daną korektą, podejmowane są dalsze kroki.



Rys. 4. Źródła zmian w poszczególnych etapach życia systemu[16]

W czasie analizowania takiej zmiany należy rozpoznać konsekwencje takich działań, ich wpływ na koszty, ilość pracy, jaka przepadnie w przypadku

[17] IBM Corporation "Rational RequisitePro - User Guide" (2002) http://www306.ibm.com/software/rational/

akceptacji zmiany, wymagania, które ulegną zmianie, a także wpływ tych działań na linię bazową. W momencie, gdy decyzja o zmianie zostanie podjęta to kolejnym krokiem jest analiza jej wpływu na inne elementy systemu (oprogramowanie, kod, interfejsy, inne systemy, testy) oraz oszacowanie związanych z tymi działaniami kosztów. Końcowym etapem jest implementacja takiej zmiany, czyli uwzględnienie jej w dokumentacji wymagań.

Pojawianie się różnego rodzaju zmian wymaga ciągłego sprawdzania, czy definiowane elementy są zgodne z wcześniejszymi założeniami. Dlatego też weryfikacja wymagań jest bardzo ważnym elementem w cały procesie. Korygowanie wszelkich nieścisłości w późniejszych etapach jest bardzo kosztowne, dlatego też najlepiej wychwycić je na początku. Weryfikacja wymagań definiowana jest następująco:

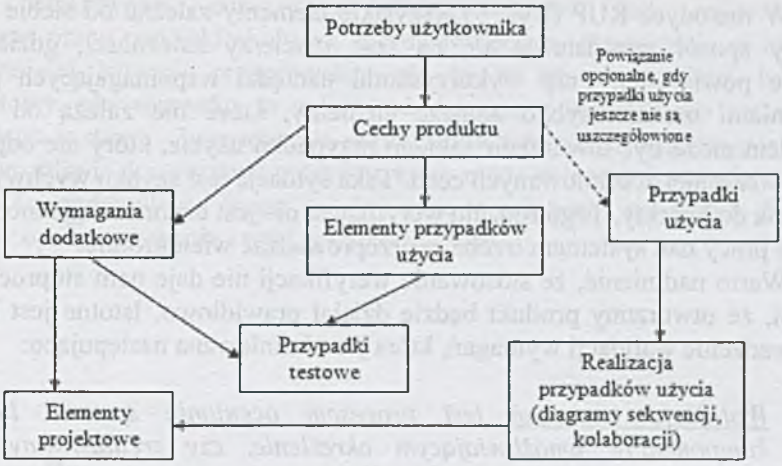
Def.3: *Weryfikacja wymagań jest procesem oceniania systemu lub jego komponentów określający, czy produkt będący na danym etapie spełnia wszystkie oczekiwania zdefiniowane na początku tego etapu"* [18]

Ważne jest dopilnowanie, aby wszelkiego rodzaju niepotrzebne wymagania, prowadzące do wykonania niepotrzebnej pracy, zostały usunięte. Należy przy tym pamiętać, że weryfikacja nie oznacza sprawdzenia poprawności działania, a jest to tylko upewnienie się, że uwzględniono wszystkie aspekty niezbędne z punktu widzenia produktu. Istnieje wiele technik weryfikacji wymagań, np.: analiza tworzonej specyfikacji, przedstawianie klientowi prototypów produktu, przeprowadzenie testów, automatyczna analiza niesprzeczności. Jednak największą rolę w procesie weryfikacji odgrywa możliwość śledzenia wymagań poprzez różne stadia życia systemu, począwszy od analizy aż po jego implementację.

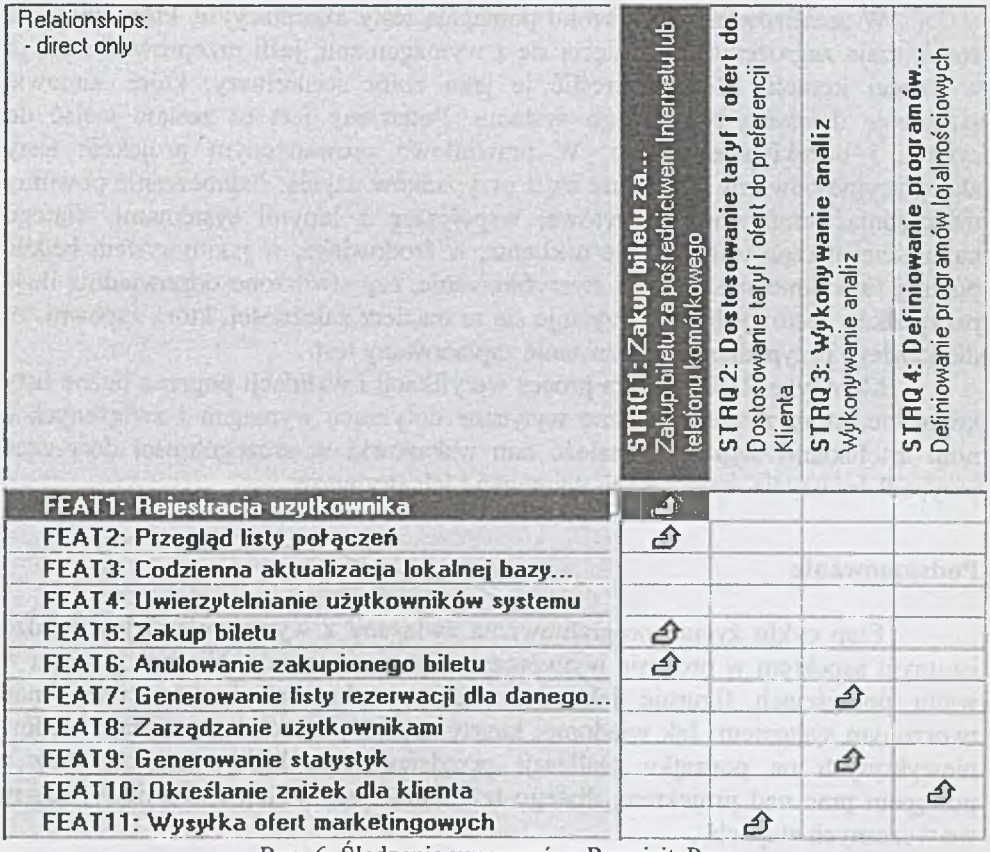
Stosowanie śledzenia wymagań pomaga dopilnować, czy system realizuje całą funkcjonalność i czy wszystkie wymagania zostały uwzględnione podczas jego implementacji. Przedstawienie zebranych wymagań i powiązanie ich ze sobą, pozwala na lepsze ich zrozumienie. W przypadku zmian wymagań łatwiej jest też aktualizować wszystkie elementy mające coś wspólnego ze zmienionym wymaganiem. Istnieje wiele strategii śledzenia, jednak najczęściej polecaną (Rys. 5) jest podejście, w którym śledzenie rozpoczyna się od cech i potrzeb udokumentowanych w dokumencie wizji poprzez przypadki użycia, aż do elementów utworzonych podczas projektowania. Śledzenie może obejmować ogromna liczbę elementów, dlatego trzeba umieć oszacować liczbę tych spośród nich, które zostaną poddane śledzeniu, ponieważ koszty takich zabiegów mogą być istotne.

---

[18] Software Engineering Standards Committee of the IEEE Computer Society "IEEE Standard for Software Verification and Validation" (2004)



Rys. 5. Zależności pomiędzy elementami w RUP[19]



Rys. 6. Śledzenie wymagań w RequisitePro

[19] IBM Corporation "IBM Rational RequisitePro - Evaluation Guide" 2003

W metodyce RUP (Rys. 6) wszystkie elementy zależne od siebie w jakiś określony sposób przedstawia się na tzw. macierzy zależności, gdzie widać wszystkie powiązania. Przy wykorzystaniu narzędzi wspomagających pracę z wymaganiami można szybko znaleźć elementy, które nie zależą od innych. Przykładem może być utworzenie takiego przypadku użycia, który nie odpowiada żadnej z wcześniej zdefiniowanych cech. Taka sytuacja jest szybko wychwytywana i zgłaszana do korekty. Tego rodzaju weryfikacja nie jest czynnością jednorazową i w trakcie pracy nad systemem trzeba ją przeprowadzać wielokrotnie.

Warto nadmienić, że stosowanie weryfikacji nie daje nam stuprocentowej pewności, że utworzony produkt będzie działał prawidłowo. Istotne jest bowiem przeprowadzenie walidacji wymagań, która jest definiowana następująco:

Def.4.: *Walidacja wymagań jest procesem oceniania systemu lub jego komponentów umożliwiającym określenie, czy zrealizowany projekt będzie w pełni zgodny z wyspecyfikowanymi wymaganiami"*[20]

W zrealizowaniu tego kroku pomagają testy akceptacyjne, które znacznie zmniejszają zagrożenie rozminięcia się z wymaganiami, jeśli przeprowadza się je w każdej iteracji. Można określić je jako zbiór scenariuszy, które stanowią symulację działania tworzonego systemu. Podawany jest tu zestaw wejść do sytemu i oczekiwane wyniki. W prawidłowo prowadzonym projekcie testy akceptacyjne powinny wywodzić się z przypadków użycia. Jednocześnie powinny uwzględniać możliwości sprzętowe, współpracę z innymi systemami, dlatego najczęściej przeprowadza się je u klienta, w środowisku, w jakim system będzie później funkcjonować. W celu zweryfikowania, czy stworzono odpowiednią ilość przypadków testowych, wykorzystuje się tu macierz zależności, która zapewni, że dla każdego przypadku użycia zostanie zaplanowany test.

Metodyka RUP wspiera proces weryfikacji i walidacji poprzez liczne listy kontrolne, które zawierają liczne wytyczne dotyczące wymagań i związanych z nimi artefaktami. Możemy znaleźć tam wskazówki w szczególności dotyczące przypadków użycia, specyfikacji wymagań i ich atrybutów.

Podsumowanie

Etap cyklu życia oprogramowania związany z wymaganiami jest bardzo istotnym aspektem w procesie wytwórczym oprogramowania. Nie bez powodu w wielu podejściach figuruje jako jeden z początkowych kroków pracy nad tworzonym systemem. Jak wiadomo, koszty naprawy wszelkiego rodzaju błędów niewykrytych na początku realizacji przedsięwzięcia będą wzrastać wraz z postępem prac nad projektem, dlatego też istotne jest skompletowanie wymagań we wczesnych etapach.

---

[20] Software Engineering Standards Committee of the IEEE Computer Society "IEEE Standard for Software Verification and Validation" (2004)

Zdefiniowanie oczekiwanej funkcjonalności jest najistotniejszym elementem pracy nad jakimkolwiek przedsięwzięciem. Oczywiście w zależności od zaistniałej sytuacji przeprowadza się to w sposób bardziej lub mniej szczegółowy, ale wszystko to zależy od dziedziny późniejszego zastosowania tworzonego systemu. Zastosowanie odpowiednich narzędzi wspomagających ten proces umożliwia dostarczenie produktu, który może w lepszym stopniu zadowolić klienta. To zadowolenie może mieć istotne przełożenie na wynik finansowy dostawcy oprogramowania i tutaj koło się zamyka ...

ROZDZIAŁ III

ZARZĄDZANIE PROJEKTAMI IT – MODEL MAPOWANIA KOMPETENCJI I AKTYWNOŚCI

Kazimierz FRĄCZKOWSKI, Aleksandra SIDORCZYK

Wprowadzenie

Według raportu „CHAOS" opracowanego przez Standish Group, w 1994 roku jedynie 16% projektów informatycznych zakończyło się sukcesem [10]. W roku 2000 liczba udanych projektów wzrosła do 28% [9], a w roku 2004 wyniosła jedynie 29% [8]. Liczba projektów kończących się porażką jest niepokojąca, zwłaszcza, że na przestrzeni kilku ostatnich lat obserwuje się lawinowy wzrost zapotrzebowania na oprogramowanie, którego rośnie złożoność oraz obszary zastosowań. Dodatkowo należy pamiętać, że zdecydowana większość przedsięwzięć z innych dziedzin kończy się sukcesem.

Jakie są zatem przyczyny niepowodzeń projektów informatycznych? Jakie są zalecenia, aby zmniejszyć wskaźnik projektów które kończą się klęska? Przede wszystkim informatyka jest stosunkowo młodą dziedziną, w której brakuje wypracowanych standardów, norm postępowania. „Wszystkie kłopoty wynikają ze stosowania niewłaściwych metod (albo z pracowania w ogóle bez metod) ... Innymi słowy, przyczyną marszu ku klęsce jest nasza głupota albo nieudolność" [1]. Doświadczenie jest tym czynnikiem, który w znacznym stopniu determinuje sukces podejmowanych działań. Należy zatem pracować nad standaryzacją prowadzenia projektów IT, nad rozwijaniem i porządkowaniem procesu zarządzania przedsięwzięciami informatycznymi. Zarządzanie projektem informatycznym obejmuje całą serię działań niezbędnych do jego satysfakcjonującej realizacji, takich jak: zarządzanie czasem, zakresem, kosztami, a także zasobami ludzkimi. Koncentracja na wszystkich wyżej wymienionych elementach jest warunkiem koniecznym sukcesu przedsięwzięcia. Nie można jednak zapominać, że projekty informatyczne są realizowane przez ludzi i to właśnie zespół projektowy w dużej mierze decyduje o ich powodzeniu. Jak powiedział Bjarne Stroupstrup „Projektowanie i programowanie to zajęcia ludzkie; zapomnij o tym i wszystko stracone". Ludzie stanowią kluczowy czynnik sukcesu projektu. Od ich kwalifikacji i umiejętności, zachowań i nastawienia zależy czy przedsięwzięcie się powiedzie. „Wybór wykwalifikowanych pracowników jest jak ulokowanie pieniędzy w banku" jak stwierdził John Boudreau. Jednak temat zarządzania kapitałem ludzkim, zarządzania zasobami jest w dalszym ciągu poruszany jedynie w tle innych aspektów realizacji przedsięwzięć informatycznych. Problem polega na tym, że cały proces alokacji zasobów w projektach bazuje przede wszystkim na kryteriach czasowych. „Zarządzanie zasobami to proces kontrolowania ich dostępności, dyspozycyjności"[6]. Jednak zarządzanie zasobami to nie tylko panowanie nad czasem, to również właściwy

dobór ludzi do zadań projektowych na podstawie ich kompetencji czyli wiedzy, umiejętności, doświadczenia. „Wiedza to pieniądz." – jak powiedział Tom Pennington. Jednak, żeby wiedza zaczęła przynosić zyski trzeba umieć nią efektywnie zarządzać, optymalnie ją wykorzystywać.

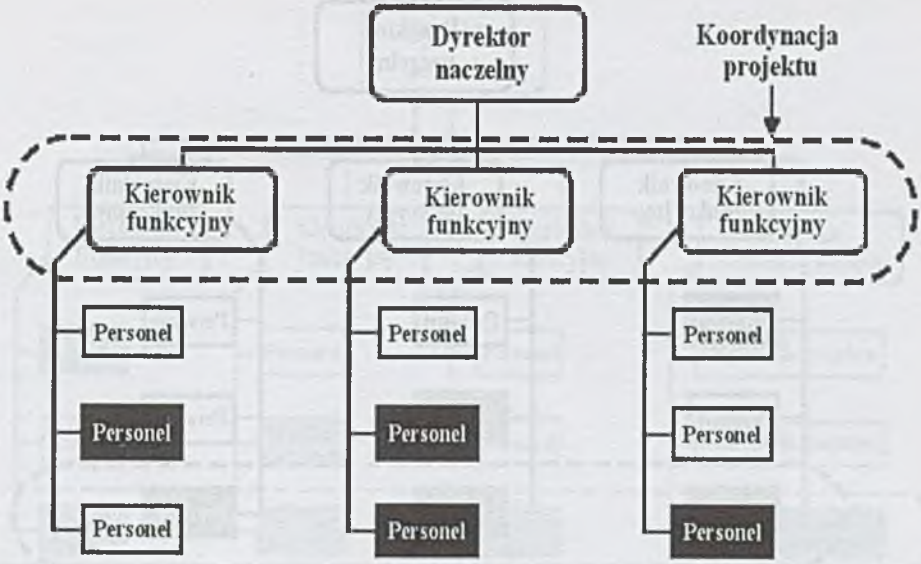
Dekompozycja projektu na mniejsze podprojekty, fazy oraz aktywności to zalecenia wynikające z badań Standish Group, które mają przyczynić się do zwiększenia szans na zakończenie projektu sukcesem. Dopiero aktywności, wyodrębnione w związku z koniecznością wykonania określonej pracy, niezbędnej do dostarczenia zaplanowanego produktu czy usługi, wskazują pożądane kompetencje konieczne do satysfakcjonującego wykonania zlecenia.

1. Środowisko realizacji projektów IT

Na przestrzeni ostatnich kilku lat popyt na usługi firm informatycznych znacznie wzrósł, niosąc za sobą potrzebę nie tylko powstawania nowych przedsiębiorstw, ale przed wszystkim rozwój już istniejących. Małe organizacje przekształcają się w średnie, co pociąga za sobą konieczność rozbudowy struktury organizacyjnej firmy. Masowe zatrudnianie nowych pracowników wymaga posiadania niezależnego działu kadr, finansów, itd., podczas gdy wcześniej odpowiedzialność za zadania wykonywane w obrębie tych działów ponosiła nierzadko jedna osoba.

W czasach gdy zapotrzebowanie na różnego rodzaju oprogramowanie nieustannie wzrasta istnieje już znikoma liczba firm informatycznych realizujących projekty sekwencyjnie. Większość przedsiębiorstw IT doszło do wniosku, że realizacja kilku równoległych projektów nie tylko przynosi większe dochody, ale również pozwala zaoszczędzić część ponoszonych kosztów (kosztów stałych) poprzez dzielenie je pomiędzy kilka wykonywanych równolegle przedsięwzięć. Firmy w ten sposób współdzielą również zasoby ludzkie, mogą zatrudnić mniej pracowników, ograniczyć ich „czas oczekiwania na wejście w kolejny projekt", ponieważ projekty przebiegają współbieżnie. Tak więc struktura organizacyjna większości firm IT jest najczęściej struktura macierzową. Łączy ona w sobie dwa podejścia do organizacji firmy: jedno zorientowane na funkcje przedsiębiorstwa, drugie zorientowane na projekty. W ogólności wyróżnia się trzy podstawowe rodzaje struktur organizacyjnych firmy omówionych szerzej w pracach [2,3,5]:

- funkcyjną
- projektową
- macierzową

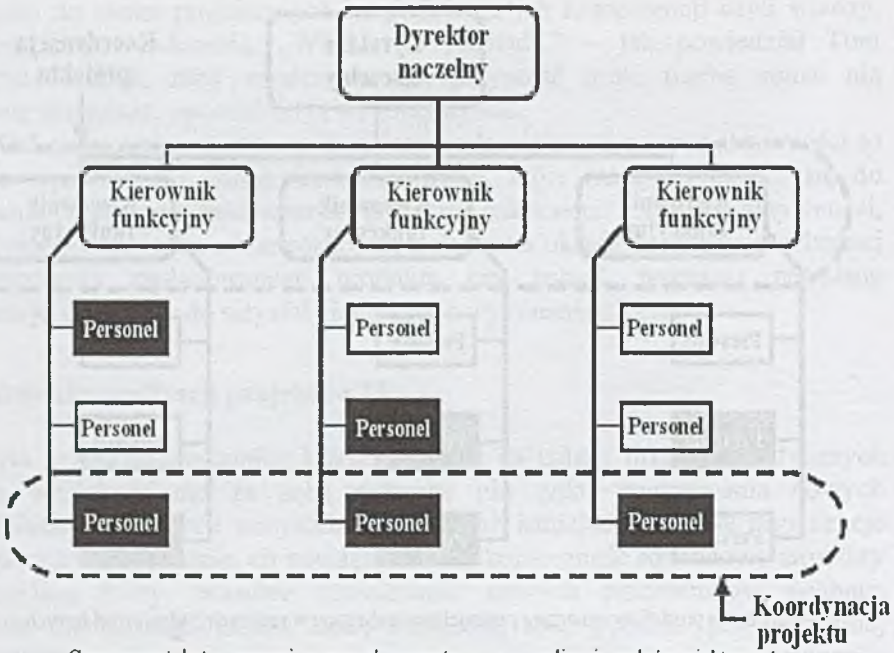


Czarne prostokąty oznaczają personel zaangażowany w realizację zadań projektowych

Rys. 1. Struktura organizacyjna firmy – funkcyjna (na podstawie [7])

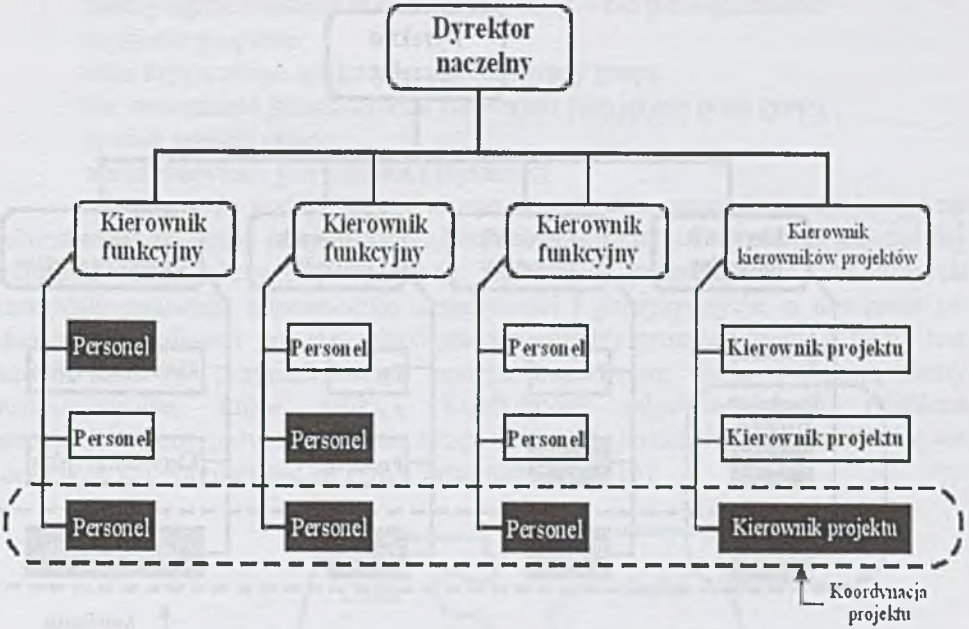
Firmy o *strukturze funkcyjnej* posiadają jasną budowę organizacyjną skupioną w obrębie działów, w której pracownik ma jednego ściśle określonego przełożonego. Projekty w takim przedsiębiorstwie nie są postrzegane całościowo, a percepcja zakresu podejmowanych przedsięwzięć jest dyktowana granicami pomiędzy działami firmy.

Struktura projektowa charakteryzuje organizacje, w których większość zasobów bierze udział w podejmowanych przedsięwzięciach, a kierownicy projektów są jednostkami niezależnymi o dużym wpływie i autorytecie. W takim przedsiębiorstwie działy albo raportują bezpośrednio do kierownika projektu, albo wspierają swą działalnością różne przedsięwzięcia.



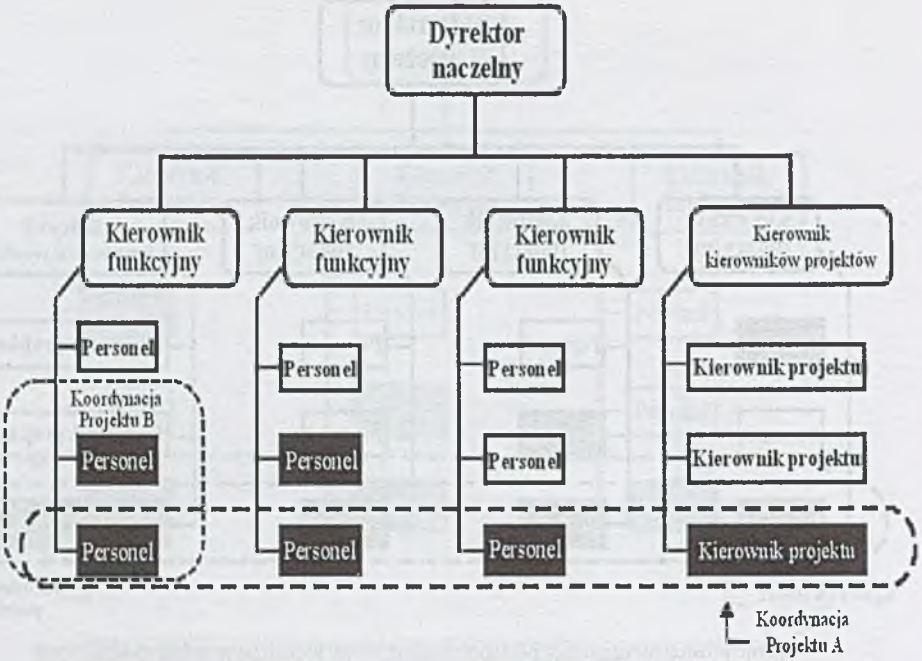
Rys. 2. Struktura organizacyjna firmy – macierzowa słaba (na podstawie [7])

Przedsiębiorstwa o *strukturze macierzowej* łączą w sobie oba wyżej wymienione podejścia. Wśród powstałych struktur macierzowych istnieją zarówno struktury bliższe podejściu funkcjonalnemu jak i takie, których charakterystyka jest bardziej zbliżona do podejścia projektowego. Tak na przykład struktura macierzowa słaba posiada wiele cech struktury funkcyjnej, a kierownik projektu pełni jedynie rolę koordynatora. Przeciwnie w silnych strukturach macierzowych kierownik projektu posiada duży autorytet i faktycznie kieruje realizacją przedsięwzięcia.



Rys. 3. Struktura organizacyjna firmy – macierzowa silna (na podstawie [7])

Firmy starają się modyfikować przestawione struktury dopasowując je do indywidualnych potrzeb. Struktura firm IT to nie jedyny składnik środowiska realizacji projektów informatycznych. Innym niezwykle istotnym czynnikiem jest sposób budowy zespołów projektowych realizujących przedsięwzięcia.



Rys. 4. Struktura organizacyjna firmy – macierzowa złożona (na podstawie[7])

Wyróżnia się dwa podstawowe podejścia w procesie tworzenia zespołów [3]. Pierwsze to *model dojrzewania*, polegający na tym, że firma posiada względnie stały zbiór pracowników, którzy realizują zlecane jej projekty. Pracownicy w razie braku kwalifikacji czy umiejętności wymaganych do realizacji wskazanego przedsięwzięcia są wysyłani na szkolenia, bądź rozwijają się i nabywają nową wiedzę bezpośrednio w trakcie realizacji podejmowanych projektów. Przedsiębiorstwo, w którym wykorzystywany jest model dojrzewania jest kojarzone jako firma ucząca się.

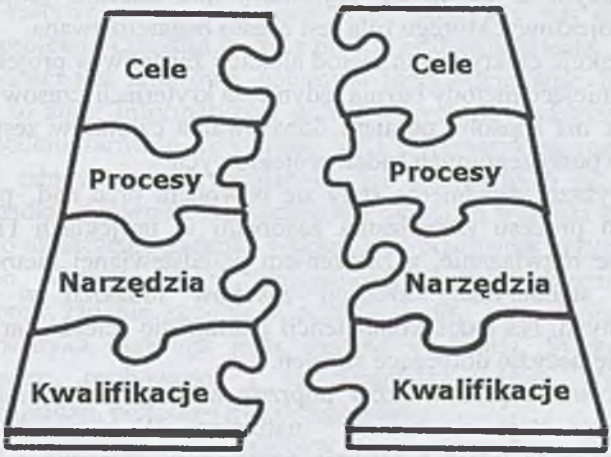
Model dojrzewania może opierać się o cztery poziomy struktur zespołu wirtualnego. Na każdym z tych poziomów pojawiają się kluczowe problemy, charakterystyczne dla każdego z nich. *Model dojrzewania* jest w stanie określić czas, jaki zajmie organizacji przejście z aktualnego poziomu do kolejnego. Miarą wydajności zespołu jest jego zdolność do realizacji projektów w narzuconym czasie i budżecie. Przejście do kolejnych poziomów zwiększa wydajność, poprzez optymalizację poszczególnych aspektów działania zespołu.

Zespół niezależnie od fizycznego usytuowania, sposobu komunikacji oraz realizowanego zadania na pewnym etapie działania nabywa cech pozytywnych oraz negatywnych do których należą między innymi [2]:

- swoista ideologia

- normy egzekwowania w sposób naturalny – nie ma regulaminu
- myślenie grupowe
- brak krytycznego spojrzenia na efekt pracy grupy
- nie rozważanie jakichkolwiek rozwiązań przyjętych poza grupą
- spadek jakości pracy
- manifestowanie jednolitości i lojalności

Alternatywne podejście nosi nazwę *modelu dopasowania*, polegającego na dobieraniu ludzi ściśle do projektów. Firma wykorzystująca ten model najczęściej zatrudnia pracowników do realizacji wybranych przedsięwzięć, do których kandydaci posiadają odpowiednie umiejętności i predyspozycje, a następnie po ukończeniu realizacji projektu współpraca pomiędzy pracownikiem, a firmą jest finalizowana. W przypadku tego podejścia pomocną dłoń wyciągają firmy outsourcingowe, które szukają kandydatów odpowiadających profilom pracowników poszukiwanych przez firmy. Aktualnie model dopasowania zdobywa coraz większą popularność na rynku informatycznym [5].



Rys. 5. Model dopasowania zespołu projektowego [2]

Powodem tego jest fakt, iż klienci oczekują otrzymania zamówionego produktu w jak najkrótszym czasie, co pociąga za sobą konieczność maksymalizacji efektywności i wydajności pracy zespołów projektowych. Brakuje czasu na dokształcanie się pracownika podczas realizacji projektu. Od członków zespołów projektowych wymaga się, żeby posiadali oni kwalifikacje i umiejętności wymagane do realizacji przedsięwzięcia od momentu jego rozpoczęcia. Również z tego powodu tak wyraźnie wzrasta rola outsourcingu.

„Zgodnie z przewidywaniami IDC, outsourcing usług IT będzie jednym z najbardziej dynamicznie rozwijających się obszarów IT w ciągu najbliższych pięciu lat. Jego wzrost jest szacowany na poziomie 16% rocznie. Zdaniem analityków, spowolnienie gospodarcze oraz ograniczanie wydatków na IT są istotnym stymulatorem do zawierania kontraktów outsourcingowych, które

zapewniają oszczędności praktycznie od momentu uruchomienia usługi." Andrzej Więckiewicz z HP

2. Kompetencyjny model alokacji zasobów w projektach IT

Przy obecnym tempie rozwoju rynku, gospodarki i technologii temat zarządzania wiedzą, kapitałem ludzkim nabiera coraz większego znaczenia. Coraz częściej słyszy się o pojęciu kompetencji. Powstają słowniki kompetencji, kompetencyjne opisy stanowisk. Przyjmując pracowników wymaga się od nich określenia zbioru posiadanych przez nich kwalifikacji.

Problemy związane z realizacją przedsięwzięć informatycznych to miedzy innymi:

- niewielka część realizowanych projektów kończy się 100% sukcesem
- wzrasta popyt na oprogramowanie, wzrastają też oczekiwania klientów
- konieczna jest optymalizacja czasu przeznaczonego na realizację projektów
- jednym z głównych czynników powodzenia projektów jest zespół projektowy, którego rola jest często bagatelizowana
- brakuje efektywnych metod alokacji zasobów w projektach
- istniejące metody bazują jedynie na kryteriach czasowych
- nie ma sposobu pomiaru dopasowania członków zespołu projektowego do poszczególnych zadań projektowych

Powyższe zagadnienia stały się powodem prac nad propozycją nowego podejścia do procesu zarządzania zasobami w projektach IT [4]. Nie jest to kompleksowe rozwiązanie, a założeniem przedstawianej metody jest stworzenie fundamentu standaryzacji alokacji zasobów ludzkich w przedsięwzięciach informatycznych. Na bazie kompetencji planuje się ścieżki kariery pracowników, podejmuje się decyzje dotyczące szkoleń.

Opisywanie pracowników poprzez wiedzę, umiejętności, doświadczenia oraz zdolności staje się rzeczą naturalną. W przypadku zwłaszcza firm informatycznych to właśnie głównie pracownicy stanowią o wartości rynkowej firmy. Zespoły projektowe zbudowane z doskonałych specjalistów są niewyobrażalnie więcej warte niż te, skupiające ludzi o mniejszych wymaganiach finansowych, którzy jednak potrzebują tygodni lub miesięcy, by przedrzeć się przez nowe procesy lub technologie zanim staną się produktywni, a ich praca wydajna i efektywna. Jedna z firm informatycznych zatrudniająca ponad 1000 pracowników umieściła na swojej stronie internetowej hasło: „To ludzie tworzą projekty." Dalej możemy przeczytać, że dobrzy pracownicy potrafią osiągać dobre wyniki nawet w niesprzyjających okolicznościach. Kwalifikacje i wielkość zespołu determinują sukces firmy i tworzonego oprogramowania. Jednak posiadanie dobrego zespołu to tylko połowa sukcesu. Trzeba wiedzieć jak efektywnie zarządzać taką grupą ludzi, żeby optymalnie eksploatować ich twórczy potencjał. Niewątpliwie każdy menadżer projektu wie doskonale, że elementem koniecznym w procesie prowadzenia przedsięwzięcia jest jego dekompozycja na elementy składowe. Już Henry Ford zauważył, że „Nic nie jest wyjątkowo trudne, jeśli

zostanie podzielone na mniejsze zadania". Zakres, wielkość, złożoność każdego problemu najłatwiej jest zdefiniować dzieląc go na czynniki pierwsze, na proste, *pojedyncze działania*. Struktura podziału pracy WBS jest elementem występującym praktycznie w każdym projekcie. Ale co następuje dalej? Wyodrębnione aktywności projektu nie są opisywane w sposób wystarczający do ich efektywnego zarządzania. Brakuje informacji, co jest warunkiem koniecznym ukończenia zadania projektowego, czyli co osoba przydzielona do realizacji aktywności powinna wiedzieć, jakie kwalifikacje powinna posiadać, aby bez większych przeszkód była w stanie ukończyć zadanie w oczekiwanym czasie oraz o satysfakcjonującej jakości. W tym momencie pojawia się wolna przestrzeń, w której widać rolę kompetencji, jako warunku koniecznego realizacji aktywności, wymagań, które muszą być spełnione, by zadanie projektowe zostało ukończone zgodnie z założeniami. Należy wykorzystać kompetencje do opisu projektu.

Rozważmy zatem następującą rzeczywistość projektową, której wejście stanowi projekt informatyczny oraz zespół projektowy przydzielony do jego realizacji. Projekt składa się ze zbioru aktywności wykonywanych sekwencyjnie bądź równolegle. Aktywności są określone przez priorytety ich wykonania w kontekście całego projektu. Oznacza to, że z punktu widzenia klienta, dostarczenie pewnej funkcjonalności jest kluczowe, natomiast innej opcjonalne. Kontynuując, zespół projektowy to zbiór informatyków delegowanych do wykonania projektu. W rozważanym modelu zarówno aktywność jak i informatyk są rozumiani jako zbiór kompetencji odpowiednio wymaganych lub posiadanych na określonym poziomie. W przypadku aktywności, kompetencje określają warunki konieczne do satysfakcjonującego ukończenia zadania, wskazują co pracownik przypisany do ich realizacji powinien wiedzieć, jakie umiejętności i doświadczenie powinien posiadać, by wykonać zadanie zgodnie z oczekiwaniami i założeniami. Z kolei kompetencje informatyka określają jego wartość w firmie. Stanowią o jego zdolnościach, wiedzy, osobowości i są pewnego rodzaju wytyczną, do wykonywania jakich zadań, powinien być delegowany.

Propozycja metody alokacji zasobów do zadań polega na szacowaniu dopasowania informatyka do aktywności na podstawie kompetencji. Porównując kompetencje i poziomy ich posiadania przez pracownika oraz kompetencje wymagane do realizacji aktywności możliwe jest dokonanie oceny wskazującej w jakim stopniu wybrany pracownik odpowiada wymaganiom realizacji zadania, aby wykonane zostało na odpowiednim poziomie jakości, w zakładanym czasie.

2.1 Definicje pojęć

Przy budowie modelu wprowadzono następujące pojęcia oraz oznaczenia

P – projekt
A – aktywność
Z – zespół projektowy
I – informatyk
k – kompetencja

w – waga aktywności
pz – poziom kompetencji

Projekt

- zbiór aktywności, o różnym poziomie istotności z punktu widzenia pomyślnego ukończenia całego projektu
- zbiór par (aktywność, waga)
  - aktywność – zadanie realizowane w ramach projektu
  - waga – priorytet aktywności w projekcie

$$P = \left\{(A,w):\ w>0 \ \wedge\ \forall_{(A_1,w_1)\in P}\forall_{(A_2,w_2)\in P}\ (A_1 = A_2) \Rightarrow (w_1 = w_2)\right\}$$

Aktywność

- czynność, bądź grupa tematycznie pogrupowanych czynności będąca częścią przedsięwzięcia; jest określona poprzez kompetencje wskazujące warunki konieczne do satysfakcjonującego jej ukończenia
- zbiór par (kompetencja, poziom)
  - kompetencja – warunek konieczny realizacji zadania
  - poziom – wymagany poziom posiadania kompetencji

$$A = \left\{(k,pz):\ pz>0 \ \wedge\ \forall_{(k_1,pz_1)\in A}\forall_{(k_2,pz_2)\in A}\ (k_1 = k_2) \Rightarrow (pz_1 = pz_2)\right\}$$

Zespół projektowy

- zbiór pracowników firmy, którzy zostali delegowani do realizacji zadań w ramach projektu
- zbiór informatyków

Informatyk

- to członek zespołu projektowego, którego zadaniem jest realizacja aktywności związanych z projektem; jest określony poprzez kompetencje stanowiące o jego zdolnościach, wiedzy, osobowości
- zbiór par (kompetencja, poziom)
  - kompetencja – właściwość informatyka
  - poziom – poziom posiadania kompetencji

$$I = \left\{(k,pz):\ pz>0 \ \wedge\ \forall_{(k_1,pz_1)\in I}\forall_{(k_2,pz_2)\in I}\ (k_1 = k_2) \Rightarrow (pz_1 = pz_2)\right\}$$

Kompetencja

- wiedza, umiejętność, zdolność, którą zasób ludzki może posiadać i która jest konieczna lub istotna z punktu widzenia realizacji konkretnej aktywności, którą zasób może wykonać samodzielnie
- warunek niezbędny do samodzielnej realizacji aktywności lub cecha posiadana przez członka zespołu projektowego

Poziom kompetencji

- określa stopień wskazanych umiejętności, wiedzy czy zdolności posiadanych przed członka zespołu projektowego lub wymaganych do wykonania konkretnego zadania w obrębie projektu
- liczba całkowita z przedziału <1;5> (patrz punkt 2.3.2)

Waga aktywności

- wskazuje priorytet realizacji zadania w ramach projektu, określa jak ważne z punktu widzenia pomyślnego ukończenia całego projektu jest wykonanie danej aktywności
- liczba całkowita z przedziału <1;3> (patrz punkt 2.3.3)

2.2. Szacowanie

Ponieważ zarówno ilościowy opis-specyfikacja aktywności, jak również kompetencje określają warunki konieczne do satysfakcjonującego ukończenia zadania. Zakład się, że pracownik przypisany-przewidziany do ich realizacji aktywności powinien wiedzieć jakie umiejętności i doświadczenie powinien posiadać, by wykonać zadanie zgodnie z oczekiwaniami i założeniami. Z kolei kompetencje informatyka z założenia szacować mają jego wartość w firmie tj. jaką wartość dodaną wnosi przy danym projekcie. Szacowane przed projektem jego predyspozycje stanowią o jego zdolnościach, wiedzy, osobowości i są pewnego rodzaju wytyczną, do wykonywania określonych zadań, przesłanka, że powinien być delegowany do „pasujących do niego" aktywności.

2.2.1 Pokrycie kompetencji

Zgodność pokrycia kompetencji :

- wskazuje w jakim stopniu pracownik spełnia wymagania aktywności związane z określoną kompetencją
- stosunek poziomu kompetencji pracownika, do wymaganego poziomu kompetencji aktywności

$$PK(k) = \min\left(\frac{pz_I}{pz_A}, 1\right) * 100\%, \quad (pz_I, k) \in I \ \wedge \ (pz_A, k) \in A$$

Zakłada się, że posiadanie kompetencji na poziomie wyższym niż wymagany do realizacji aktywności, będzie traktowane jako jej posiadanie na wymaganym poziomie, czyli pokrycie kompetencji będzie stanowiło maksymalnie 100%.

2.2.2 Pokrycie aktywności

- wskazuje w jakim stopniu pracownik spełnia wymagania aktywności określone poprzez kompetencje wymagane do jej realizacji
- średnia pokryć wszystkich kompetencji wymaganych (na poziomie co najmniej 1) do wykonania wskazanej aktywności

$$PK(A) = \frac{1}{n}\sum^{n} PK(k), \quad (k, pz) \in A \wedge n = card(A)$$

2.2.3 Pokrycie projektu

Pokrycie projektu:

- wskazuje w jakim stopniu zespół projektowy delegowany do realizacji przedsięwzięcia, spełnia wymagania projektu, określone poprzez kompetencje niezbędne do wykonania wszystkich jego aktywności
- średnia ważona pokryć wszystkich aktywności w obrębie projektu, gdzie wagę stanowi poziom istotności realizacji aktywności, w kontekście ukończenia całego projektu

$$PK(P) = \frac{\sum^{n} PK(A) * \frac{1}{w}}{\sum^{n} \frac{1}{w}}, \quad (A, w) \in P \wedge n = card(P)$$

Pokrycie mierzymy w %.

2.3. Uzupełnienie modelu

2.3.1. Kompetencja wektorowa

W powyższym podejściu kompetencja została przedstawiona jako wartość jednostkowa. W rzeczywistości jest ona jednak pojęciem bardziej złożonym. W wielu pracach rozważana jest jako wartość wektorowa, na którą składają się nie tylko wiedza, ale także praktyka, czyli umiejętność wykorzystania teorii w warunkach rzeczywistych, sprawność jej wykorzystania. Tak, więc kompetencję można przedstawić jako wektor kilku składowych – wiedzy teoretycznej, doświadczenia (praktyki), biegłości w ponownym wykorzystaniu posiadanych wiadomości teoretycznych. Takie rozdzielenie kompetencji jest istotne z punktu widzenia kompetencji posiadanych przez pracownika, który może się rozwijać, nabywać nową wiedzę, doświadczenie. W przypadku kompetencji wymaganych do wykonania aktywności rozbicie kompetencji nie jest zabiegiem koniecznym, a nawet nie do końca wskazanym ponieważ utrudniałoby opisywanie aktywności.

Powyższe uściślenie pojęcia kompetencji jest zauważalną zmianą biorąc pod uwagę poziom kompetencji zdefiniowany w modelu, który teraz dla

kompetencji posiadanych (tych związanych z pracownikiem), będzie wektorem o trzech składowych.

Poziom kompetencji:

- określa stan posiadanej wiedzy, zdobytego doświadczenie oraz biegłości w ponownym wykorzystaniu informacji teoretycznych, pracownika
- wektor o trzech składowych (wiedza, doświadczenie, biegłość)

$$pz = (w, d, b)$$

Zastosowanie poziomu wektorowego dla kompetencji posiadanych przez pracownika i poziomu skalarnego dla kompetencji wymaganych w ramach aktywności, wprowadza konieczność redukcji poziomu wektorowego do skalarnego. Jest to działanie potrzebne w celu dokonania mapowania kompetencji pracownika i zadania w procesie szacowania pokrycia kompetencji, aktywności czy całego projektu.

Zredukowany poziom kompetencji:

- określa stopień wskazanych umiejętności, wiedzy członka zespołu projektowego
- średnia poziomów: wiedzy, doświadczenia i biegłości

$$pz = round\left(\frac{w + d + b}{3}\right)$$

$$round(x) = \begin{cases} [x], & dla\ [x] - \{x\} < 0{,}5 \\ [x] + 1, & dla\ [x] - \{x\} >= 0{,}5 \end{cases}$$

2.3.2. Poziomy kompetencji – przykładowa skala pomiaru

Tablica 1. Skala pomiaru składowej *Wiedza* wektorowego poziomu kompetencji pracownika oraz jednocześnie skala pomiaru poziomu kompetencji aktywności

Poziom	Wiedza
Ekspert (5)	Wszechstronna znajomość tematu
Biegły (4)	Biegła znajomość tematu
Kompetentny (3)	Podstawowa znajomość tematu
Podstawowy (2)	Podstawowe rozeznanie
Wstępny (1)	–

Tablica 2. Skala pomiaru składowej *Doświadczenie* wektorowego poziomu kompetencji pracownika

Poziom	Doświadczenie
Ekspert (5)	powyżej 2 lat
Biegły (4)	od 6 mies. do 2 lat
Kompetentny (3)	od 2 do 6 mies.
Podstawowy (2)	od 2 tyg. do 2 mies.
Wstępny (1)	poniżej 2 tyg.

Tablica 3. Skala pomiaru składowej *Biegłość* wektorowego poziomu kompetencji pracownika

Poziom	Biegłość
Ekspert (5)	poniżej 2 tyg.
Biegły (4)	od 2 tyg. do 2 mies.
Kompetentny (3)	od 2 do 6 mies.
Podstawowy (2)	od 6 mies. do 2 lat
Wstępny (1)	powyżej 2 lat

W tabeli 3. *Biegłość* oznacza kiedy ostatnio pracownik wykorzystywał rozważaną kompetencję w praktyce. Jeśli jest to okres poniżej *2 tygodni*, poziom składowej *Biegłość* wynosi *Ekspert (5)*.

2.3.3. Wagi aktywności – przykładowa skala pomiaru

Tablica 4. Priorytety aktywności w projekcie.

Priorytet	Opis
Wysoki (1)	Wykonanie zadania jest kluczowe z punktu widzenia realizacji całego projektu
Średni (2)	Wykonanie zadania jest ważne z punktu widzenia realizacji całego projektu
Niski (3)	Wykonania zadania jest elementem dodatkowym z punktu widzenia realizacji całego projektu

3. Przykład

Rozważmy projekt informatyczny składający się kilkuset aktywności oraz zespołu projektowego złożonego z kilkudziesięciu informatyków. W celu zademonstrowania działania modelu w szacowaniach zostanie uwzględniony jeden uproszczony etap realizacji projektu, obejmujący zaprojektowanie bazy danych oraz część zespołu projektowego odpowiedzialna za jego wykonanie.

W ramach projektowania bazy danych wyodrębniono następujące aktywności:

- Identyfikacja aktorów, diagram kontekstowy
- Identyfikacja encji i klas

- Identyfikacja reguł biznesowych oraz ograniczeń dziedzinowych
- Transformacja do modelu logicznego (relacyjnego)
- Transformacja do modelu fizycznego

Część zespołu projektowego, przydzielona do zadań związanych z wykonaniem projektu bazy danych składa się z trzech osób.
Warunkiem wykorzystania modelu jest opisanie aktywności projektowych zbiorem kompetencji wymaganych do ich realizacji oraz określenie ich priorytetów w ramach przedsięwzięcia. Konieczne jest również opisanie zespołu projektowego zbiorem kompetencji charakteryzujących jego wiedzę, umiejętności i doświadczenie.

Tablica 5. Kompetencyjny opis aktywności projektu

Aktywność	Waga	Kompetencja	Poziom
Identyfikacja aktorów, diagram kontekstowy	2	UML	3
		Identyfikacja wymagań	2
Identyfikacja encji i klas	1	UML	4
		Analityczne myślenie	3
Identyfikacja reguł biznesowych oraz ograniczeń dziedzinowych	1	SQL	3
		Metody projektowania baz danych	3
		Analityczne myślenie	4
Transformacja do modelu logicznego (relacyjnego)	1	Metody projektowania baz danych	4
		Metody normalizacji baz danych	3
		SQL	4
Transformacja do modelu fizycznego	1	SQL	3
		MySQL	3

Tabela 6. Kompetencyjny opis zespołu projektowego

Kompetencja	Wiedza	Doświadczenie	Biegłość						
	P1	P2	P3	P1	P2	P3	P1	P2	P3
Metody projektowania baz danych	3	2	1	3	3	1	2	2	1
Metody	4	3	2	3	2	2	2	2	1

normalizacji baz danych									
MySQL	4	2	1	4	2	1	3	2	1
SQL	4	3	2	4	3	1	4	4	1
Budowa WBS	2	1	1	1	1	1	1	1	1
Analityczne myślenie	3	2	2	3	2	2	3	2	2
Angielski	4	2	2	5	3	2	5	2	2
UML	3	2	1	2	2	1	3	2	1

W tabeli 6. P1, P2, P3 oznaczają trzech pracowników zespołu projektowego.

Rozpatrzmy aktywność *Transformacja do modelu fizycznego*, do której realizacji wymagane jest posiadanie następujących kompetencji na wskazanym poziomie:

- SQL : 3
- MySQL: 3

oraz pracownika P2 o kompetencjach odpowiednio:

- SQL : (3,3,4)
- MySQL: (2,2,2)

gdzie wektor oznacza poziom wiedzy, doświadczenia i biegłości pracownika związany ze wskazaną kompetencją

Aby oszacować pokrycie kolejnych kompetencji, czy dalej pokrycie całej aktywności należy obliczyć zredukowany poziom kompetencji posiadanej przez zasób delegowany do wykonania zadania.

Dla kompetencji *SQL* i pracownika *P2* poziom zredukowany zgodnie ze wzorem wynosi:

$$pz = round\left(\frac{w+d+b}{3}\right) = round\left(\frac{3+3+4}{3}\right) = 3$$

Kolejnym krokiem zbliżającym do obliczenia pokrycia projektu, jest obliczenie pokrycia kompetencji wymaganych w ramach realizacji danego zadania projektowego.

Pokrycie kompetencji *SQL*, w ramach aktywności *Transformacja do modelu fizycznego*, kompetencją zasobu P2 wynosi:

$$PK(k) = \min\left(\frac{pz_I(k)}{pz_A(k)}, 1\right) * 100\% = \min\left(\frac{3}{3}, 1\right) * 100\% = 100\%$$

Tablica 7. Poziomy kompetencji wymaganych i posiadanych dla aktywności *SQL*

Kompetencja	Poziom wymagany	Poziom (P1)	Poziom (P2)	Poziom (P3)	Pokrycie dla (P1)	Pokrycie dla (P2)	Pokrycie dla (P3)
SQL	3	4	3	1	100%	100%	33,33%
MySQL	3	4	2	1	100%	66,67%%	33,33%
Pokrycie aktywności					100%	83,33%	33,33%

W celu obliczenie pokrycia całej aktywności, należy policzyć średnią pokryć wszystkich kompetencji wymaganych do realizacji wskazanego zadania.

Obliczając pokrycie aktywności *Transformacja do modelu fizycznego* kompetencjami pracownika P2, należy uwzględnić pokrycie dwóch kompetencji: *SQL i MySQL*

$$PK(A) = \frac{1}{n}\sum^{n} PK(k) = \frac{1}{2}(PK(SQL) + PK(MySQL)) = \frac{1}{2} * (100\% + 66{,}67\%) = 83{,}33\%$$

Ostatnim krokiem wykorzystania modelu alokacji zasobów jest oszacowanie pokrycia całego projektu na podstawie pokrycia kolejnych jego aktywności.

Tablica 8. Pokrycie aktywności projektu

Aktywność	Waga	Pokrycie aktywności (P1)	Pokrycie aktywności (P2)	Pokrycie aktywności (P3)
Identyfikacja aktorów, diagram kontekstowy	2	50%	33,33%	16,67%
Identyfikacja encji i klas	1	87,5%	58,33%	45,83%
Identyfikacja reguł biznesowych oraz ograniczeń dziedzinowych	1	91,67%	72,22%	38,89%
Transformacja do modelu logicznego (relacyjnego)	1	91,67%	63,89%	38,89%

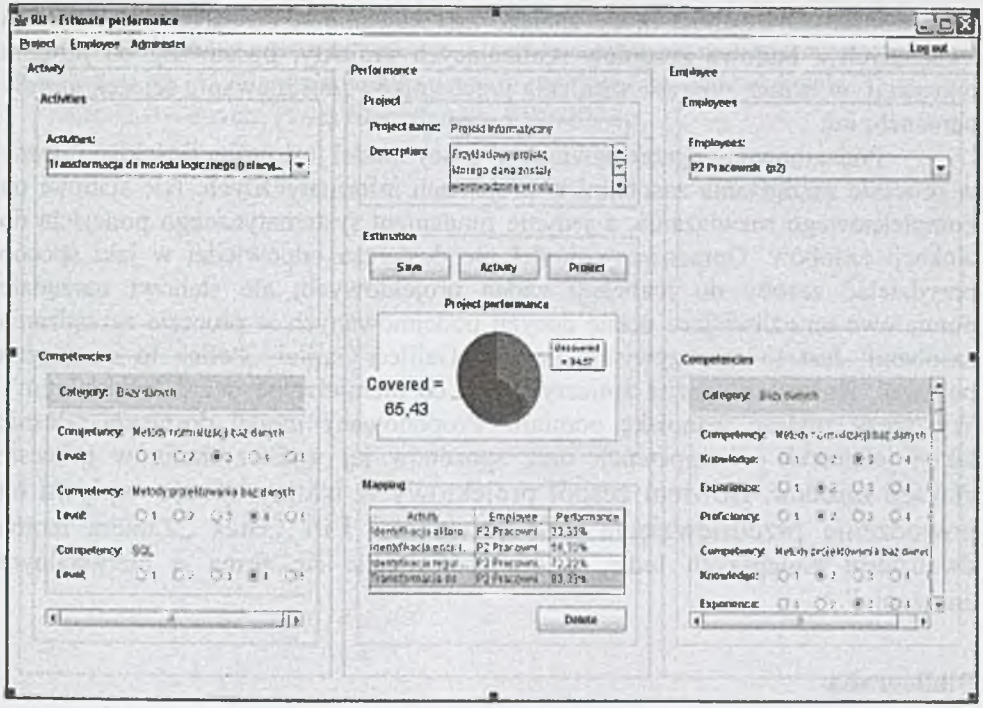
Transformacja do modelu fizycznego	1	**100%**	**83,33%**	**33,33%**
Pokrycie projektu		**87,96%**	**65,43%**	**36,73%**

Pokrycie projektu jest rozumiane jako średnia ważona pokryć wszystkich jego aktywności składowych, gdzie wagę stanowi priorytet zadania określający konieczność jego wykonania z punktu widzenia oczekiwań klienta.

Przyjmijmy chwilowo, że rozważany projekt składa się jedynie z 5 powyższych zadań, a do ich realizacji został przypisany pracownik P2.

$$PK(P)=\frac{\sum^{n} PK(A)*\frac{1}{w}}{\sum^{n}\frac{1}{w}}=\frac{\left(PK(A_1)*\frac{1}{2}+PK(A_2)+PK(A_3)+PK(A_4)+PK(A_5)\right)}{\left(\frac{1}{w_1}+\frac{1}{w_2}+\frac{1}{w_3}+\frac{1}{w_4}+\frac{1}{w_5}\right)}=65{,}43\%$$

Dane dotyczące realizacji projektu (uproszczonego) zostały wprowadzone do aplikacji będącej implementacją rozważanego modelu. Dane przedstawione w tabeli 8. pokazują jak waha się pokrycie projektu gdy do jego realizacji zostali przydzieleni różni pracownicy. Poniżej został dołączony widok okna aplikacji, w którym dokonywane jest szacowanie pokrycia projektu.



Rys. 5. Widok okna aplikacji implementującej model alokacji zasobów [12].

W części centralnej ekranu mamy graficznie przedstawiony poziom pokrycia kompetencji (ang. Projekt performance) w tym przypadku mamy pokrycie na poziomie 65,43% (ang. Covered) oraz braki tego pokrycia na poziomie 34,57% (ang. Uscovered)

Podsumowanie

Znaczna grupa komponentów procesu zarządzania przedsięwzięciami informatycznymi została już częściowo rozpoznana, uporządkowana, ustandaryzowana. Opracowano metody wskazujące jak poruszać się w poszczególnych obszarach zarządzania projektem IT[11]. Rozważana praca została poświęcona efektywnemu zarządzaniu wyborem i przydziałem zasobów ludzkich do aktywności projektowych, ponieważ zdaje się to być teren, bez satysfakcjonujących mechanizmów i uznanych praktyk. Przedstawione w pracy podejście jest próbą sformalizowania procesu alokacji zasobów. Pozwala ono na uporządkowany opis całego przedsięwzięcia w sposób, jednoznacznie wskazujący jakie kompetencje są niezbędne do jego realizacji. Umożliwia także opisanie zespołu projektowego, stanowiącego jeden z najważniejszych czynników sukcesu projektu, determinując tym samym jak delegować zasoby do zadań, by optymalizować proces realizacji przedsięwzięć oraz minimalizując ryzyko ich

niepowodzeń[2,4]. Model może znaleźć zastosowanie w obszarze wielu działań związanych z budową zespołów realizujących projekty, począwszy od procesu rekrutacji w firmie, poprzez szkolenia pracowników, definiowanie ścieżek kariery personelu itd.

Podsumowując, przedstawiony w pracy model jest próbą ilościowej oceny w procesie zarządzania zasobami w projektach informatycznych. Nie stanowi on kompleksowego rozwiązania, a jedynie fundament systematycznego podejścia do alokacji zasobów. Opracowany model nie dostarcza odpowiedzi w jaki sposób przydzielać zasoby do realizacji zadań projektowych, ale stanowi narzędzie pomiarowe umożliwiające ocenę decyzji podejmowanych w procesie zarządzania zasobami. Jest to urzeczywistnienie idei Galileo Galilei „Policz to co można policzyć, zmierz co można zmierzyć, a to co niemierzalne uczyń mierzalnym". Wystarczy znaleźć jednostkę pomiaru. Proponowany model dostarcza właśnie takiej jednostki – kompetencji oraz sposobów jej wykorzystania w procesie alokacji zasobów. Bowiem zespół projektowy w istotny sposób wpływa na powodzenie przedsięwzięcia. Jak powiedział Bill Gates „Zabierz moich dwudziestu najlepszych ludzi, a Microsoft stanie się firmą o marginalnym znaczeniu".

Bibliografia

1. Czarnacka-Chrobot B., *Pomiar rozmiaru funkcjonalnego systemu informatycznego – sposób na "chaos permanens?"* W: Współczesne kierunki rozwoju informatyki, PTI, Katowice-Mrągowo, 2004
2. Frączkowski K., Zarządzanie Projektem Informatycznym". Wydawnictwo PWr, Wrocław, 2002.
3. Frączkowski K., *Modele zarządzania zasobami projektu informatycznego i organizacja zespołów – telepraca*, WNT, Warszawa 2003
4. Frączkowski K., *Model of mapping activities and competence ICT Project.* W. Annales Informatica, 2005
5. Kaczmarek T., *Modele zarządzania wiedzą*, Gazeta IT, październik 2005, nr 9
6. Kircher M., Jain P., *Pattern-Oriented Software Architecture: Patterns for Resource Management*, John Wiley & Sons, 2004
7. PMI Standards Committee, *A Guide to the Project Management Body of Knowledge,* Project Management Institute, 1996
8. Standish Group, *2004 Third Quarter Research Report*, 2004 http://www.standishgroup.com/sample_research/PDFpages/q3-spotlight.pdf (20.03.2006)
9. Standish Group, *Extreme CHAOS*, 2001 http://www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf (20.03.2006)
10. Standish Group, *The CHAOS Report (1994)*, 1995 http://www.standishgroup.com/sample_research/chaos_1994_4.php (20.03.2006)

11. Strona domowa Project Management Institute http://www.pmi.org/ (15.01.2006)
12. Sidorczyk A., Mapowanie kompetencji i aktywności w projektach informatycznych. Praca Magisterska, PWr.2006 r.

ROZDZIAŁ IV

ZAGADNIENIA WDRAŻANIA ZINTEGROWANYCH SYSTEMÓW ZARZĄDZANIA PROJEKTAMI W FIRMIE INFORMATYCZNEJ

Jan WEREWKA

1. Wstęp

Projekty informatyczne należą do przedsięwzięć, dla których stopień ryzyka i niepowodzenia jest bardzo wysoki. Spowodowane to jest wieloma przyczynami, do których należą między innymi: nieprecyzyjna specyfikacja wymagań, brak odpowiedniego doświadczenia i wiedzy personelu realizującego projekt, zastosowanie niewłaściwych metod i narzędzi wytwarzania oprogramowania, itp. Problematyka ta jest szczególnie istotna dla małych i średnich przedsiębiorstwach informatycznych (MSPI), w których często jest brak oddzielnego zespołu zajmującego się jedynie zagadnieniami zarządzania projektami informatycznymi, zarządzaniem jakością w procesie wytwarzania oprogramowania. Opracowane metody dla dużych przedsięwzięć informatycznych nie mogą być bezpośrednio wykorzystane w tego typu przedsiębiorstwach.

Na rynku gospodarczym działa wiele firm informatycznych, które możemy uważać za małe i średnie. Są to firmy liczące według ustawy od 5 do 250 pracowników. Mała firma informatyczna ma do 50 pracowników. Dodatkowym wymogiem jest by takie firmy nie były zależne finansowo od dużych firm. Oznacza to, że jeśli istnieje firma informatyczna zatrudniająca kilkanaście pracowników, a udział kapitałowy dużej firmy jest znaczny, to firma pierwotna nie będzie traktowana już jako mała. W dużych firmach informatycznych lub w dużych przedsięwzięciach informatycznych stosuje się zwykle zweryfikowane metody wytwarzania oprogramowania związane z ich długoletnim doświadczeniem. Firmy takie dysponują służbami wspomagającymi wytwarzanie oprogramowania, nie zaangażowanymi bezpośrednio w realizowanych projektach rynkowych.

MSPI posiadają szereg własności odróżniających je od innych firm:

- Płaska struktura sterowania.
- Wszechstronność wykonywanych prac przez grupy robocze (projektowanie, dokumentowanie, kodowanie, testowanie i wdrażanie).
- Działalność ściśle uzależniona od sytuacji rynkowej.
- Ograniczenia finansowe niepozwalające na stosowanie drogich renomowanych narzędzi informatycznych.
- Brak specjalistów z różnych dziedzin. Dysponowanie specjalistami z wąskiego zakresu.
- Brak dużych doświadczeń związanych z całym cyklem wytwarzania oprogramowania.

- Uczestnictwo pracowników w kilku projektach jednocześnie.

Działające w Polsce małe i średnie firmy informatyczne powstały głównie dzięki pomysłom innowacyjnym ich założycieli. Rozwój tych firm jest często hamowany na pewnym etapie z powodu braku dobrze zorganizowanego procesu wytwarzania oprogramowania. Kierownictwo tych firm często zdaje sobie sprawę z potrzeby zmian w organizacji procesu wytwarzania, ale przeważnie nie dysponuje wiedzą jak te zmiany zrealizować. Wykorzystanie własnej załogi jest trudne, gdyż firmy te bazują na specjalistach, których wiedza dotyczy tylko części obszaru wytwarzania oprogramowania. Próby realizacji tych zmian są zatem celowe, lecz rozciągają się w czasie i nie ma pewności co do właściwego doboru rozwiązań, ze względu na brak środków na przeprowadzenie dokładniejszych analiz. Wdrożenie jednorazowe właściwych metod wytwarzania oprogramowania jest możliwe zazwyczaj poprzez wykorzystanie instytucji zewnętrznej. Rozwiązanie takie nie zawsze kończy się powodzeniem, gdyż proponowane rozwiązania mają raczej charakter administracyjny.

Jednym z istotnych zagadnień dla MSPI jest rozpoczęcie zmian procesów wytwarzania oprogramowania umożliwiających sukcesywną poprawę jakości. Dlatego ważnym zagadnieniem jest uzyskanie w określonej perspektywie czasowej pewnych standardów jakościowych, co oprócz prestiżu firmy miałoby istotne znaczenie polegające na umożliwieniu firmie dalszej ekspansji na rynku. Należy podkreślić, że zastosowanie nowych rozwiązań musi przynieść wymierne efekty i być zaakceptowane przez grupy projektowe. Akceptacja ta polega na tym, że uzyskane efekty muszą być większe niż wysiłek związany z utrzymaniem wyższej jakości.

2. Zintegrowane systemy zarządzania projektami

Jak wiadomo szybkiej zmianie ulegają środowiska pracy związane z wytwarzaniem oprogramowania. Dla większości małych firm nie stanowi większego problemu utrzymywanie lokalnej sieci komputerowej, posiadającej połączenie ze światem oraz budowa serwerów utrzymujących wspólne narzędzia programistyczne i bazy danych. Sytuacja ta implikuje szukanie nowych rozwiązań, a praca grupowa w Intranecie może znacznie ułatwić realizację zintegrowanego systemu zarządzania. W związku z tym za uzasadnione należy przyjąć założenie, że dostęp do zintegrowanego systemu zarządzania będzie realizowany poprzez Intranet z każdego stanowiska roboczego.

W związku z powyższym celowe jest tworzenie w tych firmach zintegrowanych systemów informatycznych wspomagających procesy zarządzania i wytwarzania oprogramowania, nazywanych w dalszej części krótko **systemami zintegrowanymi**. System zintegrowany powinien spełnić kilka podstawowych warunków w odniesieniu do MSPI, są to między innymi:

- Niski koszt systemu (akceptowalny koszt narzędzi wchodzących w skład systemu).
- Efektywne wspomaganie procesów wytwórczych.

- Wspomaganie wytwarzania oprogramowania zgodnie z docelowymi standardami jakości.
- Stosowane narzędzia powinny mieć przyjazne interfejsy, z zakres ich funkcjonalności być odpowiedni dla MSPI.
- Koszty eksploatacji i pielęgnacji systemu powinny być akceptowalne.
- Narzędzia wchodzące w skład systemu powinny się łatwo integrować ze sobą.

Określenie zasad i podstaw konstrukcji takich systemów jest dużym wyzwaniem, gdyż nie należy się spodziewać wspomagania tego typów rozwiązań przez duże firmy oferujące drogie i kompleksowe (ale zazwyczaj nie w pełni) rozwiązania. Badania w tym zakresie mogłyby być interesujące dla instytucji naukowych, ze względu na brak większych analiz i rozwiązań w tym zakresie. Dodatkowo prace dotyczące wymienionych zagadnień ułatwiłoby wypracowanie otwartych koncepcji fabryk oprogramowania realizowanych w mniejszych firmach, co spowodowałoby wzrost konkurencyjności polskich firm.

Celowość budowy systemów zintegrowanych

W literaturze i w Internecie można znaleźć cały gąszcz opisów różnych rozwiązań dotyczących spomagania wytwarzania oprogramowania. Powstaje pytanie czy nowe propozycje i rozwiazania mogą wnieść coś nowego i są uzasadnione. Należy zdać sobie sprawę, że rozwiązania proponowane kilka lat temu szybko się dezaktualizują. Wynika to z stosowania nowych technologii z których oprogramowanie jest zbudowane oraz nowych środowisk, w którym takie oprogramowanie jest zainstalowane. Również w procesie wytwarzania stosowane są nowe metody i technologie. Doskonalone są i podlegają specjalizacji metody wytwarzania oprogramowania.

W związku z zmianami technologicznymi oraz ze względu na olbrzymie znaczenie ekonomiczne i strategiczne, celowe jest prowadzenie prac w tym zakresie. Jednostki badawcze w kraju (tak wynika to z analizy materiałów konferencyjnych KKIO) realizują głównie adaptacje zaproponowanych metodyk opracowanych głównie w USA i Wielkiej Brytanii dla potrzeb projektów informatycznych. Rozwiązania te dotyczą głównie dużych firm, które zazwyczaj posiadają wypracowane swoje własne rozwiązania.

W Polsce działa duża liczba małych firm, dla których proponowane rozwiązania nie mają bezpośredniego zastosowania. Firmy te po uzyskaniu pewnego poziomu rozwoju, napotykają na trudności związane z problemami technologicznymi, kompetencji zasobów ludzkich i zarządzania projektami. W związku z tym celowe jest opracowanie efektywnych metod wytwarzania i zarządzania w takich firmach. W tym celu konieczna jest analiza istniejących rozwiązań oraz opracowanie metod i modeli wykorzystywanych przy budowie zintegrowanych systemów zarządzania dla takich przedsiębiorstw.

3. Przegląd rozwiązań istotnych przy tworzeniu systemów zintegrowanych

Projekty informatyczne obarczone są dużym ryzykiem, z tego powodu od wielu lat opracowywane są metody zwiększenia efektywności procesów

projektowych poprzez opracowanie różnych metodologii. Powstaje pytanie, czy w sytuacji, gdy istnieją setki różnych metodologii, możliwe jest opracowanie nowej propozycji. Zatem przy wyborze rozwiązania celowe jest dokonanie przeglądu istniejących rozwiązań i dostosowanie ich dla potrzeb małych i średnich firm ze szczególnym uwzględnieniem tych procesów, które dadzą największe efekty. Odwzorowanie to zostanie dokonane z myślą o zintegrowanym komputerowym systemie zarządzania.

Przy zarządzaniu wytwarzaniem oprogramowania mogą być wykorzystane różne szkielety postępowania (ang. frameworks), opracowane z myślą o różnych obszarach zastosowania:

1. Standardy i zalecenia (ang. guidelines):

System ISO zarządzania jakością. System ISO 9000/9001 należy do najbardziej rozpowszechnionych i akceptowanych standardów na całym świecie. Posiada też rozpowszechnioną sieć instytucji przyznających certyfikat jakości. ISO jest szczególnie atrakcyjne z punktu widzenia firm informatycznych, gdyż jest możliwe do osiągnięcia przy ograniczonych kosztach w stosunkowo w krótkim czasie. Jednym z głównych problemów jest własna definicja procesów zarządzania. Wymienione procesy wejdą w skład systemu zarządzania jakością SZJ zgodnie z normą ISO 9001:2001. W systemach identyfikuje się następujące procesy główne: *procesy organizacyjne* – zarządzanie całym systemem i jego doskonalenie, *procesy operacyjne* – procesy wchodzące w skład cyklu wytwarzania oprogramowania, *procesy pomocnicze* – procesy stwarzające warunki do poprawnej realizacji procesów zarządzania jakością i procesów operacyjnych. Standard ten ma charakter ogólny i może być zastosowany do dowolnych systemów wytwarzania. W celu ułatwienia firmom informatycznym wdrożenia tego standardu, próbowano opisać sposób jego zastosowania w firmach informatycznych. Dla przykładu rząd brytyjski zlecił opracowanie schematu (nazywanego programem TickIT) oznaczonym jako ISO9000/BS5750 interpretujący normę ISO dla sektora IT [23].

ISO 12207. Standard ISO 12207 dotyczy całego cyklu rozwoju oprogramowania (ang. Standard for Software Lifecycle Processes).

ISO 15504:2004 opracowany w roku 2004 zawiera metodyki i struktury i oceny dla procesów wytwarzania oprogramowania. Sposób oceny przedstawiony jest za pomocą modelu SPICE (Software Process Improvement and Capability Determination).

CMM (Capability Maturity Model). Jednym z bardziej znanych standardów jest model CMM. Zawiera standardy, które mają bezpośrednie zastosowanie dla wytwarzania oprogramowania. Celem CMM jest wskazanie sposobów polepszenia poziomu dojrzałości wytwarzania oprogramowania. Przejawia się to przez definiowanie pięciu poziomów dojrzałości wytwarzania. Firma informatyczna może doskonalić proces wytwarzania oprogramowania poprzez uzyskiwanie kolejnych poziomów. W ostatnim okresie powstała nowa wersja CMM zwana CMMI (CMM Integration). CMMI zawiera więcej informacji oraz bardziej szczegółowe wskazówki dotyczące spełnienia warunków

poszczególnych poziomów dojrzałości.

Opracowywane są także zalecenia (instrukcje), które opisują jak należy postępować w procesach wytwarzania oprogramowania. Opracowania w tym zakresie wspierają instytucje rządowe lub wojskowe. Przykładem może być GSAM (Guidelines for Succesful Acquisition and Management of Software Intensive Systems) [5], który w przejrzystej formie przedstawia zalecania, które należy stosować przy tworzeniu systemów wyposażonych w oprogramowanie.

2. Certyfikacje, wyróżnienia i oceny

Ważnym zagadnieniem jest ustalenie dla firmy informatycznej stopnia poprawności działania w zakresie wytwarzania oprogramowania. Ma to duże znaczenie, gdyż umożliwia określenie aktualnego stanu i kierunków rozwoju firmy, przyczynia się do prestiżu firmy w przypadku spełnienia standardów, umożliwia zdobycie kontraktów w przypadku lepszej ich oceny w porównaniu z innymi firmami. Firmy informatyczne zwykle ubiegają się o przyznanie standardów ISO i CMM, w tym celu poddają się odpowiedniemu procesowi certyfikacji.

W celu poprawy jakości funkcjonowania firm, rządy niektórych krajów tworzą programy wyróżnień (nagród) dla firm, które spełniają określone kryteria. Przykładowym takim wyróżnieniem jest Malcom Baldrige National Quality Award przyznawany corocznie kilku firmom amerykańskim.

Warto również zaznaczyć, że firmy zlecające zamówienia związane z projektami informatycznymi, mogą żądać od wykonawców poddania się ocenie. Przykładami sposobów porównania firm jest związany z CMM system ocen zwany SCE (Software Capability Evaluation), metoda oceny SCAMPI (Standard CMMI Assessment Metod for Process Improvement) dotycząca CMMI lub związany z Siłami Powietrznymi USA system ocen SDCE (Software Development Capability Evaluation).

3. Modele wytwarzania oprogramowania (modele inżynierii oprogramowania dla całego cyklu życia). Dla procesów wytwarzania oprogramowania opracowano szereg metodyk, których reprezentatywne przykłady wymieniono poniżej.

RUP. Jednym z bardziej znanych metodyk wytwarzania oprogramowania jest RUP (Rational Unified Process) [20]. Jest to metodyka iteracyjna wytwarzania oprogramowania. Metodyka ta jest wspomagana przez oprogramowanie aktualnie oferowane przez firmę IBM.

MSF. Microsoft proponuje dla projektów informatycznych metodykę MSF (Microsoft Solution Framework) [10, 11, 12]. Jest ona zaliczana jest do metodyk zwinnych i jest w pewnym zakresie wspomagana przez oprogramowanie VSTS (Microsoft Visual Studio Team System).

4. Również dla zarządzania projektami (niekoniecznie informatycznymi) wybrano i opracowano efektywne metody zarządzania. Przykładami takich rozwiązań podano poniżej.

PMBOK (Project Management Body of Knowledge). Jednym z takich bardziej znanych rozwiązań jest opracowany przez PMI (Project Management

Institute) zbiór o nazwie PMBOK [16] zawierający sprawdzone praktyki stosowane w zarządzaniu projektami. PMBOK nie jest ukierunkowany na projekty informatyczne, lecz ma bardziej ogólny charakter. Zaletą PMBOK jest przedstawienie w sposób przejrzysty procesów związanych z zarządzaniem projektem. W PMBOK definiuje się dziewięć obszarów zarządzania projektami.

PRINCE (PRoject in Controlled Environment). Innym rozwiązaniem jest PRINCE, który jest metodą zarządzania projektami o dowolnej wielkości, opracowaną w Wielkiej Brytanii. W pracy [15] przedstawiono pewną wstępną propozycję wspomagania wytwarzania oprogramowania wykorzystującą tę metodykę, realizowaną w ramach grantu KBN.

Do oceny jakości zarządzania projektami określone są także modele dojrzałości PMMM (Project Management Maturity Model). Poziom 2 dotyczy określenia wspólnych procesów, a poziom 3 wprowadza pojedynczą metodologię (opisane w [9]).

Zintegrowane zarządzanie

Aktualnie trwają prace nad integracją zarządzania wszystkich obszarów związanych z wytwarzaniem oprogramowania. Przykładem mogą być próby odwzorowania metod związanych z inżynierią wytwarzania na metody zarządzania PMBOK i RUP.

CMMI (Capability Maturity Model Integration) [3, 1] jest rozszerzeniem bazującym na pięciu poziomach CMM. CMMI jawnie opisuje powiązania pomiędzy procesami wytwarzania oprogramowania jak i zarządzania, lepiej powiązany jest z standardem ISO. Firmy starają się przechodzić z CMM do CMMI. Zdefiniowane zostały szczegółowe modele CMMI przeznaczone między innymi dla inżynierii systemowej (CMMI for System Engineering), inżynierii oprogramowania, (CMMI for Software Engineering) oraz dla zintegrowanego rozwoju procesu i produktu (CMMI for Integrated Product and Process Development).

Przegląd narzędzi

Na rynku dostępnych jest szereg narzędzi wspomagających różne obszary zarządzania i wytwarzania oprogramowania. W opracowaniu określone zostaną zasady doboru narzędzi wspomagających. Wybór narzędzi i uznanie ich jako standardy w MSPI powinny odegrać ważną rolę porządkującą procesy wytwarzania.

Wybór narzędzi, które mogłyby być potencjalnymi składowymi systemu zintegrowanego jest bardzo duży dla różnych obszarów zarządzania i wytwarzania oprogramowania. Przykładowo zostaną wymienione niektóre obszary i narzędzia:

- W zakresie organizacji projektów, planowania i sterowania istnieją setki różnych narzędzi. Dobrym punktem wyjściowym do rozpoczęcia przeglądu jest opracowanie [18].
- Zarządzanie wymaganiami jest dobrze rozpoznana dziedziną. W opracowaniu [25] zawarto przegląd kilkudziesięciu narzędzi przeznaczonych do zarządzania wymaganiami.
- Zarządzanie zmianami (opisana lista narzędzi w [2]) jest realizowane przez

oddzielne narzędzia lub jako narzędzia wchodzące w skład narzędzi do zarządzania konfiguracją oprogramowania.

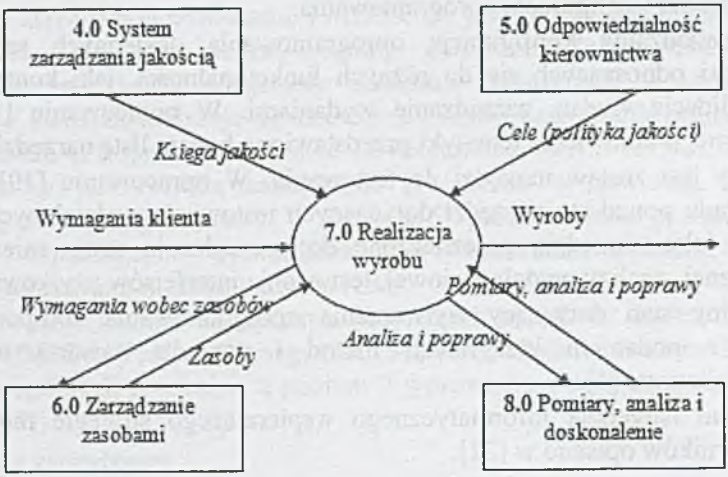
- Do zarządzania konfiguracją oprogramowania dostępnych są dziesiątki narzędzi odnoszących się do różnych funkcjonalności, jak: kontrola wersji, konsolidacja wydań, zarządzanie wydaniami. W opracowaniu [17] oprócz obszernego omówienia tematyki przedstawiono bogatą listę narzędzi.
- Bogaty jest zestaw narzędzi do testowania. W opracowaniu [19] dokonano przeglądu ponad stu narzędzi dotyczących testowania podzielonych na różne grupy jako narzędzia przeznaczone do: zarządzania testowaniem, analizy statycznej, analizy wydajnościowej, testowania interfejsów użytkownika itd.
- Aktualny stan dotyczący wytwarzania oprogramowania komponentowego, wraz z podaniem klasyfikacji metod i narzędzi zawarto w artykule przeglądowym [4].
- Przykład narzędzia informatycznego wspierającego strategie motywowania pracowników opisano w [21].

4. Wykorzystanie podobieństw CMMI i ISO 9001:2000 przy konstrukcji systemu zintegrowanego

Przy konstrukcji systemu zintegrowanego warto się oprzeć o uznane standardy, a w szczególności na ISO 9001:2000 i CMMI będacym de facto standardem. Na podstawie analiz porównawczych uważa się, że wymagania ISO 9001:2000 znajdują się pomiędzy wymaganiami dla drugiego i trzeciego poziomu dojrzałości CMM. Z kolei wiadomo na przykładzie USA, że przy przyznawaniu kontraktów liczą się firmy (tzw. klasą A), które uzyskały poziom dojrzałości 3 i wyższy. Wydaje się, że dla MSPI najlepszym rozwiązaniem byłoby uzyskanie certyfikatu jakości ISO 9001:2000, a w następnym kroku poziomu trzeciego CMMI. Poziom trzeci CMMI (podobnie jak w CMM) związany jest z integrowaniem procesów zarządzania, a proponowane w opracowaniu podejście zakłada taką integrację. Ponadto warto zaznaczyć, że standardy ISO i CMMI (CMM) w okresie ostatnich lat zbliżyły się do siebie, dlatego niektóre firmy tworzą systemy zarządzania, spełniające obydwa standardy. Propozycje rozwiązań idących w tym kierunku przedstawiono np. w pracy [27]. Rozważanie wyższych poziomów CMMI nie jest na tym etapie celowe, gdyż ich wdrożenie dla MSPI może to prowadzić do znacznego wzrostu kosztów wytwarzania i do zahamowania mobilności działania takich firm, a więc do utraty własności związanych z ich przewagą wobec innych firm informatycznych.

System ISO może być stosowany do różnych firm o różnych profilach produkcyjnych. Proces wytwarzania, a dla firm informatycznych jest to wytwarzanie oprogramowania zawarty jest w procesie opisanym w punkcie 7 jako realizacja wyrobu. Wejściami do tego procesu są księga jakości opracowywana w procesie 4.0 zwanym systemem zarządzania jakością oraz cele podawane przez proces 5.0 dotyczący odpowiedzialności kierownictwa. Z realizacją wyroby powiązany jest proces 8.0 dotyczący pomiarów, analizy i doskonalenia. Podobnie

proces 6.0 zarządzania zasobami jest ściśle powiązany z realizacja wyrobu.



Rys. 1. Interakcje pomiędzy procesami ISO

W systemie CMMI definiowane są procesy, które można podzielić na cztery podstawowe obszary (tab. 1) zarządzania procesowego, zarządzania projektem, inżynierii oprogramowania i zarządzania pomocniczego. Dla każdego procesu podano powiązanie z poziomem dojrzałości firmy informatycznej.

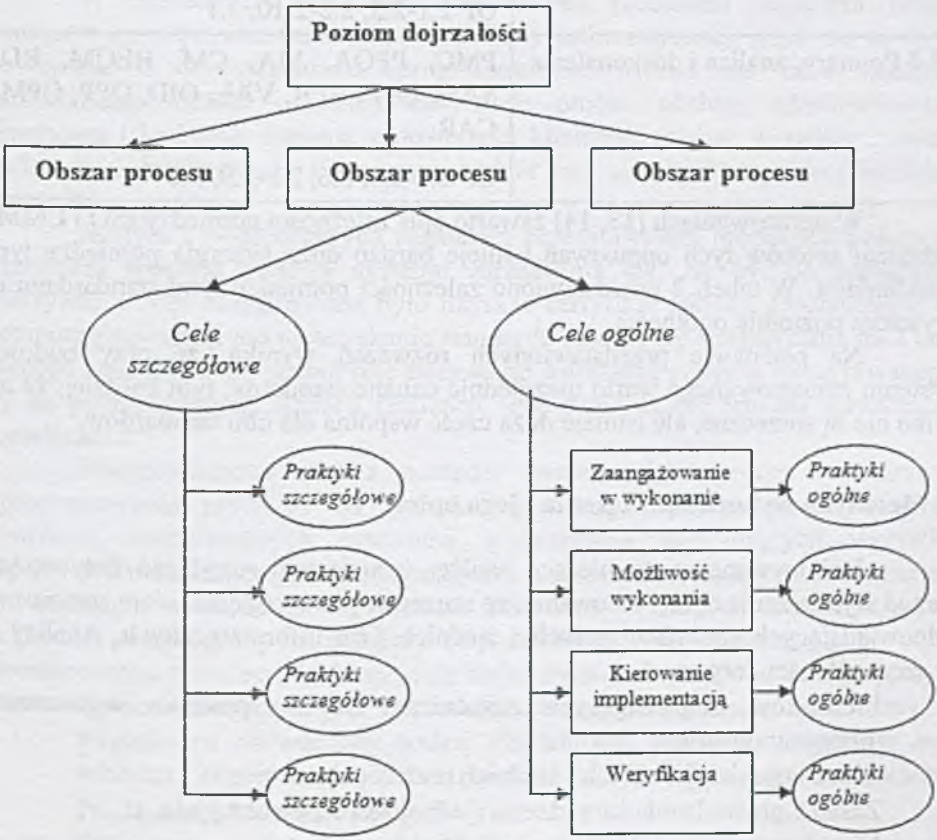
Tablica 1. Procesy w CMMI (pogrubioną czcionką oznaczone zostały procesy 2 i 3 poziomu dojrzałości)

Zarządzanie procesowe	
Definicja procesów organizacji	3 - OPD
Zorientowane na procesy organizacji	3 - OPF
Szkolenie w organizacji	3 - OT
Wydajność procesów organizacji	4 - OPP
Innowacje organizacyjne i wdrożeniowe	5 - OID

Zarządzanie projektem	
Planowanie projektu	2 - PP
Monitorowanie i kontrola projektu	2 - PMC
Zarządzanie uzgodnieniami z dostawcami	2 - SAM
Zintegrowane zarządzanie projektem	3 - IPM
Zarządzanie ryzykiem	3 – RSKM
Zintegrowanie grup	3 – IT
Zintegrowane zarządzanie uzgodnieniami z dostawcami	3 – ISM
Ilościowe zarządzanie projektem	4 – QPM

Zarządzanie wymaganiami	2 – REQM
Rozwój wymagań	3 – RD
Rozwiązania technologiczne	3 – TS
Integracja produktu	3 – PI
Weryfikacja	3 – VER
Walidacja	3 – VAL

Zarządzanie konfiguracją	2 - CM
Zapewnienie jakości procesu i produktu	2 - PPQA
Pomiary i analiza	2 - PM
Analiza decyzyjna i rozwiązywanie problemów	3 - DAR
Środowisko organizacyjne do integracji	3 - OEI
Analiza przyczynowa i rozwiązywanie problemów	5 - CAR



Rys. 2. Reprezentacja w CMMI

Dla każdego procesu w CMMI definiuje się cele szczegółowe typowe dla

takiego procesu i cele ogólne związane z zapewnieniem warunków do realizacji tego procesu.

Tablica 2. Zależności pomiędzy ISO i CMMI

ISO	CMMI
4.0 System zarządzania jakością	OPF, OPD, PP, PPQA, CM, SAM GP 2.1-2.3, 2.6-2.9, 3.1-3.2
5.0 Odpowiedzialność kierownictwa	OPF, OPD, RD, PMC, OPP, QPM GP 2.1-2.4, 2.6-2.7, 2.10, 3.1
6.0 Zarządzanie zasobami	PP, OT, OEI GP 2.3, 2.5
7.0 Realizacja wyrobu	REQM, RD, TS, PI, MA, QPM, VER, VAL, OPD, PP, PMC, IPM, CM, SAM GP 2.1-2.5, 2.6-2.10, 3.1
8.0 Pomiary, analiza i doskonalenie	PMC, PPQA, MA, CM, REQM, RD, SAM, OPF, VER, VAL, OID, OPP, QPM, CAR GP 2.1-2.4, 2.6, 2.8-2.9, 3.2

W opracowaniach [13, 14] zawarto opis zależności pomiędzy ISO i CMMI. Zdaniem autorów tych opracowań istnieje bardzo duża synergia pomiędzy tymi standardami. W tabeli 2 przedstawiono zależności pomiędzy tymi standardami na wysokim poziomie ogólności.

Na podstawie przedstawionych rozważań wynika, że przy budowie systemu zintegrowanego warto uwzględnić uznane standardy, tym bardziej, że nie tylko nie są sprzeczne, ale istnieje duża część wspólna dla obu standardów.

5. Metodyka wyboru rozwiązania i jego opisu

Dokonywane są na bieżąco analizy istniejących rozwiązań dotyczących metod wytwarzania oprogramowania, ze szczególnym uwzględnieniem zastosowań odpowiadających specyfice małych i średnich firm informatycznych. Analizy te w szczególności dotyczą:

- Elementów odgrywających zasadniczą rolę w procesie wytwarzania oprogramowania.
- Zasad organizacji małych i średnich przedsiębiorstw.
- Zasad wprowadzania zarządzania (jednocześnie, ewolucyjnie..).
- Metod etapowego wdrożenia podsystemów.
- Zasad integracji poszczególnych obszarów zarządzania.
- Sposobów zarządzania jakością.

Przy wyborze rozwiązań należy pamiętać, że nie mogą być utracone zalety małych firm, w tym głównie mobilność, która może być rozumiana jako: elastyczność na wprowadzane zmiany, szybkość wdrażania podstawowej części systemu, łatwość adaptacji oprogramowania do zmiennej architektury systemu, łatwość tworzenia prototypu systemu bazując na różnych dostępnych komponentach i narzędziach. Dodatkowo istotne jest poszukiwanie rozwiązania o ograniczonych kosztach oraz uwzględnienie braku możliwości oddelegowania dużego zespołu do wdrożenia nowo tworzonego systemu zarządzania i wytwarzania oprogramowania.

Wybór obszarów zarządzania

Na bazie przeprowadzonych analiz dokonany zostanie wybór pewnej liczby obszarów wytwarzania oprogramowania, ważnych z punktu widzenia potrzeb i efektywności funkcjonowania.

W małych i średnich przedsiębiorstwach należy skoncentrować się na najważniejszych zagadnieniach. W związku z tym należy określić, jakie składowe w procesie zarządzania będą odgrywać najważniejszą rolę.

W opracowaniu nie zajęto się innymi procesami mającymi istotne znaczenie na poprawne funkcjonowanie firmy informatycznej, gdyż nie są ściśle powiązane z wytwarzaniem oprogramowania. Można do nich zaliczyć: pozyskiwanie klienta i zasady zawierania umów; obsługę administracyjną, finansową i kadrową, badania zadowolenia klientów, zakupy wyrobów i usług, nadzór nad dokumentami i zapisami, nadzór nad infrastrukturą i środowiskiem pracy, itd.

Jednym z podstawowych zagadnień przedstawionej propozycji polega na określeniu sposobu budowy systemu zarządzania tak, by był on najbardziej efektywny i przy okazji można było uzyskać certyfikat jakości. Dochodzenie do poziomu pozwalającego na uzyskanie standardu szacuje się statystycznie na 2 lata. Dlatego ważne są takie propozycje etapowego wdrażania systemu zintegrowanego, by na poszczególnych etapach realizacji tego projektu nie istniała sprzeczność rozwiązań.

Przeprowadzona analiza narzędzi wspomagająca proces wytwarzania oprogramowania prowadzi do wniosku, że chociaż podejmowane są próby tworzenia zintegrowanych systemów wytwarzania obejmujących wszystkie aspekty wytwarzania, to jednak dla małych i średnich firm lepszym rozwiązaniem jest dokonanie wyboru odpowiednich narzędzi dla różnych podsystemów, a potem dokonanie pewnej próby ich integracji na poziomie przedsiębiorstwa.

Podsumowując powyższą dyskusję, cele badań można sformułować następująco:

- Opracowanie kryteriów pozwalających na identyfikację typu firmy ze względu na prowadzony rodzaj działalności, strukturę produktów oraz schemat organizacyjny (bazujących na metodzie opisu Macierzy Produktów i Procesów [24]).
- Stworzenie metamodelu określającego procesy, aktywności wewnątrz procesów, stany procesów, powiązania pomiędzy procesami, użyte narzędzia, artefakty oraz aktorów.

- Opracowanie wytycznych pozwalających na stworzenie modelu procesów odpowiadających zidentyfikowanemu typowi firmy. Wytyczne powinny także uwzględniać, jakie zmiany w systemie organizacji powinny być dokonywane przy zmianie rodzaju wytwarzanych produktów wyjściowych.
- Opracowanie katalogu wzorców procesów odpowiadających zidentyfikowanej strukturze organizacyjnej i strukturze produktów firmy.
- Zaproponowanie metod formalnych umożliwiających walidację modelu bazujących na odpowiednich modelach.
- Opracowanie koncepcji elastycznego (konfigurowalnego) systemu wspomagającego zarządzaniem procesu wytwarzania oprogramowania oraz dobór narzędzi.

6. Dodatkowe założenia dotyczące wdrażania systemu zintegrowanego

Zarządzanie projektem, a zarządzanie firmą informatyczną

W literaturze podaje się szereg efektywnych metod zarządzania i wytwarzania oprogramowania. Większość opracowanych metodyk odnosi się odpowiednio do:

- *Grupy projektowej*. Jest to typowe podejście, w którym analizuje się klasę projektu i w zależności od jego rodzaju określa się sposób realizacji projektu.
- *Grupy roboczej* pracującej nad rodziną oprogramowania. Grupa robocza realizuje podobne tematycznie projekty (ten sam asortyment produktów programowych).
- *Pojedynczych osób* w celu określenia rozwoju umiejętności i przydziału ról projektowych.
- *Całej firmy*, w której grupy robocze mogą realizować różnorodne projekty informatyczne.

Celem niniejszego opracowania ma być wspomaganie całej firmy informatycznej w zakresie zarządzania i wytwarzania oprogramowania. Zintegrowany system ma stworzyć dla takiego przedsiębiorstwa zdolność efektywnej realizacji projektów przy założonym standardzie jakości wytwarzania i zarządzania. Dlatego ważne jest opracowanie bardziej ogólnych metodyk, które charakteryzowałyby się spójnością i jednolitym sposobem wykorzystania. W wyniku prac powinno powstać wspólne podejście do wszystkich projektów. Natomiast specyfika dotycząca projektu powinna być udokumentowana w sposób jednolity dla wszystkich projektów.

Opracowane rozwiązania muszą uwzględnić specyfikę rozpatrywanych firm i ich szybkie reagowanie na potrzeby klienta. Oznacza to, że w proponowanych rozwiązaniach muszą być uwzględnione zwinne metodyki wytwarzania oprogramowania. Następnym ważnym zagadnieniem przyjęcie założenia o wytwarzaniu oprogramowanie z wykorzystaniem komponentów. Powoduje to konieczność stosowania pewnych rozwiązań typowych dla inżynierii systemów. Zaproponowane rozwiązania odnosić się będą do całej firmy, lecz winny mieć organizację pro-projektową, muszą dodatkowo wspomagać pracę grup roboczych i projektowych oraz poszczególnych członków grup projektowych.

Dla każdego obszaru zarządzania przewiduje się ewolucyjny sposób wdrażania. Zakłada się opracowanie założeń oraz modeli dla zintegrowanego systemu, który by wspomagały wytwarzanie oprogramowania. Ważne jest, aby taki system był prosty dla zastosowań w tego typu firmach oraz by uwzględniał stan finansowy takich firm. Zakłada się, że system zintegrowany powinien być rozwiązaniem Intranetowym zbudowanym w oparciu o różne szeroko dostępne narzędzia związane z obszarem zarządzania i wytwarzania oprogramowania. Wybór narzędzi od rożnych producentów związany jest z koniecznością uzyskania niskich kosztów systemu, możliwością ich dobrania do potrzeb rozważanych firm, koniecznością stopniowego wdrażania wybranych obszarów. Wyzwaniem dla takiego rozwiązania jest opracowania metod integracji poszczególnych narzędzi oraz zasad konstrukcji systemu informatycznego włączającego wybrane narzędzia i uzupełnionego o brakujące funkcjonalności. Wiadomo, że duże zintegrowane systemy wspomagające wytwarzania oprogramowania mają trudności wdrożeniowe na wskutek ich złożoności, braku uwzględnienia specyfiki firm oraz ze względu na wysokie koszty.

Rozpatrując charakterystyki MSPI, wydaje się za celowe opracowanie metod zarządzania wytwarzaniem oprogramowania z uwzględnieniem specyfiki mniejszych firm informatycznych. Podstawą do tworzenia takiego systemu jest wybór dostępnych na rynku narzędzi i włączenie ich do systemu zintegrowanego. Jednym z niedocenianych, ale bardzo ważnych zagadnień jest wprowadzenie i udokumentowanie pewnych standardów dla firm informatycznych. Dobrze określone i skonfigurowane standardy w firmie porządkują wiele działań.

Standardowe rozwiązania proponowane dla firm są często niedoceniane, a mogą odegrać znaczącą rolę porządkującą w procesach wytwarzania. Standardy mogą mieć postać:

1. Dokumentów określających sposoby realizacji procesów.
2. Dokumentów opisujących standardowe struktury, produkty, aktywności, itp.
3. Szablonów – standardowych dokumentów przeznaczonych do wypełnienia.
4. Standardowych narzędzi – zbiór standardowych narzędzi, które należy stosować w firmie.
5. List kontrolnych przeznaczonych do weryfikacji procesów lub produktów.

Bazując na przedstawionej metodyce oraz na podanych standardach, zakłada się, że firma informatyczna będzie w stanie samodzielnie na bazie przedstawionych opracowań lub przy pomocy konsultacji sporządzić następujące dokumenty wchodzące w skład księgi jakości, dotyczące poszczególnych, wymienionych niżej obszarów zarządzania.

1. Cel wdrożenia i kierunki rozwoju.
2. Opis procesów proponowanych do wdrożenia.
3. Plan wdrożenia (elementy obowiązkowe i opcjonalne).
4. Lista standardów (dla firmy, działu, grupy projektowej).
5. Lista standardowych narzędzi i sposobów ich wykorzystania.
6. Lista szablonów zapewniających, że informacje będą w sposób jednolity przedstawiane.
7. Listy kontrolne poprawności funkcjonowania procesu, poprawności

produktu.
8. Zasady kontroli procesów i zapewnienia, ze będą utrzymane podane założenia.
9. Wytyczne do ciągłego doskonalenia procesów.

7. Wnioski

W opracowaniu na podstawie dotychczasowych prac [6, 7,8] zawarto pewne przemyślenia związane z budową systemu zintegrowanego wspomagającego wytwarzanie oprogramowania, ważne z punktu widzenia małych i średnich firm informatycznych.

Proponowane rozwiązanie zakłada wybranie odpowiednich narzędzi z dużej liczby dostępnych, dla poszczególnych obszarów zarządzania oraz wykorzystanie ich w budowanym systemie zintegrowanym. Główne zadania będą polegały na określeniu zasad integracji narzędzi oraz podsystemów, które należy zbudować od nowa.

Dla każdego obszaru opracowane zostaną procesy i struktury danych związane z tymi procesami. Zaproponowane procesy powinny być w dokładnie opisane i przedstawione w sposób graficzny. Należy opracować szereg modeli przeznaczonych do opisu procesów związanych z funkcjonowaniem oprogramowania. Istnieje cała szeroka gama modeli do obrazowania funkcjonowania procesów. Na uwagę zasługuje język modelowania UML oraz jego rozwinięcie o nazwie SPEM (Software Process Engineering Metamodel) [22]. Istnieją metodyki, których procesy są opisane w SPEM, przykładowo należy do nich RUP (Rational Unified Process). Na uwagę zasługują modele przepływu pracy, szczególnie ze osiągnięto wysoki poziom standaryzacji i szerokie zastosowanie do modelowania procesów biznesowych [26].

Literatura

1. AHERN D. M., CLOUSE A., TURNER R.: *CMMI Distilled, A Practical Introduction to Integrated Process Improvement*, SEI Series in Software Engineering, Addison-Wesley, 2003, p. 305.
2. Change Management Learning Center, http://www.change-management-directory.com/software/web-based.html, 2005.
3. *Capability Maturity Model Integration (CMMI) Version 1.1,CMMI for System Engineering, Software Eng., Integrated Product and Process Development and Supplier Sourcing.* Carnegie Mellon SEI, CMMI Product Team, 2002, p. 724.
4. BENKEN G., HAMMERSCHALL U., BROY M., CENGARLE M. V., JURJENS J., RUMPE B., SCHOENMAKERS M.: *Componentware – State of the Art 2003*, Understanding Components Workshop, CUE Initiative, Venice, 2003, p. 44
5. *Guidelines for Successful Acquisition and Management of Software-Intensive systems.* Department of the Air Force, Software Technology Support Center,

2003.
6. WEREWKA J., PIWOWARCZYK M.: *Zagadnienie kontroli pracy w procesie wytwarzania oprogramowania.* Problemy i metody inżynierii oprogramowania. Wydawnictwa Naukowo-Techniczne, pp. 567-576.
7. WEREWKA J., SUJDAK M.: *Standaryzacja analiz wykonywanych prac w projektach informatycznych*, XII Konferencja Sieci i Systemy Informatyczne, Łódź 2004, Politechnika Łódzka, pp. 593-602.
8. SUJDAK M., WEREWKA J.: *Zagadnienie tworzenia szablonów w projektach informatycznych*, XII Konferencja Sieci i Systemy Informatyczne, Łódź 2004, Politechnika Łódzka, pp. 583-592.
9. KERZNER H.: *Project Management. A systems Approach to Planning, Scheduling and Controlling*, John Willey & Sons, Inc., 2002, p. 891.
10. *Microsoft Solution Framework, version 3.0 Overview,* Microsoft Solution Framework, White Paper, 2003, p. 28.
11. *MSF Process Model v. 3.1,* Microsoft Solution Framework, White Paper, 2002, p. 47.
12. *MSF Management Discipline v.13.1,* Microsoft Solution Framework, White Paper, 2002, p. 31.
13. Boris Mutafelija, Harvey Stromberg. ISO 9001:2000 CMMI v1.1 Mappings, 2003
14. Boris Mutafelija, Harvey Stromberg. Exploring CMMI-ISO 9001:2000 Synergy when Developing a Process Improvement Strategy. SEPG 2003 Conference.
15. NAWROCKI J., PAWŁOWSKI P., POSPIECH K.: *System Internetowy wspomagający wytwarzanie oprogramowania metodą PRINCE* , ZN Politach. Gdańskiej, 2003, p. 7.
16. *A Guide to the Project Management Body of Knowledge*, Project Management Institute, Pennsylvania USA, 2000, p. 216.
17. *Product Data Management and Software Configuration Management.* The Association of Swedish Eng. Industries, 2001, p. 181.
18. *Internet – Project Management References.* Project Management Web Sites of Interest, 2003, p. 58.
19. POHJOLAINEN P.: *Software Testing Tools* , Department of Computer Sci. and Applied Mathematics, Univ. of Kuopio, 2002, p. 79.
20. *Rational Unified Process,* IBM Rational Software, 2003.
21. SKARŻYŃSKA G.: *Rozwiązanie informatyczne wspierające strategie motywowania pracowników,* Oracle polska, 2005. p. 42.
22. *Software Process Engineering Metamodel Specification.* Object Management Group, Inc. 2002.
23. *TickIT, Guide to Software Quality Management System Construction and Certification Issue 3.0.,* Department of Trade and Industry and the British Computer Society, 1997.
24. WHEELWRIGHT S.C., *Managing New Product and Process Development: Text Cases*, Free Press, 1992
25. *Volere Requirement resources.* Requirements tools,

http://www.volere.co.uk/tools.html, 2005.
26. *Workflow Process Definition Interface – XML Process Definition Language*, Workflow Management Coalition Workflow Standard. http://www.wfmc.org, 2002.
27. YAO Y., LEE H.: *Applying ISO 9001 and CMMI in Quality-Oriented Knowledge Management for Software Process Improvement*, International Journal of Electronic Business Management, Vol. 2, No. 2, 2004, pp. 140-151.

ROZDZIAŁ V

TYPOWE CZYNNIKI NIEPOWODZENIA W REALIZACJI INFORMATYCZNYCH PRZEDSIĘWZIĘĆ PROJEKTOWYCH – SPOJRZENIE *STANDISH GROUP*

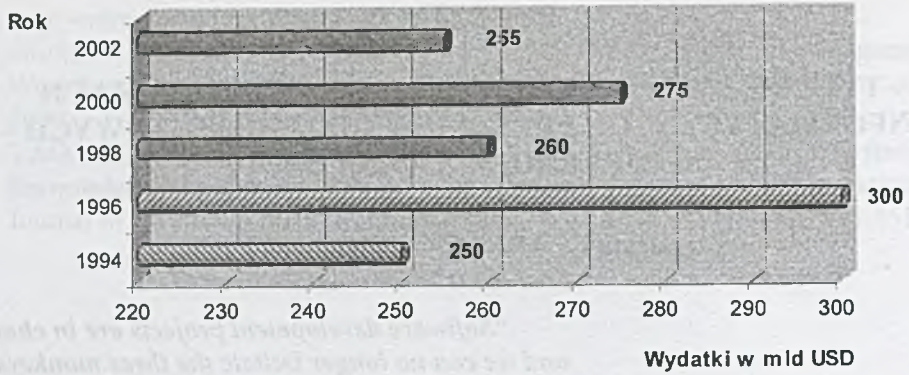
Beata CZARNACKA-CHROBOT

"Software development projects are in chaos, and we can no longer imitate the three monkeys -- hear no failures, see no failures, speak no failures"

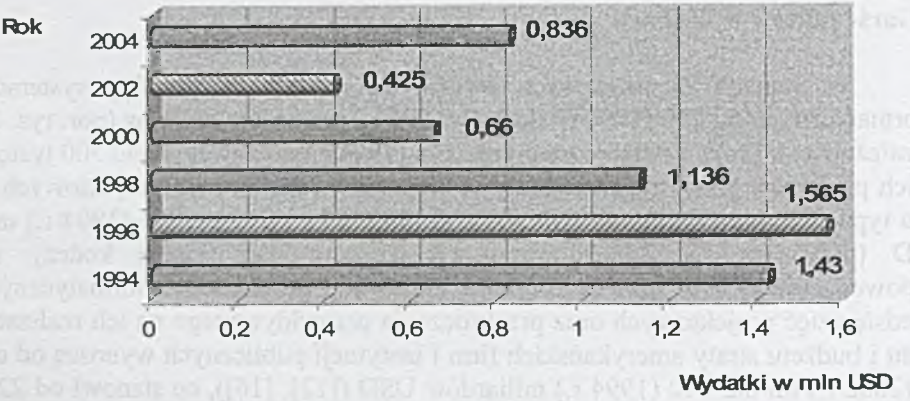
The CHAOS Report, The Standish Group International, Inc., West Yarmouth, Massachusetts 1995.

1. Garść faktów w liczbach

W Stanach Zjednoczonych wydaje się rocznie na projekty systemów informatycznych od 250 (1994 r.) do 300 (1996 r.) miliardów dolarów (por. rys. 1). W zależności od roku realizuje się od ok. 175 (1994 r.) do nawet ponad 500 tysięcy takich projektów (2001 r.). Przeciętny roczny koszt samych prac projektowych w tego typu przedsięwzięciu wynosi od ok. 0,43 (2002 r.) do ok. 1,57 (1994 r.) mln USD (por. rys. 2). Zdecydowana większość z nich ciągle kończy się niepowodzeniem, a w okresie rocznym z powodu porzucenia informatycznych przedsięwzięć projektowych oraz przekroczenia przewidywanego na ich realizację czasu i budżetu straty amerykańskich firm i instytucji publicznych wynoszą od ok. 55 (2002 r.) do ok. 140 (1994 r.) miliardów USD ([12], [16]), co stanowi od 22% do 55% kwot zainwestowanych w konstrukcję SI. Nie dość tego, wśród projektów przekraczających szacowany czas i budżet w produkcie końcowym znajdziemy obecnie tylko połowę wymagań wyspecyfikowanych przez użytkownika, a przeciętna zgodność z nimi nigdy w historii badań nie przekroczyła 70%, nawet przy uwzględnieniu przedsięwzięć zakończonych sukcesem. Generuje to po stronie odbiorców SI olbrzymie straty wynikające z utracenia potencjalnych możliwości biznesowych, a dodatkowo - niezgodność ta ma tendencję rosnącą ([12], [16]).



Rys. 1. Wydatki na informatyczne przedsięwzięcia projektowe w USA w latach 1994 - 2002 w odstępach dwuletnich
Źródło: Opracowanie własne na podstawie [16].



Rys. 2. Średni roczny koszt prac projektowych w informatycznym przedsięwzięciu projektowym w latach 1994 - 2004 w odstępach dwuletnich
Źródło: [18].

Takie mocno niepokojące dane wynikają z serii raportów dotyczących rynku IT, publikowanych systematycznie od 1995 r. przez instytucję badawczą *Standish Group International.* Zdaniem autorów powyższych badań ów *permanentny chaos* – bo tym mianem właśnie określają oni skuteczność w dostarczaniu użytkownikom wymaganych przez nich rozwiązań informatycznych - to przede wszystkim rezultat nieumiejętnego zarządzania działaniami projektowymi [13]. Jest to wniosek ogólny, wyprowadzony na podstawie analizy bardziej szczegółowych czynników decydujących o niepowodzeniu informatycznych przedsięwzięć projektowych - zaprezentowano je poniżej.

Badania *Standish Group* nie są jedynymi badaniami prowadzonymi pod tym kątem, jednak rezultaty większości z pozostałych sprowadzają się do wniosków mniej lub bardziej zbliżonych do tych, które wynikają z analiz właśnie

tej instytucji. Dotyczy to m.in. wieloletnich studiów nad przyczynami niepowodzenia projektów SI prowadzonych przez grupę naukowców z Universytetu Idaho, w tym D. Babcocka [1], J. Mereditha [10] i H. Kerznera [9], a wśród praktyków zarządzania informatycznymi przedsięwzięciami projektowymi podobne wnioski znajdziemy m.in. na listach czynników niepowodzenia tego typu przedsięwzięć zaprezentowanych w postaci „antyporadników" przez N. Karten [8] i W. Parkera [11] oraz w formie symptomów porażki przez P. Halla [4].

2. Dlaczego badania *Standish Group* właśnie?

Poza ogólną zgodnością wniosków płynących z analiz *Standish Group* (SG) z rezultatami wymienionych badań i doświadczeń, wybór tych pierwszych jako przedmiotu rozważań w niniejszym opracowaniu został podyktowany m.in. następującymi przesłankami:

- Faktem, że są one prowadzone systematycznie od 1994 r. i nadal kontynuowane, a ich zbiorcze wyniki publikuje się w odstępach z reguły dwuletnich, dzięki czemu stanowią one świadectwo zmian zachodzących w obszarze skuteczności projektowania SI i obrazują występujące w nim trendy;
- Kompleksowością badań – i to w wielu wymiarach, tj. pod względem:
  - ✓ przedmiotu badań – SG analizuje nie tylko przyczyny, ale również diagnozuje skalę i skutki niepowodzenia w realizacji projektów SI;
  - ✓ zasięgu geograficznego – w badaniach SG uwzględnia się projekty SI realizowane nie tylko w USA (obecnie prawie 60%), ale również w Europie (niemal 30%) i w innych częściach świata [18];
  - ✓ rodzaju odbiorcy – analizy obejmują projekty SI tworzone dla: bankowości, ubezpieczeń, wytwarzania, handlu, usług, opieki zdrowotnej, obronności oraz organów lokalnej, stanowej i federalnej administracji;
  - ✓ liczby przeanalizowanych projektów SI – obecnie SG może się poszczycić ponad 50 tysiącami przeanalizowanych zakończonych informatycznych przedsięwzięć projektowych (projekty porzucone również uznawane są za przedsięwzięcia zakończone, chociaż zakończone porażką) [18], a co dwa lata liczba ta wzrasta o kolejne 10 tysięcy;
  - ✓ wielkości projektów – obecnie projekty duże stanowią 45%, średnie 35%, a małe 20% przeanalizowanych przedsięwzięć [18][1];
  - ✓ rodzaju projektów - por. tablica 1.

[1] Pod względem rozmiaru *Standish Group* dzieli projekty na 3 grupy według kryterium wielkości firmy, w której dane przedsięwzięcie jest realizowane: za projekt mały uznaje przedsięwzięcie realizowane w firmie z przychodami rocznymi do 200 mln USD; za duży – w firmie z przychodami powyżej 500 mln dolarów rocznie (przedsiębiorstwa z listy 1000 największych magazynu *Fortune*); za średni zaś uznaje projekt tworzony w firmie o rocznych przychodach wynoszących od 200 do 500 mln USD.

Tablica 1. Rodzaje projektów SI uwzględnionych w badaniach *Standish Group* - stan na 2004 r.

Rodzaj przedsięwzięcia	Udział w badaniach
Projektowanie od koncepcji przy wykorzystaniu tradycyjnych metod i języków programowania	36%
Projektowanie od koncepcji przy wykorzystaniu obiektowych metodyk projektowania	19%
Niektóre komponenty projektowane, a inne nabyte	16%
Aplikacje nabyte i zmodyfikowane	13%
Komponenty nabyte i zintegrowane w aplikację	6%
Aplikacje nabyte i rozległe zmodyfikowane	6%
Aplikacje nabyte i niezmodyfikowane	4%

Źródło: [18].

- Spójnością wyników badań w wymiarze przedmiotowym (przyczyny są powiązane logicznie ze skutkami);
- Jasno zdefiniowanymi kryteriami (nie)powodzenia projektu SI;
- Faktem, iż dane źródłowe do analiz pochodzą od odbiorców rozwiązań informatycznych - nigdy zaś od ich dostawców;
- Finansowaniem badań ze środków własnych;
- Docenieniem ich teoretycznej wartości - wnioski, które z nich płyną zostały niejednokrotnie potwierdzone przez studia nadzorowane przez takie autorytety w omawianej dziedzinie, jak np. Capers Jones z *Software Productivity Research*, Lary Putnam czy Paul Strassmann;
- Docenieniem ich praktycznej wartości – np. na poziomie administracji rządowej rezultaty badań SG wpłynęły na [18]:
  - ✓ anulowanie przez *General Services Administration* (GSA), tj. amerykański naczelny urząd ds. zamówień na potrzeby administracji, kontraktów na informatyczne przedsięwzięcia projektowe o wartości 14 miliardów USD;
  - ✓ utworzenie w Wielkiej Brytanii *Office of Government Commerce* (OGC), czyli Rządowego Biura ds. Handlu, zajmującego się m.in. projektami informatycznymi realizowanymi na rzecz administracji tego kraju;
  - ✓ zasadniczą treść ogłoszonego w 1996 r. w Stanach Zjednoczonych aktu prawnego, który dotyczył reformy zarządzania technologią informatyczną w administracji rządowej (*The Information Technology Management Reform Act – ITMRA*, popularnie zwany *Clinger-Cohen Bill*) - reguluje on m.in. konieczność oceny efektów inwestycji informatycznych podejmowanych przez administrację tego kraju.

Wszystkie te fakty powodują, że *Standish Group* to instytucja badawcza uznawana w branży IT za dostawcę rzeczywiście wiarygodnych danych statystycznych na temat skuteczności realizacji informatycznych przedsięwzięć projektowych. Toteż, mimo znanego powiedzenia Hugo Steinhausa, że *„są kłamstwa, wielkie kłamstwa i... statystyka"*, jak również mimo ich niewygodnych dla środowiska IT rezultatów, przez co, wbrew ich docenieniu przez znaczącą jego

większość, zwłaszcza w USA, nadal można się spotkać z ich sceptycznym odbiorem - a może właśnie przede wszystkim dlatego - nie można ich ignorować.

Pierwsza wersja raportu *Standish Group*, obejmująca rezultaty uzyskane z analizy danych za rok 1994, ukazała się w 1995 r. pod jednoznacznym tytułem *„The CHAOS Report"* [12]. Wnioski z badań, jak również spostrzeżenie, iż w roku opublikowania raportu niepowodzeniem zakończyło się jeszcze więcej projektów SI niż wynikało z analiz, spowodowały, iż postanowiono je kontynuować. Tym razem celem studiów miało być przede wszystkim znalezienie zasadniczych przyczyn owego zjawiska. Zbiorcze raporty, obejmujące rezultaty badań prowadzonych od 1994 r. do roku poprzedzającego ich wydanie, ukazywały się w latach 2001 i 2003, nosząc odpowiednio tytuły: *„The CHAOS Chronicles II"* [15] oraz *„The CHAOS Chronicles III"* [16]. Wydanie czwartej wersji *„Kronik CHAOSU"* zapowiadano na 2005 r., ale jak dotąd udostępniono tylko częściowe dane za rok 2004 [18]. Jak widać z powyższego, za każdym razem w tytule opracowania pojawia się słowo *„CHAOS"*, które zdaniem autorów badań najlepiej odzwierciedla stan rzeczy mający miejsce w odniesieniu do skuteczności realizacji projektów SI. I w istocie - analizując dane z omawianych raportów trudno temu zaprzeczyć, o czym przekonują rozważania zawarte w dalszych punktach niniejszego opracowania.

Standish Group prowadzi swoje badania przede wszystkim z punktu widzenia tzw. współczynnika sukcesu informatycznych przedsięwzięć projektowych, w tym stopnia zgodności produktów końcowych z wymaganiami wyspecyfikowanymi przez użytkowników ([12], [15], [16], [18]). **Współczynnik sukcesu** to udział w ogólnej liczbie projektów SI zakończonych w danym roku takich przedsięwzięć, które zakończyły się powodzeniem.

3. Sukces w realizacji informatycznego przedsięwzięcia projektowego według *Standish Group*

Powyżej nadmieniono, iż jedną z zasadniczych zalet badań SG stanowią jasno zdefiniowane kryteria, zgodnie z którymi zakończony projekt SI zalicza się do przedsięwzięć udanych. Według SG projekt SI zakończony powodzeniem musi spełniać następujące trzy warunki konieczne:

- został zakończony w ramach zaplanowanego czasu,
- został zakończony w granicach oszacowanego budżetu,
- jego rezultatem jest produkt końcowy (niemal) całkowicie zgodny z wymaganiami wyspecyfikowanymi przez użytkownika co do funkcji i cech.

Przeciwwagę dla udanych projektów SI stanowią, co oczywiste, przedsięwzięcia zakończone niepowodzeniem. Omawiane badania traktują o dwóch rodzajach niepowodzeń informatycznych przedsięwzięć projektowych, a mianowicie o:

- **Niepowodzeniu w aspekcie ilościowym**, czyli udziale w ogólnej liczbie wszystkich zakończonych w danym roku projektów SI tych przedsięwzięć, które zakończyły się:

- **niepowodzeniem całkowitym**, tj. zostały porzucone;

lub:

- **niepowodzeniem częściowym**, tj.:
  - ✓ przekroczyły estymowany czas trwania działań projektowych;

  i/lub
  - ✓ przekroczyły budżet planowany na dane przedsięwzięcie.

❑ **Niepowodzeniu w aspekcie jakościowym**, a zatem pod względem niskiego stopnia zgodności produktu końcowego co do funkcji i cech ze specyfikacją wymagań użytkownika. Tego rodzaju niepowodzenie dotyczy przedsięwzięć zakończonych ilościowym niepowodzeniem częściowym, jako że z definicji współczynnika sukcesu wynika, że udany projekt to taki, który jest (niemal) całkowicie zgodny z ową specyfikacją, zaś projektów porzuconych nie da się oczywiście pod tym kątem przeanalizować.

Zgodnie zatem z interpretacją przyjętą przez *Standish Group* do grupy projektów nieudanych zalicza się nie tylko te, które zakończyły się niepowodzeniem całkowitym, ale również te, które zakończyły się niepowodzeniem częściowym, w tym niepowodzeniem w aspekcie jakościowym, jako że generują one straty spowodowane:

- przekroczeniem planowanego czasu i budżetu (możliwe do oszacowania), oraz
- brakiem zgodności z wymaganiami wyspecyfikowanymi przez użytkownika (trudne lub nawet niemożliwe do oszacowania).

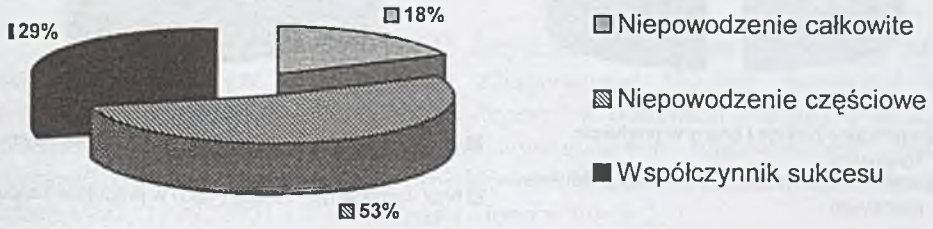
Zdefiniowane przez SG kryteria nie uwzględniają praktycznie żadnej tolerancji dla czasu i kosztów realizacji działań projektowych, jak również dla dostarczonych w produkcie końcowym funkcji i cech: rzeczywisty czas i koszty nie mogą przekraczać wartości planowanych, zaś funkcje i cechy dostarczonego SI muszą być całkowicie zgodne ze specyfikacją wymagań użytkownika. Dlatego można spotkać się z krytyką tych kryteriów – pojawiają się mianowicie opinie, że są one zbyt ostre. Nikt jednak, przynajmniej jak dotąd, nie podał kryteriów, które byłyby wobec nich konkurencyjne.

4. Aktualna skala niepowodzeń w realizacji informatycznych przedsięwzięć projektowych

Według najnowszych danych płynących z badań *Standish Group*, dla informatycznych przedsięwzięć projektowych średnio, tj. niezależnie od ich wielkości, w 2004 r. [18]:

- współczynnik sukcesu wynosił 29%;
- projekty zakończone niepowodzeniem częściowym stanowiły 53% przedsięwzięć;
- porzucono 18% projektów SI.

Udział informatycznych przedsięwzięć projektowych, które zakończyły się niepowodzeniem w aspekcie ilościowym, tj. niepowodzeniem pełnym (całkowitym) lub częściowym, wynosił zatem 71% (por. rys. 3).



Rys. 3. Średnia skala niepowodzenia ilościowego projektów SI w 2004 r. Źródło: [18].

W tymże roku wśród projektów zakończonych niepowodzeniem częściowym zauważono ponadto [18]:

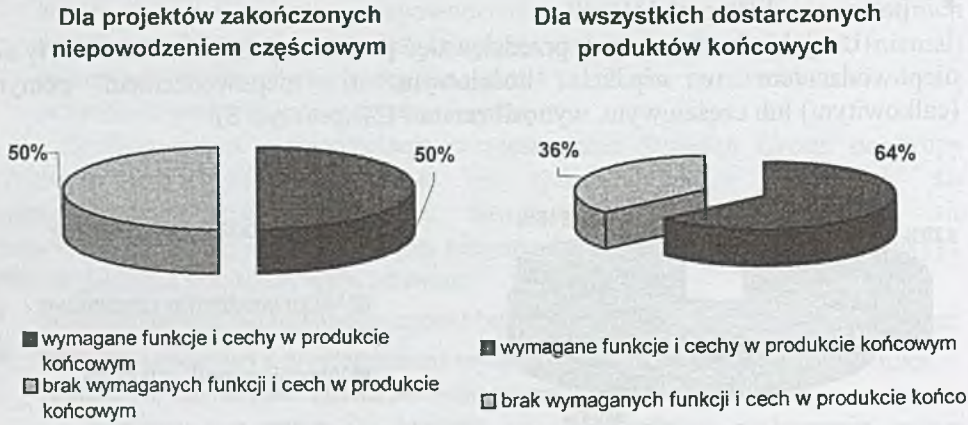
- przekroczenie czasu realizacji projektu SI przeciętnie o 84% w stosunku do czasu planowanego;
- przekroczenie kosztów realizacji projektu SI przeciętnie o 56% w stosunku do kosztów planowanych.

Zakładając, iż poziom wydatków na informatyczne przedsięwzięcia projektowe w Stanach Zjednoczonych w 2004 r. nie był mniejszy od ich poziomu z 2002 r. (255 mld USD), można oszacować skalę strat amerykańskich firm i instytucji rządowych spowodowanych niepowodzeniem projektów SI w aspekcie ilościowym w 2004 r. – w takim przypadku należy je ostrożnie ocenić na minimum 63 miliardy dolarów (niemal 25% wydatków)[2], w tym w wyniku:

[2] Autorka nie dysponuje danymi *Standish Group* dotyczącymi poziomu wydatków na projekty IT w USA w 2004 r., a oceny szacunkowe zostały przez nią wyprowadzone na podstawie danych o stratach za 2002 r. [16]. Szacunek ten ma jednak charakter ostrożny, gdyż nie uwzględniono w nim wzrostu średniego przekroczenia czasu i kosztów, który

- niepowodzenia całkowitego, tj. porzucenia projektów (18% przedsięwzięć zakończonych) straty mogły wynieść minimum 45,6 mld USD,
- niepowowodzenia częściowego (53% projektów zakończonych) straty mogły wynieść minimum 17,4 mld.

Jeżeli zaś ów poziom wydatków był w 2004 r. znacząco większy w zestawieniu z rokiem 2002, co po zakończeniu kryzysu w branży IT jest bardzo prawdopodobne, to straty te również były sporo większe.



Rys. 4. Średnia skala niepowodzenia jakościowego projektów SI w 2004 r.[3]
Źródło: Opracowanie własne na podstawie: [16], [18].

Wnioskując na podstawie najnowszych [18] oraz wynikających z poprzedniego raportu danych [16], w grupie informatycznych przedsięwzięć projektowych przekraczających estymowany czas i koszty realizacji (obecnie 53% zakończonych projektów SI) niepowodzenie w aspekcie jakościowym w 2004 r. można prawdopodobnie ostrożnie ocenić na ok. 50%, co oznacza, że średnia zgodność produktu końcowego z wymaganiami wyspecyfikowanymi przez użytkownika co do funkcji i cech w przypadku przedsięwzięć zakończonych ilościowym niepowodzeniem częściowym wahała się w granicach jedynie połowy (por. rys. 4)[4]. Przy uwzględnieniu wszystkich dostarczonych produktów końcowych, a więc również tych zakończonych sukcesem, znajdziemy w nich

zanotowano w 2004 r. w stosunku do roku 2002 (dotyczy ilościowego niepowodzenia częściowego).

[3] Wykres przedstawiający skalę niepowodzenia jakościowego dla projektów zakończonych niepowodzeniem częściowym należy potraktować jako hipotetyczny.

[4] W [18] nie ma danych dotyczących omawianej grupy projektów, dlatego ocena ta ma charakter hipotetyczny, chociaż ostrożny - w 2004 r. nieco wzrósł bowiem odsetek projektów zakończonych niepowodzeniem cześciowym. Ocena ta została wyprowadzona w oparciu o dane o tej grupie projektów z 2002 r. [16] oraz w oparciu o dane o ogólnej zgodności z 2004 r. [18].

przeciętnie tylko 64% funkcji i cech ze specyfikacji wymagań (por. rys. 4). Jak już wcześniej nadmieniono, niepowodzenie w ujęciu jakościowym generuje olbrzymie straty, które jednak bardzo trudno oszacować, gdyż wynikają one przede wszystkim z utraconych możliwości biznesowych przez użytkowników SI.

Jak wynika z powyższych statystyk, obecnie na 100 realizowanych projektów SI 71 kończy się niepowodzeniem, w tym 18 całkowitym upadkiem. Na pozostałe 53 przedsięwzięcia trzeba faktycznie wydać średnio 156% szacowanych kosztów, a na ich rzeczywiste zakończenie można liczyć przeciętnie po upływie 184% czasu planowanego, w efekcie uzyskując w produkcie końcowym jedynie średnio ok. 50% wymaganych funkcji i cech. Niemal 25% środków przeznaczonych na informatyczne przedsięwzięcia projektowe można uznać za zmarnotrawione.

5. Typowe czynniki niepowodzenia informatycznych przedsięwzięć projektowych niegdyś i obecnie

Ponieważ rezultaty badań SG dotyczące skali chaosu zaskoczyły nawet samych ich autorów, postanowiono zdiagnozować również przyczyny tego zjawiska. Począwszy od *„The CHAOS Report"* w odstępach z reguły dwuletnich prezentowano zestawienia czynników wywołujących niepowodzenie tysięcy przeanalizowanych w tym okresie projektów SI. Wnioski z analiz opublikowane w latach 1995, 1999, 2001 i 2003 przedstawiono w tabeli 2.

Aktualne czynniki niepowodzenia projektów SI, wynikające z przeanalizowania przez *Standish Group* kolejnej kilkutysięcznej grupy przedsięwzięć zakończonych, zaprezentowano natomiast w tabeli 3. Zawarto w niej również zestawienie dzisiejszych przyczyn upadków projektów SI z rezultatami badań poprzednich oraz wynikami opublikowanymi dziesięć lat wcześniej. Pierwsze cztery czynniki z tej tabeli to obecne czynniki krytyczne (oznaczone jako (k)), zaś czynniki wymienione jako 4., 6., 7. i częściowo 10. to czynniki pojawiające się w takiej formie w najnowszym badaniu po raz pierwszy.

Porównanie przyczyn chaosu w realizacji informatycznych przedsięwzięć projektowych z 2004 r. z czynnikami niepowodzenia z 1994 r. pozwala sformułować następujące wnioski (por. tablica 3):

1) Przez 10 lat nie zmieniło się znaczenie dwóch pierwszych krytycznych czynników upadków projektów SI, mianowicie:
   - braku faktycznego zaangażowania użytkownika w działania projektowe;
   - niedostatecznego wsparcia projektu ze strony zarządu organizacji (sponsora).
2) Do grupy czynników krytycznych w ciągu dekady awansowały:
   - niejasne cele biznesowe przedsięwzięcia (obecnie na miejscu 3.) – zastąpiły one czynnik z 1994 r.: niejasna wizja i cele projektu, który ówcześnie był dopiero na miejscu 9.;
   - niezoptymalizowany zakres projektu SI i wymagania wobec niego (obecnie na pozycji 4.) – pojawił się on na miejsce czynnika: zbyt długie

etapy projektowania, który w 1994 r. znajdował się na miejscu 6., w wyniku postulowanej zmiany podejścia do działań projektowych z kaskadowego na przyrostowe;

3) W okresie po 1994 r. na omawianej liście pojawiły się jak dotąd jedynie dwie zupełnie nowe przyczyny niepowodzenia projektów SI:
   - niedoświadczony kierownik projektu (obecnie na pozycji 5.);
   - niewykorzystywanie standardowej infrastruktury operacyjnej i standardowych narzędzi wspomagających projektowanie (obecnie na pozycji 10.).
4) W analizowanym czasie uszczegółowiono czynnik krytyczny z miejsca 4. w 1994 r., tj. niewłaściwe planowanie, który rozbito na dwa nowe:
   - niewłaściwe zarządzania finansami przedsięwzięcia (obecnie na pozycji 7., z tendencją do wzrostu znaczenia);
   - niewykorzystywanie formalnej metodyki projektowania (obecnie na pozycji 9., z tendencją do spadku znaczenia).
5) Po latach powrócono do następujących dwóch czynników niepowodzenia, jednocześnie rozszerzając ich znaczenie:
   - nieiteracyjny i nieelastyczny proces projektowania, w tym proces ustalania wymagań wobec SI (obecnie na miejscu 6.) – w badaniach sprzed dekady czynnik ów, w formie: niejasne formułowanie wymagań, był czynnikim krytycznym (na pozycji 3.), ale w badaniach z lat pośrednich (1998, 2000 – por. tablica 2) większe znaczenie przywiązywano do celu, tj. faktu istnienia ustalonych podstawowych wymagań, niźli do przebiegu procesu ich formułowania, podczas gdy mówiąc o czynnikach niepowodzenia należy raczej wskazywać właśnie na działania, których należy unikać;
   - niedostatecznie wykwalifikowane i wprawne zasoby ludzkie (obecnie na miejscu 8.) – na liście przyczyn niepowodzenia projektów SI z 1994 r. czynnik ten ma postać: niekompetentny zespół projektowy (zajmuje pozycję 7.), jednak w badaniach za rok 2000 w ogóle się on w takim zestawieniu nie pojawia, zaś w 2002 r. jego znaczenie jest mniejsze (pozycja 10.).
6) Według najnowszych analiz w ciągu dekady na znaczeniu mocno straciły, tj. nie ma ich na liście dziesięciu obecnie najważniejszych czynników, takie przyczyny upadków projektów SI, jak (por. tablica 2):
   - nierealne oczekiwania wobec SI;
   - niejasno określona własność projektu (odpowiedzialność za projekt);
   - brak koncentracji zespołu projektowego na realizowanym projekcie.

Tablica 2. Czynniki niepowodzenia projektów SI w latach: 1994, 1998, 2000 i 2002

Rok	Lp.	Czynnik niepowodzenia	Znaczenie (miejsce czynnika)			
			1994	1998	2000	2002
1994	1	**Brak zaangażowania użytkownika**	**1. (k)**	**1. (k)**	**2. (k)**	**1. (k)**
	2	**Brak wsparcia ze strony zarządu (sponsora)**	**2. (k)**	**2. (k)**	**1. (k)**	**2. (k)**
	3	**Niejasne formułowanie wymagań**	**3. (k)**	przeformułowany na 13-sty		przeformuł. na 18-sty
	4	**Niewłaściwe planowanie**	**4. (k)**	8.	brak w takiej formie - zamieniony na 16-sty i 17-sty	
	5	Nierealne oczekiwania	5.	brak		
	6	Zbyt długie etapy projektowania	6.	5.	przeformułowany na 14-sty	
	7	Niekompetentny zespół projektowy	7.	7.	brak	10.
	8	Niejasno określona własność projektu (odpowiedzialność)	8.	9.*	brak	
	9	Niejasna wizja i cele projektu	9.	przeformułowany na 11-sty		
	10	Brak koncentracji zespołu na projekcie	10.	brak		
1998	11	**Niejasne cele biznesowe projektu**	przeformuł. z 9-ego	**3. (k)**	**4. (k)**	**4. (k)**
	12	***Niedoświadczony kierownik projektu***	brak	4. (n)	**3. (k)**	**3. (k)**
	13	Nieustalone podstawowe wymagania	przeformuł. z 3-ego	6.	7.	przeformuł. na 18-sty
2000	14	Niezminimalizowany zakres projektu	przeformułowany z 6-ego		5.	5.
	15	*Niewykorzystywanie standardowej infrastruktury oprogramowania*	brak		6. (n)	7.
	16	Niewykorzystywanie formalnej metodyki	brak w takiej formie - oba potraktowane jako uszczegółowienie 4-tego		8.	8.
	17	Brak wiarygodnych oszacowań			9.*	9.
2002	18	Nieelastyczny proces ustalania wymagań	przeformułowany z 3-ego		przeformuł. z 13-ego	6.

(k) Czynnik krytyczny w danym roku (dodatkowo zaznaczony czcionką pogrubioną);
(n) Czynnik, który nie był w żadnej formie uwzględniany w poprzednich badaniach (wskazany dodatkowo przez kursywę).
* Na miejscu 10. w raporcie znajduje się pozycja „Inne".

Źródło: Opracowanie własne na podstawie: [12], [14], [15] i [16].

Tablica 3. Czynniki niepowodzenia projektów SI w 2004 r. i ich porównanie z czynnikami z roku 1994 i 2002

Lp.	Czynnik niepowodzenia (2004 r.)	Znaczenie (miejsce) w 1994 r.	Znaczenie (miejsce) w 2002 r.
1 (k)	**Brak zaangażowania użytkownika**	**1. (k)**	**1. (k)**
2 (k)	**Brak wsparcia ze strony zarządu organizacji**	**2. (k)**	**2. (k)**
3 (k)	**Niejasne cele biznesowe przedsięwzięcia**	9. w formie: Niejasna wizja i cele projektu	**4. (k)**
4 (k)	***Niezoptymalizowany zakres projektu i wymagania wobec niego***	6. w formie: Zbyt długie etapy projektowania	5. w formie: Niezminimalizowany zakres projektu
5	Niedoświadczony kierownik projektu	brak	**3. (k)**
6	Nieiteracyjny i nieelastyczny proces projektowania	częściowo **3. (k)** w formie: Niejasne formułowanie wymagań	częściowo 6. w formie: Nieelastyczny proces ustalania wymagań
7	Niewłaściwe zarządzanie finansami przedsięwzięcia	brak w takiej formie - szerzej jako **4. (k)** w formie: Niewłaściwe planowanie	częściowo 9. w formie: Brak wiarygodnych oszacowań
8	Niedostatecznie wykwalifikowane i wprawne zasoby ludzkie	7. w formie: Niekompetentny zespół projektowy	10. w formie: Niekompetentny zespół projektowy
9	Niewykorzystywanie formalnej metodyki	brak w takiej formie - szerzej jako **4. (k)** w formie: Niewłaściwe planowanie	8.
10	Niewykorzystywanie standardowej infrastruktury operacyjnej i *standardowych narzędzi wspomagających projektowanie*	brak	cześciowo 7. w formie: Niewykorzystywanie standardowej infrastruktury oprogramowania

Źródło: Opracowanie własne na podstawie: [12], [16] i [18].

Reasumując zmiany, które wystąpiły na liście przyczyn upadków informatycznych przedsięwzięć projektowych w okresie dekady, należy podkreślić, że wzrosło znaczenie następujących czynników niepowodzenia projektów SI[5]:

[5] Wyrazem wzrostu znaczenia jest również uszczegółowienie czynników, które miały postać bardziej ogólną.

- niejasne cele biznesowe przedsięwzięcia,
- niezoptymalizowany zakres projektu i wymagania wobec niego (zmienił formę),
- niedoświadczony kierownik projektu (nie istniał na liście z 1994 r.),
- niewłaściwe zarządzanie finansami przedsięwzięcia (miał postać bardziej ogólną),
- niewykorzystywanie formalnej metodyki projektowania (miał postać bardziej ogólną),
- niewykorzystywanie standardowej infrastruktury operacyjnej i standardowych narzędzi wspomagających projektowanie (nie istniał na liście z 1994 r.).

Z istniejących w obecnym zestawieniu czynników nieco zmalało znaczenie wykwalifikowanych i wprawnych kadr. Jednak najważniejszy wniosek płynący z porównania aktualnej listy przyczyn niepowodzenia projektów SI z listą sprzed dziesięciu lat, niewątpliwie warty ponownego zaakcentowania, brzmi następująco: **w okresie dekady nie zmieniło się krytyczne znaczenie dwóch podstawowych przyczyn upadków projektów SI, do których należy:**

1) **Brak faktycznego zaangażowania użytkownika w działania projektowe**, co często skutkuje:
   - niezgodnością produktu końcowego z wymaganiami użytkownika co do funkcji i cech,
   - porzuceniem już zrealizowanego produktu,
   - przerwaniem przedsięwzięcia,
   - przekroczeniem zaplanowanych dla projektu zasobów.
2) **Niedostateczne wsparcie projektu ze strony jego sponsora**, w efekcie czego najczęściej występuje:
   - przekroczenie szacowanego czasu trwania przedsięwzięcia,
   - przekroczenie planowanych kosztów projektu,
   - przerwanie projektu,
   - niezgodność produktu końcowego ze specyfikacją wymagań,
   - porzucenie już zrealizowanego produktu.

6. Zmiana skali i skutków niepowodzeń informatycznych przedsięwzięć projektowych w dekadzie chaosu

W tabeli 4 oraz na rysunku 5 przedstawiono średnią skalę niepowodzenia projektów SI w aspekcie ilościowym odnotowaną w badaniach SG w latach 1994 – 2004. Zaprezentowane tam dane prowadzą do następujących wniosków:

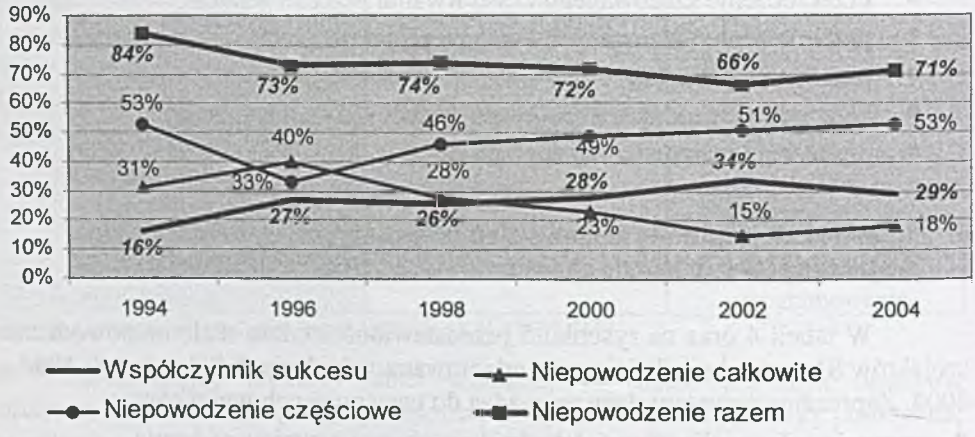
- w czasie całej analizowanej dekady skala chaosu z punktu widzenia niepowodzenia ilościowego uległa zmniejszeniu w sumie o 13 punktów procentowych (z 84 do 71%);

- powodem spadku tej skali w badanym okresie było zmniejszenie się niepowodzenia całkowitego – właśnie o 13 punktów procentowych (z 31 do 18%);
- w analizowanej dekadzie ilościowe niepowodzenie częściowe, po dużym spadku w 1996 r. i dalszym systematycznym wzroście, pozostało w efekcie na niezmienionym poziomie (53%);
- najniższą skalę chaosu z punktu widzenia ilościowego zanotowano w 2002 r. (66%), kiedy to zmniejszyła się ona w stosunku do jego skali początkowej o 18 punktów procentowych;
- **w 2004 r. nastąpił ponowny wzrost skali chaosu w aspekcie ilościowym** – w stosunku do poprzednich badań o 5 punktów procentowych, w tym z powodu zwiększenia się:
  - ✓ o 2 punkty procentowe niepowodzenia częściowego,
  - ✓ o 3 punkty procentowe niepowodzenia całkowitego.

Tablica 4. Średnia skala niepowodzenia ilościowego projektów SI w latach 1994 – 2004 w odstępach dwuletnich

Dane za rok	Współczynnik sukcesu (średnio)	Niepowodzenie częściowe (średnio)	Niepowodzenie całkowite (średnio)	Niepowodzenie razem (średnio)
1994	16%	53%	31%	**84%**
1996	27%	33%	40%	**73%**
1998	26%	46%	28%	**74%**
2000	28%	49%	23%	**72%**
2002	34%	51%	15%	**66%**
2004	**29%**	**53%**	**18%**	***71%***

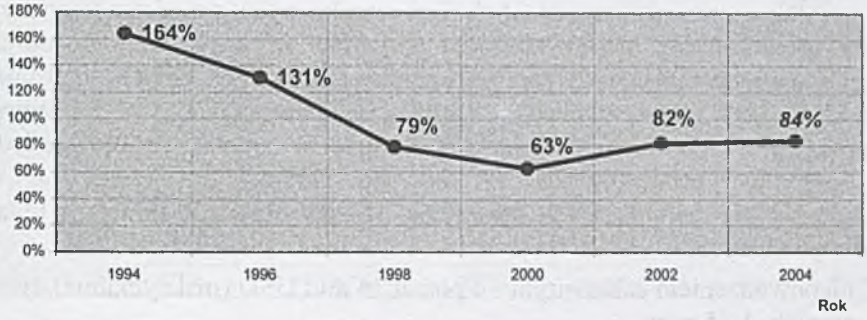
Źródło: Opracowanie własne na podstawie [18].



Rys. 5. Średnia skala niepowodzenia ilościowego projektów SI w latach 1994 – 2004 w odstępach dwuletnich
Źródło: Opracowanie własne na podstawie [18].

Dla niepowodzenia częściowego w aspekcie ilościowym interesujące są również dane dotyczące średniej skali przekroczenia planowanego czasu i szacowanych kosztów realizacji projektu SI – zaprezentowano je na rys. 6 i 7. Wynika z nich, że:

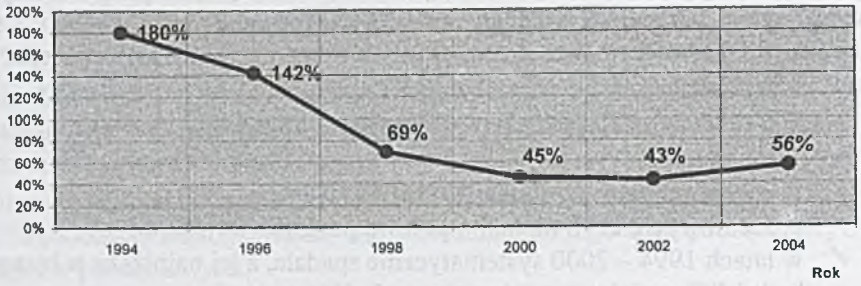
- Średnia skala przekroczenia estymowanego czasu trwania działań projektowych (por. rys. 6):
  - ✓ w całym analizowanym okresie zmniejszyła się o 80 punktów procentowych, czyli niemal o połowę poziomu wyjściowego;
  - ✓ w latach 1994 – 2000 systematycznie spadała, a jej najniższy w historii badań SG przeciętny poziom wynosił 63%;
  - ✓ po roku 2000 zanotowano jej wzrost – do obecnego przeciętnego poziomu 84%.



Rys. 6. Średnia skala przekroczenia planowanego czasu realizacji projektów SI w latach 1994 – 2004 w odstępach dwuletnich
Źródło: [18][6].

- Średnia skala przekroczenia szacowanych kosztów realizacji projektów SI (por. rys. 7):
  - ✓ w całym analizowanym okresie zmniejszyła się aż o 124 punkty procentowe, a zatem ponad trzykrotnie;
  - ✓ w latach 1994 – 2002 systematycznie spadała, a jej najniższy w historii badań SG przeciętny poziom to 43%;
  - ✓ w 2004 r. ponownie wzrosła – do przeciętnego poziomu 56%.

[6] Przy czym w [12] i [15] podana jest inna wielkość dla 1994 r.: zamiast 164% widnieje tam 222%.



Rys. 7. Średnia skala przekroczenia planowanych kosztów realizacji projektów SI w latach 1994 – 2004 w odstępach dwuletnich
Źródło: [18][7].

Jak zaprezentowano w tabeli 5, w wyniku wyżej przedstawionych zjawisk straty spowodowane niepowodzeniem projektów SI w aspekcie ilościowym w okresie analizowanej dekady zmalały przypuszczalnie (por. pkt 4) o maksimum 77 mld USD, czyli ponad dwukrotnie (o 55% poziomu strat z 1994 r.). Wprawdzie współczynnik niepowodzenia projektów SI spadł w omawianym okresie jedynie o 13 punktów procentowych (z 84 do 71%), to jednak z dużym prawdopodobieństwem można stwierdzić, iż zdecydowanie mniejsza była skala strat wywołanych:

- niepowodzeniem całkowitym - o ponad 34 mld USD (maksymalnie), tj. straty zmalały 1,75 raza;
- niepowodzeniem częściowym – o prawie 43 mld USD (maksymalnie), tj. straty zmalały niemal 3,5 raza.

W analizowanej dekadzie rozbieżność pomiędzy niewielką zmianą współczynnika niepowodzenia informatycznych przedsięwzięć projektowych a względnie dużym zmniejszeniem skali strat można wytłumaczyć kombinacją następujących zaobserwowanych zjawisk:

- spadkiem niepowodzenia całkowitego, a zatem tego, które generuje największe straty, o 13 punktów procentowych, czyli właśnie o tyle, o ile zmniejszył się współczynnik niepowodzenia projektów SI (por. tablica 4);
- w odniesieniu do niepowodzenia częściowego, które pozostaje na tym samym poziomie (por. rys. 6 i 7):
  - ✓ spadkiem średniego przekroczenia planowanego czasu o 80 punktów procentowych (por. rys. 6);
  - ✓ spadkiem średniego przekroczenia szacowanych kosztów o prawie 125 punktów procentowych (por. rys. 7);
  - ✓ zdecydowanym wzrostem liczby realizowanych przedsięwzięć (w latach 1994 – 2000 już aż o 60%), co przy wydatkach na niemal identycznym

[7] Przy czym w [12] i [15] podana jest inna wielkość dla 1994 r.: zamiast 180% widnieje tam 189%.

poziomie (por. tablica 5) oznacza, że przeciętnie stały się one tańsze i mniejsze.

Tablica 5. Skala strat w wyniku niepowodzenia ilościowego projektów SI w roku 1994, 2002 i 2004 w USA

Źródło strat		1994	2002	2004[8]	Zmiana 2004/1994[9]
WYDATKI (w USD)		250 mld	255 mld	min. 255 mld	min. + 5 mld
niepowodzenie częściowe	w USD	60 mld	17 mld	min. 17,4 mld	maks. - 42,6 mld
	w % wydatków	24%	7%	min. 7%	maks. - 17 p. %*
	w % strat	100%	28%	min. 29%	maks. - 71%
niepowodzenie całkowite	w USD	80 mld	38 mld	min. 45,6 mld	maks. - 34,4 mld
	w % wydatków	32%	15%	min. 18%	maks. - 14 p. %
	w % strat	100%	47,5%	min. 57%	maks. - 43%
RAZEM	**w USD**	**140 mld**	**55 mld**	**min. 63 mld**	**maks. - 77 mld** (ponad 2x)
	w % wydatków	**56%**	**22%**	**min. 25%**	**maks. - 31 p. %**
	w % strat	**100%**	**39%**	**min. 45%**	**maks. - 55%**

* p.% - punkty procentowe.

Źródło: Opracowanie własne na podstawie: [12], [16] i [18].

Według opinii analityków ze *Standish Group* **główną przyczyną zanotowanego w okresie dekady spadku skali chaosu w realizacji projektów SI z punktu widzenia niepowodzenia ilościowego oraz zmniejszenia strat z tego tytułu była minimalizacja zakresu realizowanych przedsięwzięć, stanowiąca konsekwencję częstszego stosowania wraz z upływem czasu iteracyjnego podejścia do projektowania** - w przeciwieństwie do podejścia kaskadowego, wymagającego zdefiniowania wszystkich wymagań na samym początku cyklu projektowego, co – jak pokazuje praktyka – nie jest możliwe [17]. Ów pogląd znajduje odzwierciedlenie w aktualnym zestawieniu czynników niepowodzenia informatycznych przedsięwzięć projektowych (por. tablica 3).

[8] Przypomnijmy, iż dane za rok 2004 pochodzą z szacunków dokonanych przez autorkę niniejszego opracowania na podstawie danych za rok 2002 [16] i przy założeniu braku zasadniczej zmiany w poziomie wydatków w roku 2004 w zestawieniu z rokiem 2002 (ok. 255 mld USD). Nie uwzględniono w nich również wzrostu średniego przekroczenia czasu i kosztów, który zanotowano w 2004 r. w stosunku do roku 2002 (dotyczy ilościowego niepowodzenia częściowego).

[9] W związku z faktem, iż dane za rok 2004 pochodzą z szacunków, również te zmiany mają charakter szacunkowy.

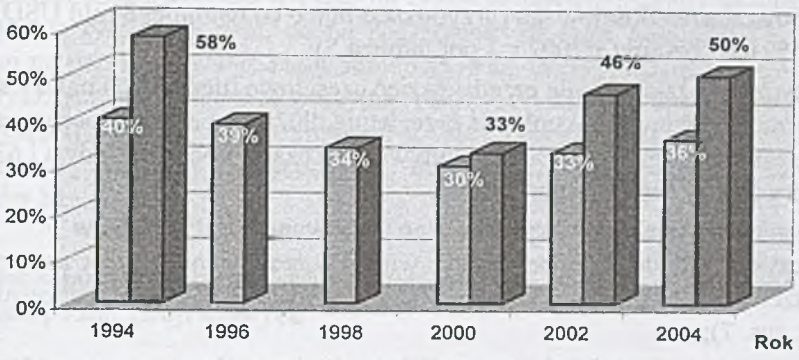
Zaobserwowane w analizowanym okresie pozytywne zmiany mogą świadczyć o tym, że po upływie dekady osoby zarządzające informatycznymi przedsięwzięciami projektowymi rzeczywiście doceniły znaczenie czynników niepowodzenia projektów SI i próbowały w praktyce im zapobiegać. Potwierdzenie tego wniosku znajdziemy w komentarzu Jima Johnsona, prezesa *Standish Group*, do wyników raportu z 2003 r.: *„Peaple have become much more savvy in project management. When we first started the research, project management was a sort of black art. People have spent time trying to get it right (...).”* [17].

Ciągle pozostaje wszak pytanie, czy to już wszystko, co można zrobić, czyli czy faktycznie wykorzystuje się wszystkie możliwości zapobiegania upadkom projektów SI? **Spowodowane wszak niepowodzeniem przedsięwzięć w aspekcie ilościowym *„koszty (...) błędów i przekroczeń to tylko szczyt przysłowiowej góry lodowej. Koszty utraconych [przez użytkownika – przyp. aut.] możliwości nie są mierzalne, ale swobodnie mogłyby wynieść biliony dolarów (...). [Na przykład – przyp. aut.] porażka w tworzeniu niezawodnego oprogramowania do obsługi bagaży na nowym lotnisku w Denver kosztuje miasto 1,1 miliona dolarów dziennie.”*** [12]. Te straty zaś wynikają z niepowodzenia projektów SI w aspekcie jakościowym, czyli z niewystarczającego poziomu zgodności dostarczonego produktu końcowego z wymaganiami wyspecyfikowanymi przez użytkownika pod względem funkcji i cech.

Dane dotyczące kształtowania się niepowodzenia jakościowego informatycznych przedsięwzięć projektowych w latach 1994 – 2004 przedstawiono na rys. 8. Z zaprezentowanego tam wykresu wynika, iż:

- W odniesieniu do wszystkich projektów SI, które zakończyły się stworzeniem produktu końcowego:
  - ✓ w okresie dekady przeciętne niepowodzenie w aspekcie jakościowym spadło jedynie o 4 punkty procentowe – z 40% w 1994 r. do 36% w 2004 r.;
  - ✓ w latach 1994 – 2000 średnia skala niepowodzenia w aspekcie jakościowym systematycznie malała - w sumie spadła o 10 punktów procentowych;
  - ✓ nigdy w historii badań SG przeciętna zgodność produktu końcowego z wymaganiami użytkownika pod względem cech i funkcji nie przekroczyła poziomu 70% - była ona największa w 2000 r.;
  - ✓ **po 2000 r. średnie niepowodzenie jakościowe ma wyraźną tendencje rosnącą.**
- W odniesieniu do projektów SI, które zakończyły się niepowodzeniem częściowym, a zatem dla znacznej większości przedsięwzięć zrealizowanych (por. współczynnik sukcesu i niepowodzenie częściowe w tabeli 4), prawdopodobnie:
  - ✓ w okresie dekady przeciętne niepowodzenie w aspekcie jakościowym spadło o 8 punktów procentowych – z 58% w 1994 r. do 50% w 2004 r.;

- ✓ w latach 1994 – 2000 średnia skala niepowodzenia w aspekcie jakościowym znacznie się zmniejszyła;
- ✓ **po 2000 r. średnie niepowodzenie jakościowe ma wyraźną tendencje rosnącą.**



- brak wymaganych funkcji i cech w produkcie końcowym dla wszystkich zrealizowanych projektów SI (tj. za wyjątkiem porzuconych);
- hipotetyczny brak wymaganych funkcji i cech w produkcie końcowym dla projektów SI zakończonych niepowodzeniem częściowym (dla 2004 r. dane szacunkowe).

Rys. 8. Średnia skala niepowodzenia jakościowego projektów SI w latach 1994 – 2004 w odstępach dwuletnich
Źródło: Opracowanie własne na podstawie: [12], [15], [16] i [18].

Wprawdzie w całym analizowanym okresie niepowodzenie w aspekcie jakościowym nieco zmalało, jednak niewielka jak na dekadę skala zmian (niezależnie od grupy uwzględnianych projektów SI), a przede wszystkim jego wzrost po 2000 r. (dla każdej uwzględnianej na powyższym wykresie grupy projektów SI) nasuwają pewne wątpliwości co do wcześniej wysuniętego wniosku o rzeczywistym docenieniu w praktyce znaczenia czynników niepowodzenia informatycznych przedsięwzięć projektowych – **trudno bowiem stopień zapobiegania tym czynnikom w przypadku tak niskiego, niegdyś i obecnie, stopnia zgodności dostarczonego produktu końcowego z wymaganiami wyspecyfikowanymi przez użytkownika uznać nawet za „wystarczająco dobry"**. Tym bardziej, że wzrost niepowodzenia jakościowego to nie jedyna negatywna zmiana zanotowana w ostatnich latach.

7. Chaos is back?

Po opublikowaniu ostatnich rezultatów badań SG [12] pojawiły się opinie o powrocie chaosu do projektowania SI [5]. Świadczyć mają o tym następujące niepokojące zjawiska, które zaobserwowano w 2004 r. w zestawieniu z wynikami analiz za 2002 r.:

- wzrosło średnie niepowodzenie w aspekcie ilościowym (z 66 do 71%), w tym zwiększył się udział przedsięwzięć (por. tablica 4):
  - ✓ porzuconych (z 15 do 18%),
  - ✓ przekraczających planowany czas i budżet (z 51 do 53%);
- z tego powodu wzrósł prawdopodobnie (por. pkt 4) poziom strat z tytułu niepowodzenia ilościowego (przypuszczalnie o co najmniej 8 mld USD, tj. o 14,5% ich poziomu z 2002 r. - por. tablica 5);
- faktycznego zakończenia przedsięwzięć częściowo nieudanych należy się obecnie spodziewać po upływie przeciętnie dłuższego czasu w porównaniu z czasem planowanym (84% przewidywanego czasu wobec 82% dwa i 63% cztery lata wcześniej – por. rys. 6);
- obecnie do projektu przekraczającego planowane koszty trzeba w rzeczywistości dopłacić przeciętnie więcej w zestawieniu z szacowanym budżetem (56% przewidywanego budżetu wobec 43% dwa lata wcześniej – por. rys. 7);
- mimo wzrostu średnich faktycznych kosztów i przeciętnego rzeczywistego czasu realizacji projektów zakończonych niepowodzeniem częściowym (w stosunku do wielkości planowanych), produkty końcowe powstałe w ich wyniku nie są lepsze, a gorsze jakościowo (dalszy wzrost niepowodzenia jakościowego - por. rys. 8).

Zgodnie z opinią analityków ze *Standish Group* zasadniczą przyczynę tych negatywnych zmian stanowi fakt, że **wraz z zakończeniem kryzysu w branży IT realizowane informatyczne przedsięwzięcia projektowe ponownie stały się większe i droższe, czego następstwem jest zauważalny powrót do bardziej tradycyjnych praktyk w tworzeniu SI**, czyli odejście od iteracyjnego i elastycznego podejścia do projektowania, opartego na tzw. technikach „miękkich" i postrzeganego jako trudne do zastosowania przy dużych przedsięwzięciach [5]. Tradycyjne praktyki zaś zmniejszają szansę na zakończenie projektu SI powodzeniem, zarówno w aspekcie ilościowym, jak i jakościowym, jako że wymagają one zdefiniowania wszystkich wymagań na początku cyklu projektowego, co – jak dowodzi praktyka – nie jest możliwe. Stąd negatywny skutek w postaci ponownego wzrostu skali chaosu.

Skoro w historii badań *Standish Group* najmniejszą skalę niepowodzenia ilościowego informatycznych przedsięwzięć projektowych odnotowano w czasie ostatniego kryzysu, to oznacza to, że ów stan – w przeciwieństwie do zwiększania wydatków - sprzyja wzrostowi współczynnika sukcesu w realizacji projektów SI. I nic w tym dziwnego – w takim okresie bowiem skrupulatnie liczy się koszty, zawęża budżety, oszczędza na zasobach, czyli duże przedsięwzięcia podejmuje się stosunkowo rzadko. Kryzys zmusza zatem do podejmowania małych projektów, łatwiejszych do zarządzania i pozwalających na pełne zastosowanie projektowania przyrostowego, czego bezpośrednią konsekwencję stanowi spadek niepowodzenia w skuteczności ich tworzenia.

Ostatnie lata przyniosły więc pogorszenie sytuacji nie tylko pod względem jakościowym (po 2000 r.), ale również pod względem ilościowym (po 2002 r.) -

stąd opinia o powrocie chaosu. Czy jednak wyniki badań *Standish Group* nie przekonują nas o tym, że mieliśmy do czynienia z tym zjawiskiem w sposób ciągły? Trudno bowiem największy w historii badań współczynnik sukcesu w projektowaniu SI w wysokości 34% (2002 r.), któremu zresztą towarzyszył wzrost niepowodzenia w aspekcie jakościowym (por. rys. 8), uznać za świadczący o zaniku chaosu. Obserwowane w ostatnim czasie negatywne zmiany są więc raczej dowodem nie tyle na powrót, co na pogłębienie się tego zjawiska. Są również dowodem na przywiązywanie znów mniejszej wagi do czynników niepowodzenia informatycznych przedsięwzięć projektowych – **obecnie bowiem bez wątpienia nie wyczerpujemy wszystkich znanych i, jak pokazuje praktyka, stosunkowo prostych, a zarazem skutecznych możliwości zapobiegania upadkom projektów SI.**

8. Niedoceniane sposoby zapobiegania niepowodzeniom informatycznych przedsięwzięć projektowych

W opinii analityków *Standish Group* kluczem do sukcesu jest **właściwe zarządzanie zakresem projektu SI**, a w tym przede wszystkim **podejmowanie *małych* przedsięwzięć informatycznych** [5].

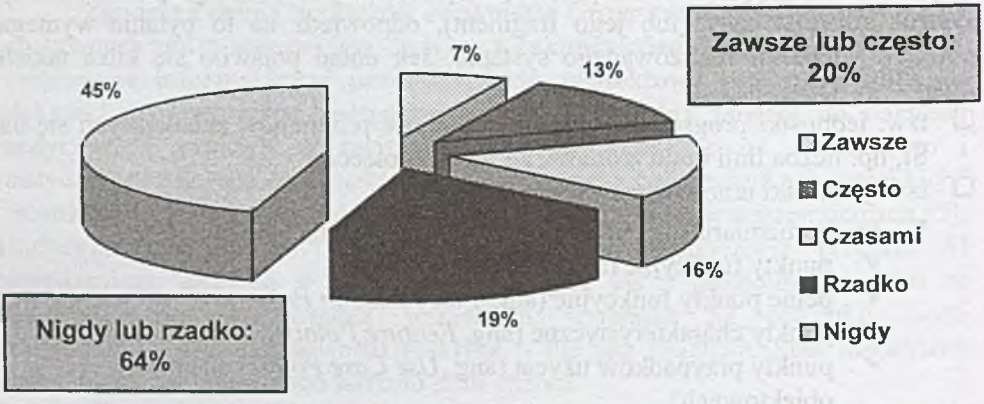
Powstaje więc pytanie: kiedy projekt SI można uznać mały? Przyjmując, że jest nim niewątpliwie takie przedsięwzięcie, w wyniku którego powstaje mały system informatyczny (lub jego fragment), odpowiedź na to pytanie wymaga pomiaru wielkości realizowanego systemu. Jak dotąd pojawiło się kilka takich miar:

- tzw. jednostki programowe – mierzą długość programów składających się na SI, np. liczba linii kodu źródłowego, liczba poleceń;
- tzw. jednostki umowne, a wśród nich:
  - miary rozmiaru funkcjonalnego (złożoności funkcjonalnej), np.:
    - ✓ punkty funkcyjne (ang. *Function Points*),
    - ✓ pełne punkty funkcyjne (ang. *Full Function Points*).
    - ✓ punkty charakterystyczne (ang. *Feature Points*),
    - ✓ punkty przypadków użycia (ang. *Use Case Points* - dla metod obiektowych),
    - ✓ punkty internetowe (ang. *Internet Points* – dla projektów stron web'owych);
  - miary złożoności konstrukcyjnej SI, np. punkty obiektowe (ang. *Object Points*).

Miary oparte na zwykłych (np. metoda IFPUG) oraz pełnych (np. metoda COSMIC-FFP) punktach funkcyjnych (PF) zostały zaakceptowane przez ISO jako międzynarodowy standard pomiaru rozmiaru funkcjonalnego SI (metoda IFPUG w części: ISO/IEC 20926: 2003; zaś metoda COSMIC-FFP w całości: ISO/IEC 19761: 2003) [2]. W dużym przybliżeniu można założyć, że 1 zwykły PF odpowiada ok. 125 liniom kodu w języku programowania klasy C.

W związku z powyższym jedna z możliwych odpowiedzi na pytanie o mały projekt SI brzmi następująco: mały projekt SI to taki, w wyniku którego ma zostać dostarczony system informatyczny (lub jego fragment) o wielkości nieprzekraczającej 1500 PF. Badania statystyczne informatycznych przedsięwzięć projektowych pokazują bowiem, iż istnieje pewien punkt graniczny rozmiaru SI, powyżej którego ryzyko niepowodzenia projektu SI wyraźnie wzrasta - przy dominującej dziś technologii (CASE, 4GL, RDBMS, różne warianty architektury klient/serwer) barierę tę określa się właśnie na około 1500 PF (ok. 180 - 190 tysięcy linii kodu w języku klasy C). Owe 1500 PF to zatem granica wielkości SI, poniżej której istnieje możliwość stosunkowo bezpiecznej jego realizacji. Duże projekty SI należy więc tak planować, aby jednorazowo budowany fragment SI charakteryzował się wielkością w granicach do 1500 PF, przy czym 1500 PF pięcioosobowy zespół może uzyskać w czasie około 7,5 - 8 miesięcy [19].

Realizacji małych projektów SI powinna sprzyjać również świadomość, że w typowym systemie informatycznym nie wykorzystuje się w ogóle niemal połowy dostarczonych (w odpowiedzi na wymagania użytkownika) funkcji i cech (por. rys. 9). Jest to kolejny wniosek płynący z badań *Standish Group*, w związku z którym J. Johnson proponuje zmianę spojrzenia na proces ustalania wymagań: zamiast zadawać sobie pytanie *„czy rzeczywiście mamy wszystko, czego potrzebujemy?"*, użytkownicy powinni zastanowić się *„czy rzeczywiście potrzebujemy wszystkiego, co mamy?"* [6].



Rys. 9. Średnie wykorzystanie funkcji i cech w typowym systemie informatycznym
Źródło: [6].

Co więcej, już przy takim granicznym rozmiarze tworzonego SI powinno się wykorzystywać narzędzia wspomagające zarządzanie zakresem. Jak bowiem wynika z badań *Software Productivity Research* (SPR), wiele projektów SI przekraczających planowany czas i/lub koszty konstruuje się przy niedbałych i zbyt optymistycznych, zwykle sporządzanych ręcznie, przewidywaniach. Te zaś przedsięwzięcia, przy których wykorzystuje się formalne narzędzia estymacji i pomiaru, oparte na przyjętych standardach i danych benchmarking'owych, charakteryzują się zdecydowanie lepszymi osiągnięciami: zwykle mieszczą się w budżecie i nie mają zbyt dużych opóźnień [7].

Znaczenie narzędzi wspomagających planowanie i kontrolę zakresu projektu SI rośnie wraz z rozmiarem tworzonego SI. Im bowiem realizowane przedsięwzięcie jest większe, tym jest ono zwykle bardziej złożone, a właśnie rosnący stopień złożoności stanowi zasadniczą przyczynę niepowodzenia "manualnych" sposobów pomiaru i estymacji rozmiaru konstruowanego SI, na tej podstawie zaś jego kosztów i czasu realizacji. W przedsięwzięciu takim występuje bowiem wiele czynników, które determinują rezultat tego procesu, a których kombinacji nie są w stanie uwzględnić proste algorytmy właściwe dla takich sposobów. Toteż, jak ocenia SPR, zautomatyzowane rozwiązania służące wymiarowaniu projektów SI są zdecydowanie skuteczniejsze od manualnych szacunków już dla aplikacji o rozmiarze większym od 500 PF. Dla SI o wielkości powyżej 5000 PF oceny manualne nigdy nie są wystarczająco dokładne w zestawieniu z automatycznymi. Wykorzystywane w celu estymacji narzędzia mają bowiem możliwość dynamicznego wspomagania przez cały czas realizacji przedsięwzięcia kolejnych szacunków rozmiaru, czasu i kosztów, co ma szczególne znaczenie przy długotrwałym cyklu projektowym. W praktyce zatem sukces osiąga się przy na tyle dojrzałych standardach zarządzania projektami SI, że dają się one częściowo zautomatyzować [7].

Podejmowanie małych projektów SI z oszacowanymi we właściwy sposób (na podstawie przewidywanego rozmiaru) kosztami i czasem realizacji oraz konstruowanie ich w sposób przyrostowy i elastyczny przy wykorzystaniu narzędzi wspomagających zarządzanie projektowaniem (w tym przede wszystkim zakresem i wymaganiami) to odpowiedź na pytanie o znane i - jak pokazuje praktyka - stosunkowo proste, a zarazem skuteczne możliwości zapobiegania upadkom informatycznych przedsięwzięć projektowych, które nie są dzisiaj w wystarczającym stopniu doceniane.

Chociaż o znaczeniu iteracyjnego i elastycznego podejścia do projektowania oraz narzędzi wspomagających ów proces jest wyraźnie mowa dopiero w badaniach SG za rok 2004 (por. tablica 3), to przecież brak minimalizacji zakresu projektu, jak również brak wiarygodnych oszacowań, które nie są możliwe do przeprowadzenia w sposób właściwy bez stosowania odpowiednich miar wielkości konstruowanego SI, to czynniki istniejące na liście przyczyn niepowodzenia projektów SI już w 2000 r. (por. tablica 2), a zatem przed ponownym wzrostem skali chaosu. Dawało to szansę, niestety niewykorzystaną, rzeczywistego zapobiegania im w praktyce. O ile zatem dzisiejsze trzy pierwsze czynniki upadku informatycznych przedsięwzięć projektowych, tj. brak zaangażowania użytkowników w działania projektowe, niedostateczne wsparcie przedsięwzięcia ze strony sponsora, niejasne cele biznesowe (wszystkie wspomniane w raporcie sprzed dziesięciu lat), są już dla zarządzających projektami SI oczywiste, stosunkowo umiejętnie bowiem zapobiega się im w praktyce, o tyle konieczność:

- minimalizacji/optymalizacji zakresu projektu (obecnie czynnik krytyczny na miejscu 4.),

- prowadzenia przedsięwzięcia w sposób przyrostowy i elastyczny (dzisiaj miejsce 6.),
- właściwego zarządzania finansami projektu, w tym uzyskiwania wiarygodnych oszacowań co do rozmiaru kontruowanego SI, a na tej podstawie czasu i kosztów przedsięwzięcia (obecnie miejsce 7.),
- wykorzystywania standardowych narzędzi wspomagających zarządzanie projektowaniem (dziś miejsce 10.),

to czynniki niepowodzenia, do których nadal podchodzi się bardzo sceptycznie, nie doceniając ich wagi. Zdaniem Johnsona, zauważalny sceptycyzm specjalistów wobec na przykład nowych, tzw. „miękkich" technik projektowania, właściwych dla przyrostowej i elastycznej realizacji tego procesu, wynika z tego, iż *„nie rozumieją oni, że te miękkie metodyki wcale nie oznaczają odejścia od formalizmów, zasad, które trzeba przestrzegać. Jest wręcz odwrotnie. Techniki miękkie wymagają konsekencji i jasnych reguł. (...) Wydaje się, że są odpowiedzią na inne niż dawniej wymagania biznesu, a przede wszystkim potrzebę szybszego niż dawniej realizowania systemów w znacznie bardziej dynamicznym i zmiennym środowisku, w którym w oczywisty sposób w trakcie wdrożenia zmieniają się wymagania i założenie projektowe. Tu nie ma odwrotu."* [3].

Zważywszy skalę niepowodzeń w realizacji projektów SI wymaganiem teraźniejszości wydaje się być przekonanie osób odpowiedzialnych za zarządzanie takimi przedsięwzięciami o istotnym znaczeniu tych czynników. **Praktyka pokazuje bowiem, że zdecydowanie zbyt rzadko zdajemy sobie sprawę z wagi problemu i z możliwości jego rozwiązania, tracąc z tego powodu poważne kwoty, wydatkowane na projekty SI, które albo nie spełniają wymagań użytkowników i/lub są realizowane w czasie i po kosztach znacznie odbiegających od wartości planowanych, albo zostają porzucone.**

9. I znowu wszystkiemu winny czynnik ludzki

Tezę dotyczącą wspomnianego powyżej sceptycyzmu potwierdza tzw. paradoks Cobba[10]: *"Wiemy, dlaczego projekty upadają, wiemy jak zapobiec tym upadkom – więc dlaczego one ciągle upadają?"* [18]. Wobec zaprezentowanych powyżej faktów autorka tego opracowania stawia niniejszym dodatkowe pytanie: *I dlaczego projekty SI nadal będą - w tak dużej przypuszczalnie skali - upadać?*

Prezentując dane *Standish Group* dotyczące ostatniej dekady pokazano, że rzeczywiste docenienie znaczenia czynników niepowodzenia informatycznych przedsięwzięć projektowych nie pozostaje bez wpływu na zmniejszenie skali chaosu w ich realizacji. Toteż problem nie tkwi w tym, że nie potrafimy w praktyce skutecznie zapobiegać przyczynom upadków, że istnieją tzw. czynniki obiektywne, na które nie mamy wpływu, bo to nie z ich winy większość przedsięwzięć kończy się niepowodzeniem, a w tym, że nie wykorzystujemy do tego celu wszystkich

[10] Martin Cobb to pracownik Kanadyjskiej Rady Skarbu Państwa, a swój paradoks sformułował on na konferencji *CHAOS University* w listopadzie 1995 r. w Chatham.

możliwości. Dlaczego? Innymi słowy, jakie są praprzyczyny upadków projektów SI?

Bardzo dosadnie rzecz ujmuje Edward Yourdon, twierdząc, że *„(...) Wszystkie kłopoty wynikają ze stosowania niewłaściwych metod (albo z pracowania w ogóle bez metod), złych narzędzi lub też nieodpowiednich technik zarządzania projektem. Innymi słowy, przyczyną marszy ku klęsce jest nasza głupota albo nieudolność."* [20]. W tym twierdzeniu jest oczywiście sporo przesady - opiera się ono zapewne na założeniu, że „cel uświęca środki" - trudno jednak zaprzeczyć, że ów niekwestionowany autorytet w praktycznym tropieniu powodów „marszy ku klęsce" wytacza właściwy kierunek poszukiwań. Dopatruje się on bowiem przyczyn chaosu nie w specyfice produktu, nie w błędach technologicznych, nie w braku odpowiednich źródeł wiedzy czy właściwych doświadczeń, nie w niezwykle dynamicznej zmienności IT i otoczenia biznesowego, nie w braku podobieństw między inżynierią oprogramowania a inżynierią lądową (jak to czynią niektórzy, np. w [12], [7]), ale w przyczynach zależnych od uczestników informatycznych przedsięwzięć projektowych, a przede wszystkim od osób odpowiedzialnych za zarządzanie projektami SI.

I ten sposób rozumowania pokrywa się z ogólnym wnioskiem płynącym z badań *Standish Group*: *"W 1906 roku na wschodnim wybrzeżu miało miejsce 177 katastrof statków morskich. Owe katastrofy spowodowane były wypadkami, złym zarządzaniem, zaniedbaniami, błędami w ocenie sytuacji i błędami osób pilotujących statki – **wszystkie te przyczyny przypisuje się błędom ludzkim** (w przeciwieństwie do błędów technologicznych). **To samo wynika również z raportu Standish Group dotyczącego błędów w tworzeniu projektów informatycznych.**"* [13].

Praprzyczyny niepowodzenia projektów SI leżą zatem w omylnej naturze ludzkiej – czy jednak tylko po stronie osób zarządzających nimi? Okazuje się, iż po wielu latach analiz J. Johnson uzupełnia ogólny, powyżej zaprezentowany wniosek płynący z badań SG, zauważając, iż nierzadko kierownicy projektów SI są zmuszeni zmierzyć się z takimi postawami użytkowników, które nawet najbardziej kompetentnym zarządzającym odbierają szansę na zakończenie wspólnego wszak przedsięwzięcia sukcesem. Ów bowiem skutecznie udaremniają spotykane czasami postawy odbiorców SI, które charakteryzują się przesadną ambicją, arogancją, ignorancją, niegospodarnością (zauważalną zwłaszcza w administracji publicznej), świadomym brakiem aktywności i chęci do współpracy, co skutkuje brakiem rzeczywistego zaangażowania użytkownika w realizację projektu, oraz niechęcią do szybkiego podejmowania decyzji ([18], [3]). W opinii Johnsona one także stanowią źródłowe powody upadków projektów SI [18].

Chaos w projektowaniu SI spowodowany jest zatem czynnikami zależnymi od uczestników tego procesu: z jednej strony zarządzający informatycznymi przedsięwzięciami projektowymi popełniają błędy wynikające głównie z niewystarczającego doceniania w praktyce możliwości skutecznego zapobiegania upadkom projektów SI, z drugiej strony – mogą oni napotkać na bariery wynikające z niewłaściwych postaw występujących czasami po stronie odbiorców SI. Trudno zaś oczekiwać nie tylko tego, żeby winna tym praprzyczynom

niepowodzeń natura ludzka uległa zmianom, ale nawet tego, aby owe praprzyczyny – owszem, kontrowersyjne i niepopularne - zostały w najbliższym czasie właściwie odebrane i docenione przez większość uczestników informatycznych przedsięwzięć projektowych. Tymczasem przekonanie o ich znaczeniu odbiorców SI, np. o racjonalności ich rzeczywistego zaangażowania w projekt, leży w interesie i gestii osób zarządzających procesem projektowania, jednak – jak się wydaje - oni sami są z reguły nastawieni do tych praprzyczyn upadków zbyt sceptycznie. Być może wynika to z powszechnego w środowisku IT przywiązania do poglądu, że sukces projektu SI zależy od jasnych, algorytmizowalnych reguł i formalizmów – a ich gwarancji na próżno szukać w nieprzewidywalnej naturze ludzkiej, zwłaszcza w zespole ludzi. Na zmniejszenie tego, skądinąd również naturalnego, sceptycyzmu wobec owych źródłowych przyczyn trzeba zapewne jeszcze trochę poczekać – dopiero jak zostaną one docenione pojawi się realna szansa na znaczny i trwały spadek skali chaosu w projektowaniu SI. Dlatego projekty SI, przynajmniej w najbliższym czasie, nadal będą - w tak dużej przypuszczalnie skali - upadać. A więc znów sceptycyzm...? Raczej świadomość, że w obszarze skuteczności realizacji informatycznych przedsięwzięć projektowych jest jeszcze wiele do zrobienia.

Literatura:

1. Babcock D. L., Morse L. C., *Managing Engineering and Technology: An Introduction to Management for Engineers*, Third Edition, Prentice Hall, Upper Saddle River, New Jersey 2002.

2. Czarnacka-Chrobot B., *Pomiar rozmiaru funkcjonalnego systemu informatycznego - sposób na "chaos permanens"?*, w: *Współczesne kierunki rozwoju informatyki*, praca zbiorowa pod red. P. W. Fuglewicza, B. Jackowskiego, E. Mizerskiej, A. Ostaszewskiej, Materiały XX Jesiennych Spotkań PTI, Polskie Towarzystwo Informatyczne, Katowice-Mrągowo, listopad 2004.

3. Gamdzyk P., *Koszt i ryzyko*, Computerworld nr 27/2005, 4.07.05.

4. Hall P., *Avoiding Project Failure*, SM /ASM 2002 Conference, StickyMinds.com, por. http://www.stickyminds.com/sitewide.asp?ObjectId=3365&Function=DETAILBROWSE&ObjectType=ART (5.10.05).

5. Hayes F., *Chaos is back*, Computerworld, November 8, 2004, por. http://www.computerworld.com/printthis/2004/0,4814,97283,00.html (27.10.05).

6. Johnson J., *Standish Group Study*, The XP 2002 Conference, Sardinia, May 2002.

7. Jones C., *Software Project Management in 21st Century*, American Programmer, Vol. XI, No. 2; February 1998.

8. Karten N., *Ten Ways to Guarantee Project Failure*, StickyMinds.com, 4.07.03, por. http://www.stickyminds.com/sitewide.asp?ObjectId=6370&Function=DETAILBROWSE&ObjectType=COL (4.10.05).

9. Kerzner H., *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, Eight Edition, Wiley, John & Sons, New York 2003.

10. Meredith J. R., Mantel S.J., Jr., *Project Management, A Managerial Approach*, Sixth Edition, Wiley, John & Sons, New York 2005.

11. Parker W., *The Eight Keys to Project Management Failure*, The Workstar Library, 2003 r., por. http://workstar.net/library/pm1.htm (5.10.05).

12. *The CHAOS Report,* The Standish Group International, Inc., West Yarmouth, Massachusetts 1995.

13. *Unfinished Voyages*, The Standish Group International, Inc., West Yarmouth, Massachusetts 1996.

14. *CHAOS: A Recipe for Success,* The Standish Group International, Inc., West Yarmouth, Massachusetts, 1999.

15. *The CHAOS Chronicles II,* The Standish Group International, Inc., West Yarmouth, Massachusetts, 2001.

16. *The CHAOS Chronicles III,* The Standish Group International, Inc., West Yarmouth, Massachusetts, January 2003.

17. *Standish: Project Success Rates Improved Over 10 Years*, Software Mag.COM, The IT Software Journal, por. http://www.softwaremag.com/L.cfm?Doc=newsletter/2004-01-15/Standish (27.10.05).

18. Johnson J., *CHAOS Rising*, The Standish Group International, Inc., Materiały konferencyjne z *II-giej Krajowej Konferencji Jakości Systemów Informatycznych*, Computerworld, czerwiec 2005.

19. Stokalski B., *Grać aby wygrać. Ryzyko i zarządzanie projektami*, InfoVide, 1998.

20. Yourdon E., *Marsz ku klęsce. Poradnik dla projektanta systemów*, WNT, Warszawa 2000.

ROZDZIAŁ VI

JAKOŚĆ JAKO PODSTAWA EFEKTYWNEGO WDROŻENIA SYSTEMU INFORMATYCZNEGO

Anna KACZOROWSKA, Anna PAMUŁA

W wieku powszechnej informatyzacji i popularności norm z serii ISO 9000 jednym z najważniejszych czynników różnicujących ofertę firm jest jakość wyrobów i usług. Statystyki niepowodzeń dużych przedsięwzięć informatycznych pokazują właśnie, że jedną z najistotniejszych przyczyn takiego stanu rzeczy jest nieuruchamianie przez firmy realizujące projekty procesów doskonalenia jakości.

1. Jakość systemu informatycznego

Powstało wiele definicji jakości. Każda z nich rozpatruje inny aspekt jakości, w inny sposób lub w innym kontekście. Z każdej wynikają także inne wymagania na produkt, system wytwórczy, sposób obsługi klienta, itd. Jeśli jakość ma być osią porządkującą wszystkie działania firmy, należy przyjąć dowolną jej definicję, ale konsekwentnie i formalnie wprowadzoną w całej firmie i jawnie zakomunikowaną wszystkim kooperantom i klientom.

Próbując zdefiniować pojęcie jakości należy przyjąć, że nasze sądy jakościowe pojawiają się zawsze w odniesieniu do konkretnego wyrobu, usługi czy jakiegoś stanu rzeczy. Nie istnieje uogólnione pojęcie wysokiej i niskiej jakości. *Jakość jest zatem relacją, dynamiczną i zmienną.* Relacja ta istnieje tylko wraz z naszym nastawieniem do danego przedmiotu wartościowanego jakościowo, a to oznacza, że każdy człowiek tworzy własną, oddzielną relację, za każdym razem inną. Każda relacja budowana jest na podstawie zróżnicowanego i zmiennego katalogu kryteriów, które sumarycznie kształtują poziom jakości danego produktu. Na kryteria te ma wpływ szereg zmiennych związanych ze sposobem patrzenia na świat, wartościami, do których jesteśmy przywiązani lub przekonani, kontekstem danej sytuacji. *Jakość jest zatem również subiektywna i kontekstowa.*

Zarządzanie jakością najczęściej definiuje się jako ogół działań zmierzających do określenia i wdrożenia w firmie polityki jakości. Z kolei polityka jakości jest oficjalnym wyrażeniem tego jak dana organizacja rozumie jakość swoich produktów i jak chce tę jakość zapewnić.

W rzeczywistości, jakość produktu końcowego – systemu informatycznego, będzie wypadkową jakości różnorodnych, złożonych i zmiennych procesów - sumą jakości wymagań, jakości projektowania, jakości wykonania, jakości zarządzania, jakości użytkowania, itd. Każdy element systemu wytwórczego i każda relacja między jego elementami ma wpływ na jakość końcową. Samo mądre i przemyślane stworzenie właściwych warunków produkcyjnych, w których produkt osiągnie najwyższą jakość nie wystarczy. Na pierwszym miejscu listy czynników

wpływających na jakość będzie zawsze człowiek, jego wiedza, umiejętności, doświadczenie i motywacja. Oprócz czynników wewnętrznych, znajdujących się w systemie wytwórczym, a mających wpływ na jakość występuje również wiele innych czynników, które działają w szeroko rozumianym otoczeniu produkcyjnym i o ile na te pierwsze, w pewnym zakresie, mamy wpływ, to na te drugie – zewnętrzne, znikomy. Czynniki zewnętrzne określają np. tzw. wymagany poziom jakości, czyli kryteria, które decydują o stopniu dopasowania produktu do oczekiwań społecznych, przyjętych norm i standardów. Określają także uwarunkowania realizacyjne, które ograniczają bądź zmuszają producenta/dostawcę do podjęcia określonych działań. Ich cechą wspólną jest to, że są zmienne i nieprzewidywalne. Bardzo trudno odnosić się do nich na bieżąco, zwykle oddziałują na produkt i procesy wytwórcze bezpośrednio lub pośrednio w najmniej oczekiwanych momentach. Takie zjawiska jak tradycja, zwyczaje, moda, poziom kultury, powodują, że produkt, którego jakość sprawdziła się w jednym kraju niekoniecznie musi się potwierdzić w innym ([1]}. Nie każdy produkt niesie tak uniwersalne wartości jak system operacyjny Windows. Zabezpieczenie właściwej jakości nie tylko technicznej, ale takiej, która na danym rynku się sprzeda nie jest sprawą trywialną i wymaga rozbudowanych badań i zabiegów marketingowych oraz analizy ryzyka.

Na użytek zarządzania jakością systemu informatycznego realizowanego poprzez ustanowienie projektu pojęcie *jakość należy rozumieć jako ogół cech i właściwości produktu decydujących o jego zdolności do zaspokojenia stwierdzonych lub przewidywanych potrzeb klienta.* Dlatego też, należy nieustannie badać wymagania i oczekiwania Klienta a za jakość uczynić odpowiedzialnym cały system firmy, a nie tylko dział testów (kontroli technicznej). Co więcej, należy stworzyć mechanizmy nieustannego doskonalenia jakości produktu i procesów zachodzących w firmie.

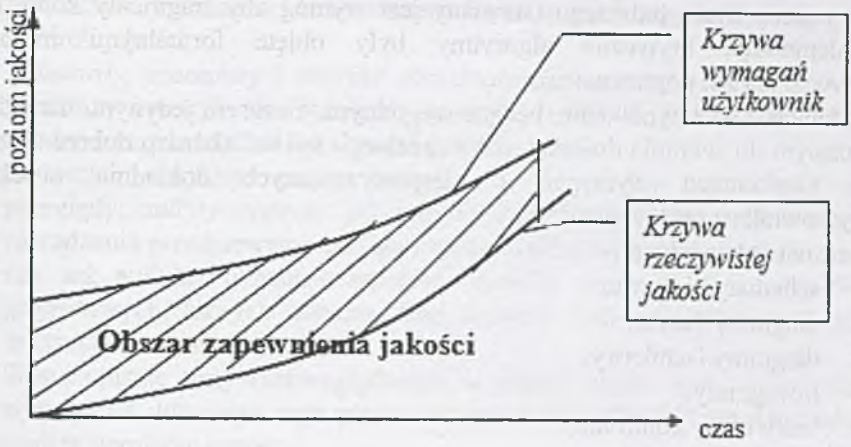
2. Sposoby zapewnienia jakości oprogramowania

Według definicji ISO 9000: *zapewnienie jakości są to wszystkie zaplanowane i systematyczne działania, które są niezbędne do uzyskania i utrzymania odpowiedniego stopnia wiarygodności, że wyrób lub usługa spełni ustalone wymagania jakościowe.* Zapewnienie jakości leży więc w obszarze między wymaganiami użytkownika, a tym co w rzeczywistości powstaje (rys. 1). Wymaga dołożenia wszelkich starań, aby krzywe oczekiwanej i rzeczywistej jakości systemu kiedyś (najlepiej przed trafieniem produktu do użytkownika) się przecięły.

Powołując zespół kontroli jakości projektu nie należy w pierwszej kolejności myśleć o możliwych do zastosowania metodach i narzędziach, ale o ich umiejscowieniu i skutecznym wykorzystaniu w całym cyklu życia systemu informatycznego. Do podstawowego zastawu metod wspierających zapewnienie jakości systemu informatycznego, oddziałujących na system wytwórczy i organizacyjny firmy, należą:

- inżynieria wymagań,
- metodyka projektowania,

- weryfikacja i walidacja,
- inne narzędzia.



Rys. 1. Krzywe rzeczywistej i wymaganej jakości systemu informatycznego

Źródło: Opracowanie na podstawie [6]

Walidacja jest procesem oceny oprogramowania na zakończenie procesu produkcyjnego (lub fazy) w celu sprawdzenia czy jest wolne od wad i niezgodności. *Weryfikacja* natomiast, jest procesem określenia czy produkt danej fazy produkcyjnej spełnia wymagania postawione w fazie poprzedniej. Koncentruje się na znalezieniu tych błędów i niezgodności, które mogą stać się przyczyną wad produktu końcowego. Weryfikacja i walidacja dotyczą wszelkich produktów poszczególnych faz cyklu wytwórczego, a więc wymagań, specyfikacji, projektu, implementacji, konfiguracji, zmian, itp., a nie tylko ostatecznej postaci systemu.

Wyróżnia się następujące sposoby weryfikacji i walidacji oprogramowania:

- śledzenie wymagań, będące zestawem technik koncentrujących się na szczegółowym i udokumentowanym śledzeniu wymagań użytkownika od momentu ich powstania, przez analizę, projekt, implementację i testowanie, z zamiarem najlepszego ich zrealizowania,
- przeglądy techniczne oprogramowania, które są formą testowania oprogramowania „na sucho" i mogą przebiegać jako:

przeglądy wzajemne (nieformalne przeglądy, których celem jest akceptacja danej postaci produktu),

inspekcje (przeglądy, recenzje), tzn. sformalizowane i udokumentowane przeglądy wyników pracy, których celem jest znalezienie błędów i problemów oraz podjęcie działań zmierzających do ich usunięcia,

audyty (niezależne, formalne i udokumentowane przeglądy produktów i procesów, których celem jest ocena i wskazanie kierunków poprawy procesów produkcyjnych oraz produktów),

- testowanie oprogramowania (pochłaniające niekiedy 40% kosztu oprogramowania; testowaniu jest poświęcony par. 4.2),
- dowodzenie poprawności, wymagane zwłaszcza w sytuacjach, gdzie oprogramowanie bezpośrednio odpowiedzialne jest za bezpieczeństwo lub utrzymanie życia ludzkiego (stawiany jest wymóg aby fragmenty kodu, które implementują krytyczne algorytmy były objęte formalnymi metodami dowodzenia ich poprawności),
- symulacje i prototypowanie, będące wygodnym, a czasem jedynym, narzędziem służącym do zebrania doświadczeń z „realnego świata" (bardzo dobrze służy też do konfrontacji wymagań i niesprecyzowanych dokładnie oczekiwań użytkownika z rzeczywistością),
- inne narzędzia jak np.:
  - schematy blokowe,
  - diagramy Pareto,
  - diagramy Ischikawy,
  - histogramy,
  - karty i listy kontrolne,
  - wykresy korelacji,
  - księga jakości,
  - metryki jakości,
  - statystyczna kontrola procesów.

3. Plan jakości projektu

Narzędziem, które ma zabezpieczyć projekt przed przekroczeniem terminów i kosztów zawartych w kontraktach projektowych jest ujęcie tych działań w formie planu jakości. Jest on szczegółowym zapisem działań, mających na celu wdrożenie ogólnych postanowień sformułowanych jako polityka firmy w kontekście konkretnego produktu/projektu. Plan taki powinien powstać z zamiarem stworzenia dokumentacji wyszczególniającej specyficzne sposoby, środki i kolejność postępowania związane z jakością systemu informatycznego będącego produktem danego przedsięwzięcia. Plan ten określa jednoznacznie role, strukturę i zasoby odpowiedzialne za powstanie systemu informatycznego, spełniającego ustalone wymagania klienta.

Szablonowy dokument planu zapewnienia jakości, opracowany według zaleceń normy IEEE 730.1-1995 ([8]), wyróżnia następujące jego składowe:

1. *Wstęp*: zawiera definicję polityki jakości przedsięwzięcia i wynikający z niej cel oraz zakres zapewnienia jakości w projekcie; uwzględniony jest w nim zakres dokumentu (panu zapewnienia jakości), przeznaczenie, struktura i zawartość jego punktów, a także referencje do innych dokumentów projektowych,
2. *System zapewnienia jakości*: określa strukturę organizacyjną systemu jakości i jej relacje do struktury organizacyjnej przedsięwzięcia; charakteryzuje role w systemie jakości, ich zadania i przypisane do ich realizacji zasoby; opisuje ustalone dla projektu zasady zarządzania jakością oraz zasady monitorowania i

doskonalenia systemu jakości,

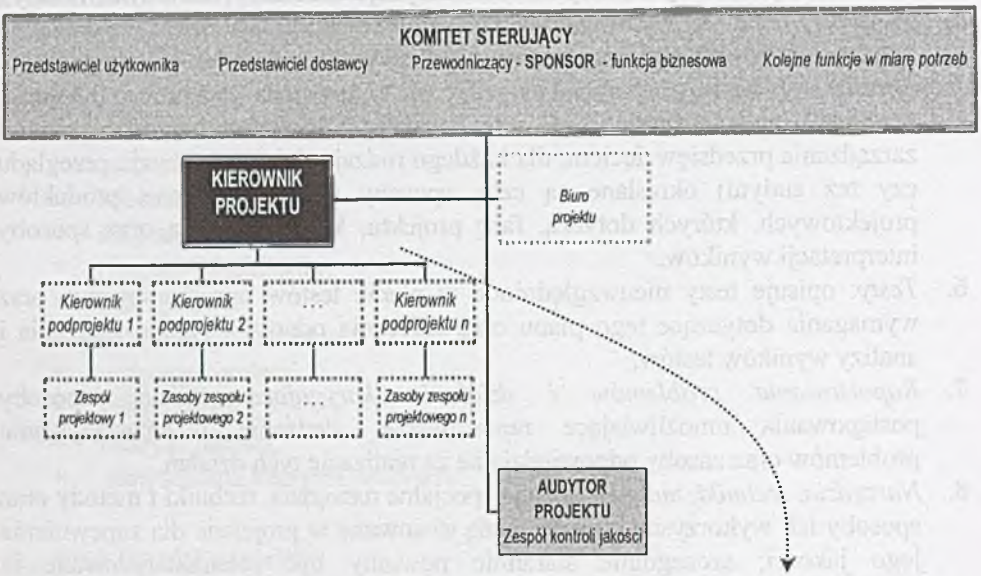
3. *Dokumentacja*: określa i opisuje plany zarządzania przedsięwzięciem, dokumentację projektową, plany weryfikacji i pielęgnacji produktów oraz dokumentację użytkową; przedstawia kryteria jakości poszczególnych produktów (zwłaszcza punktów kontrolnych w harmonogramie) oraz sposoby kontroli jakości,
4. *Standardy, procedury i metryki*: charakteryzuje standardy (metodyki, normy), procedury i metryki stosowane dla zapewnienia jakości oraz zasady badania ich zgodności z produktami i procesami projektowymi,
5. *Recenzje, przeglądy, audyty* (par. 2): są to zarówno techniczne recenzje, przeglądy, audyty systemu jak i te związane z różnymi innymi aspektami zarządzania przedsięwzięciem; dla każdego rodzaju działań (recenzji, przeglądu czy też audytu) określane są cele, sposoby realizacji, zakres produktów projektowych, których dotyczą, fazę projektu, którą obejmują oraz sposoby interpretacji wyników,
6. *Testy*: opisuje testy nieuwzględnione w planie testów oprogramowania oraz wymagania dotyczące tego planu oraz zalecenia odnośnie przeprowadzenia i analizy wyników testów,
7. *Raportowanie problemów i działania korygujące*: opisuje sposoby postępowania umożliwiające raportowanie, śledzenie i rozwiązywanie problemów oraz zasoby odpowiedzialne za realizację tych działań,
8. *Narzędzia, techniki, metody*: opisuje specjalne narzędzia, techniki i metody oraz sposoby ich wykorzystania, które będą stosowane w projekcie dla zapewnienia jego jakości; szczególnie starannie powinny być scharakteryzowane te narzędzia, które dopiero trzeba będzie utworzyć na potrzeby konkretnego przedsięwzięcia,
9. *Wytyczne do zarządzania konfiguracją*: uwzględnia wskazówki, co do sposobu zarządzania konfiguracją sytemu informatycznego, dokumentów i produktów projektowych oraz metod ich identyfikowania, składowania, poprawiania, itp.,
10. *Zapisy jakości*: charakteryzują system pozyskiwania, gromadzenia, zarządzania i usuwania informacji dotyczących przeprowadzonych działań w ramach systemu jakości oraz możliwych do wykonania analiz statystyczno – historycznych,
11. *Szkolenia*: uwzględnia opis zakresu, tematyki i wymagań na niezbędne szkolenia, umożliwiające nabycie kompetencji koniecznych do wytworzenia produktów projektowych o wymaganym poziomie jakości,
12. *Wdrożenie planu zapewnienia jakości*: zawiera zalecenia co do procesów oceny, rozpowszechnienia, planowania zadań i wykonania planu zapewnienia jakości.

Za realizację planu jakości odpowiedzialni są wszyscy pracownicy stykający się z produktem. Pozytywne czy negatywne wyniki kontroli jakości są po analizie źródłem decyzji, które zmierzają do:

- dokumentowania działań,
- podjęcia działań korekcyjnych, śledzenia ich realizacji oraz weryfikacji ich skuteczności.

W przypadku systemów informatycznych realizowanych poprzez

ustanowienie projektu odpowiedzialność za kontrolę projektu pod względem zachowania wymaganych standardów jakości, finansowych i technicznych zawsze spoczywa na komitecie sterującym i nie można jej przenieść, ale w celu jej realizacji można tę funkcję delegować na członków zespołu jakości. Właśnie zadania zagwarantowania wymaganego poziomu jakości w projekcie obsługiwane są przez ten zespół (rys. 2).



Rys. 2. Ramowa struktura organizacyjna projektu

Źródło: Opracowanie własne z wykorzystaniem [9, str. 34]

4. Najtrudniejsze i najważniejsze pod względem zapewnienia jakości etapy podczas wdrożenia systemu informatycznego

4.1. Analiza wymagań

Większość błędów, a szczególnie te najpoważniejsze powstają w fazie analizy wymagań. Problem polega na tym, że do projektowania i kodowania używa się sformalizowanych języków opisu, podczas gdy wymagania użytkownika są wyrażone językiem naturalnym. Stwarza to możliwość powstawania niezgodności spowodowanych niezrozumieniem i niespójnością wymagań, a te z kolei prowadzą do błędów w programie. Problemy tego rodzaju są bardzo trudne do wykrycia podczas przeglądów projektu i testów, gdyż procesy walidacji odnoszą się tylko do takich wymagań, jakie postawiono i wyspecyfikowano. Nieprzewidziane sytuacje zawsze mogą się pojawić już w trakcie działania programu u użytkownika.

Ważniejszymi źródłami wad w wymaganiach użytkownika są: nieprawidłowość, niespójność, niejasność, nielogiczność czy opuszczenie.

Dlatego też, szczególnie istotne jest stosowanie najefektywniejszych narzędzi do zarządzania jakością wymagań użytkowników.

Na omówienie zasługuje metoda Quality Function Deployment (QFD), gdyż w istocie jest to metoda gromadzenia, analizy i specyfikacji wymagań użytkownika, który to etap powoduje dotychczas najwięcej i najpoważniejszych błędów w ramach tworzenia systemu informatycznego. Nazwę tej metody można przetłumaczyć jako Rozpisanie Funkcji Jakości. QFD powstała pod koniec lat 60-tych w japońskiej stoczni koncernu Mitsubishi w Kobe. Przez lata 70-te dojrzewała w różnych gałęziach przemysłu japońskiego, a pod koniec lat 80-tych trafiła do USA i Europy, gdzie zyskała ogromną popularność jako narzędzie doskonalenia jakości produktów, u którego podstaw leżą następujące założenia:

- pomyślna realizacja produktu przekracza możliwości pojedynczych jednostek i zawsze jest skutkiem pracy zespołowej i międzywydziałowej (specjalistów od marketingu, analityków, projektantów, programistów, itd.),
- współdziałanie wielu zespołów, korzystających z różnych narzędzi i języków opisu produktu musi być uporządkowane i posiadać odpowiednią strukturę,
- zawsze brakuje środków na realizację wszystkich wymagań w stopniu zadowalającym, konieczna jest nadanie im właściwych priorytetów i skoncentrowanie się na tych najważniejszych,
- zawsze najważniejszy jest punkt widzenia klienta, który ma istotny wpływ na ważność poszczególnych wymagań oraz końcową akceptację ich realizacji.

Korzyści ze stosowania tej metody należy upatrywać w oparciu procesu opracowywania wymagań na pracy zespołowej, przekraczaniu barier międzywydziałowych, redukcji zmian w procesie projektowania nawet do 50%, skróceniu cykli projektowych oraz wsparciu dla innych metodyk pozyskiwania i analizy wymagań użytkownika.

Podstawowym narzędziem metody jest tzw. dom jakości (*house of quality*), pozwalający na wydobycie związków między wymaganiami (wiersze), a metodami ich realizacji (kolumny). Wagi przypisane poszczególnym wymaganiom wpływają na sumaryczne wagi poszczególnych elementów produktu nadając im właściwy priorytet. Wielokrotne operacje uściślania parametrów domu jakości wymagają współpracy między różnymi działami firmy, a przede wszystkim zaangażowania w ten proces klienta (użytkownika).

Proces pozyskiwania, analizy, specyfikacji i nadawania priorytetów wymaganiom jest podzielony na 4 fazy:

1. planowanie produktu, gdzie wymagania klienta są konfrontowane z wymaganiami funkcjonalnymi i założeniami projektowymi systemu,
2. projekt produktu, gdzie wymagania funkcjonalne są odwzorowywane na architekturę systemu,
3. planowanie procesu, gdzie komponenty systemu są konfrontowane z poszczególnymi fazami produkcyjnymi, metodykami, narzędziami,
4. planowanie produkcji, gdzie poszczególne fazy realizacyjne, metodyki, itd. są rozpisywane między dostępne zasoby projektowe.

Wszystkie 4 fazy uzupełniane są dodatkowymi elementami analizy jak np. tabele wariantowania rozwiązań, czy tabele rozpisania kryteriów jakości.

Tab. 1. Tabela z zestawieniem uniwersalnych kryteriów ważności i kryteriów jakości produktu projektowego – harmonogramu projektu dla analizy ważności poszczególnych kryteriów jakości (za pomocą metody uniwersalnych kryteriów ważności przedstawionej w szacie graficznej house of quality, gdzie we wierszach zapisane są wymagane przez klienta kryteria jakości dla harmonogramu projektu, a w kolumnach ujęte są uniwersalne kryteria jakości wraz z nadanymi im stopniami ważności)

Kryterium jakości dla produktu (harmonogramu projektu)	Symbole uniwersalnych kryteriów ważności										Iloczyn
	A	B	C	D	E	F	G	H	I	J	
	Stopnie kryteriów ważności [1...10]										
	3	7	9	8	5	4	10	4	2	6	
Estymacja czasu trwania	1	5	5	4	2	0	5	1	0	0	0
Estymacja przydziału zasobów	1	4	5	4	2	0	5	1	0	0	0
Prezentacja graficzna	0	1	1	0	0	5	3	0	0	1	0
Kompletność harmonogram.	0	1	5	5	3	1	5	2	0	5	0
Ilość punktów kontrolnych	1	1	4	4	3	2	4	4	3	5	23040
Suma punktów	3	12	20	17	10	8	22	8	3	11	

Źródło: opracowanie własne z wykorzystaniem [7]

4.2. Testowanie oprogramowania

Testowanie jest ważnym i kosztownym etapem realizacji projektu informatycznego, pochłaniającym, w systemach o wysokich wymaganiach niezawodnościowych, wartość pięciokrotnego kosztu innych faz systemu informatycznego. Powinno ono być integralnym składnikiem procesu kontroli jakości oprogramowania. Wydawać by się mogło, że naturalne jest, aby tak ważny i kosztowny etap realizacji projektu był przeprowadzany według sprawdzonych reguł sztuki inżynierskiej. Co więcej, koszty usuwania błędów w późniejszych fazach projektowych rosną lawinowo. Głównym sposobem eliminacji błędów powinno być zatem zapobieganie im już we wczesnych fazach realizacji systemu informatycznego z przekonaniem, że wysoka jakość wymagań czy projektu bardzo pomaga w wytworzeniu kodu o podobnej jakości. *Niezrozumienie istoty testów polega na sprowadzenie ich roli do generalnego, uniwersalnego i najczęściej jedynego narzędzia kontroli jakości systemu informatycznego.* Problem jednak tkwi w tym, że celem procesu testowania jest znalezienie błędów, a nie zapewnienie jakości. Osoba poszukująca błędów w programie nie może zapewnić wymaganej funkcjonalności systemu. To powinna gwarantować analiza wymagań, projektowanie i kodowanie.

Testowanie nie zapewnia również efektywności przetwarzania i wykorzystania zasobów. Testy pozwalają ją jedynie zmierzyć, ale to dopiero odpowiednio dobrany algorytm i jego realizacja prowadzą do osiągnięcia wymaganych parametrów. Testowanie nie zapewnia również niezawodności, ale pozwala ocenić jej poziom pośrednio przez oszacowanie liczby błędów i rozkład ich ilości w czasie.

Testowanie ma dobrze określony cel, wynikający z natury procesu wytwarzania oprogramowania, a techniki, których używa są wyspecjalizowane pod kątem realizacji tego celu. Dlatego proces testowania musi być uzupełniony innymi działaniami kontrolno – weryfikacyjnymi tak by system różnorodnych oddziaływań na jakość systemu informatycznego był pełny, tzn. ograniczał propagację błędów z wcześniejszych faz projektowych do późniejszych. *Z tych względów, często testowanie utożsamia się z pojęciem kontroli jakości w ogóle obejmując tym procesem także recenzje i przeglądy dokumentacji, modeli wymagań, projektu, itd.*.

Trudno, zatem zbudować pełny zakres lub strukturę testów, które wykryją wady oprogramowania. Usuwanie wad i korekty błędów mogą pochłaniać nawet połowę czasu pracy projektanta, dlatego każde efektywne udoskonalenie metodologii testów lub jej zautomatyzowanie pozwalające zmniejszyć nakłady czasu zużywanego na testy ma ważny wpływ na czas projektowania i koszt cyklu prac. Należy pamiętać o tym wtedy, gdy będzie planowana możliwość wsparcia prac zespołów testów przez różnego rodzaju narzędzia. Opłacalność każdej inwestycji w usprawnianie i doskonalenie testowania jest szczególne widoczna w wypadku testów regresji, które dotyczą funkcjonalności już przetestowanej, a która mogła ulec degradacji na skutek efektów ubocznych zmian np. w innym module programu. Potrzeba takich testów jest skutkiem natury błędów oprogramowania, które ujawniają się w dynamice realizacji funkcji wykonywanych w kontekście danych wejściowych. Przeprowadzanie testów regresji bez wsparcia narzędziowego oznacza wykonywanie po setki razy tych samych testów.

Podstawą do wykonywana testów są kryteria testowe opracowywane pod kątem kryteriów jakości istotnych dla klienta. Systemy informatyczne są zatem testowane dla sprawdzenia, czy nie ma błędów w realizacji jego funkcji, czy jest bezpieczny lub przyjazny dla użytkownika. Najogólniej dzielimy testowanie na funkcjonalne (black – box – testing), podczas którego szukane są błędy i niezgodności ze specyfikacją w jakiejś wydzielonej i zamkniętej części programu i tzw. white – box – testing, kiedy sprawdzane jest, czy procedura lub moduł znajdują się zawsze w oczekiwanym stanie, czy są błędy w algorytmach realizacji funkcji, czy wszystkie możliwe ścieżki programowe zostały wykonane (zakłada się przy tym typie testowania, że znana jest logika modułów i procedur).

Zależnie od stanu zaawansowania prac nad modułami systemu testowanie przechodzi przez różne, przedstawione w tabeli 2 fazy.

Tab. 2. Podstawowe fazy procesu testowania

Faza testów	Cel fazy testów
Testowanie jednostkowe	Sprawdzane są ścieżki i błędy wewnątrz modułu. Jest ono ograniczone do zakresu modułu, dzięki czemu może być prowadzone równolegle dla różnych

	modułów. Zwykle wymaga stworzenia dodatkowego środowiska testowego i specjalnych programów testowych służących do wywoływania sprawdzanych modułów z odpowiednimi parametrami
Testowanie integracyjne	Kontroluje poprawności interfejsów między modułami. Jest ono systematyczną techniką składania oprogramowania z modułów w większe fragmenty (podsystemy) z jednoczesnym testowaniem interfejsów, gdzie poszczególne moduły nie operują na danych testowych, ale na rzeczywistych danych produkowanych przez współpracujące moduły
Testowanie walidacyjne	Dotyczy uruchomionego programu, który teoretycznie powinien być pozbawiony już błędów implementacyjnych i integracyjnych. Walidacja polega na sprawdzeniu, czy program funkcjonuje w sposób oczekiwany i opisany w specyfikacji wymagań, a szczególnie w rozdziale o kryteriach akceptacji. Jest to ostatni krok przed przekazaniem oprogramowania użytkownikowi, będący jakby testami akceptacyjnymi u producenta.
Testowanie systemowe Beta-testy	Testowanie systemowe obejmuje testy w środowisku użytkownika przeprowadzane przez producenta lub w postaci beta-testów przez samego użytkownika (np. dystrybutorów, partnerów, itp.) w jego systemie i na jego rzeczywistych danych. Nazywane jest również testowaniem akceptacyjnym.

Źródło: na podstawie [5]

Duże znaczenie ma psychologiczny dobór odpowiedniego zespołu testującego. Osoba testująca, która jest autorem programu nigdy nie będzie obiektywna. Dlatego testowanie powinna przeprowadzać inna, niezależna osoba, której zadaniem jest bez „sentymentów" znaleźć błąd.

Testy powinny być zaplanowane (przykładowy szablon planu testów zawiera tabela 3) co wyznacza kolejność: najpierw strategiczne planowanie testów formułujące koncepcję testów najbardziej adekwatną do specyfiki projektowanego systemu, potem operacyjne organizujące pracę, produkty testowe i fazy testów i w końcu taktyczne, a więc procedury i przypadki testowe oraz nadzorowanie wykonania testów. Ostatnim elementem który łączy wszystkie poprzednie jest *repozytorium testowe*. Stanowi ono swego rodzaju bazę wszelkich informacji dotyczących testów w projekcie. Obejmuje wszelkie niezbędne dokumenty, z których łatwo mogą korzystać testujący, gotowe do wielokrotnego wykorzystania

procedury testowe i zestawy danych testowych. Jest to także najwłaściwsze miejsce dla rejestru błędów z historią ich obsługi, a także statystyk i charakterystyk niezawodnościowych systemu, podsystemów, modułów, itd. Jest to swoista hurtownia danych testowych, pozwalająca na różnego rodzaju wielowymiarowe analizy stanu produktu, przewidywanego poziomu niezawodności, itd.

Tab. 3. Przykładowy szablon planu testów

Plan testów	
Wprowadzenie	Opis celu i zakresu testów, ogólnej koncepcji testów i zastosowanego podejścia.
Architektura systemu	Opis architektury systemu w zakresie niezbędnym do zrozumienia potrzeby i zakresu testów
Środowisko testowe	Opis konfiguracji sprzętowo – programowej, w której będą uruchamiane testy. Środowisko testowe może się zmieniać w zależności od zakresu testów, fazy projektowej i kryteriów, które w danym momencie podlegają badaniom
Kryteria testowe	Opis warunków spełnienia wymagań przez oprogramowanie. Zwykle kryteria odnoszą się do podstawowych kategorii jakościowych tzn.: funkcjonalności, użyteczności, efektywności, bezpieczeństwa, przenośności i niezawodności
Techniki testów	Opis technik testów, które będą wykorzystywane w testach oprogramowania, np. technika logicznego pokrycia grafu wywołań
Produkty testowe	Opis produktów projektowych, które będą powstawać w wyniku przygotowania do procesów testowania (plan testów, projekty testów), w jego trakcie (zestawy danych testowych, wyniki testów, rejestr błędów) i po zakończeniu (wyniki testów, zalecenia, dokumentacja testowa, statystyki)
Organizacja procesu testowania	Opis struktury, ról, podziału odpowiedzialności i zasobów niezbędnych do przeprowadzenia testów
Harmonogram testów	Opis planu poszczególnych działań związanych z testami oprogramowania rozłożonych na osi czasowej
Repozytorium testów	Opis bazy danych testowych zawierający specyfikację produktów przechowywanych w bazie, szablony dokumentów, procedury gromadzenia danych testowych, wyników testów, narzędzie i procedury rejestracji błędów i uzgadniania przekazywania ich do poprawek, itp.

Źródło: na podstawie [5]

Są różne sposoby wykonywania testów, ale trzeba o nich myśleć i je planować już na etapie analizy wymagań. *Oprogramowane powinno mieć modularną strukturę pozwalającą na łatwe testowanie. Musi istnieć dokumentacja*

projektowa, specyfikacje wymagań, interfejsów, kryteria efektywności, bezpieczeństwa, itp. Bez tego nie da się przygotować skutecznych procedur testowych, a samo oprogramowanie także straci na niezawodności.

4.3. Dokumentacja w zapewnieniu jakości w projektach wdrożeniowych.

W procesie realizacji projektów informatycznych wykorzystywane są różne metodyki, z których każda posiada swoje standardy dokumentacyjne. Jednym z elementów zapewniających odpowiednią jakość wdrożenia systemu jest oprócz dokumentacji systemu jest opracowanie schematu dokumentów, które powstają w różnych etapach realizacji procesu wdrożenia, np. w metodyce IBM wykorzystywana jest książka kontrolna projektu o jednorodnej wersji i standardowych formularzach.

Odpowiedni system dokumentujący gwarantuje kompletność i jednoznaczność, zapewnia sprawną komunikację między klientem a zespołem wdrożeniowym oraz wewnątrz zespołu, jak również daje możliwość prowadzenia kilku wersji. Jednolity schemat dokumentacji umożliwia zapisanie podjętych decyzji, ich uzasadnienie oraz stanowi punkt odniesienia dla oceny realizacji prac. Należy zauważyć, że tworzenie dokumentów nie może stać się głównym celem projektu, a ma być narzędziem wspomagającym proces jego zarządzania.

Schemat dokumentacji powinien obejmować wszystkie grupy dokumentów związane ze wszystkimi fazami procesu tj: dostawą i instalacją oprogramowania oraz głównymi i dodatkowymi usługami wdrożeniowymi. W celu uniknięcia nieporozumień i dla prawidłowego przebiegu procesu należy opracować ujednolicony schemat korespondencji z klientem.

Wśród dokumentów dotyczących dostawy oprogramowania jako najistotniejsze należy uznać dokumenty opisujące przedmiot i istotne warunki dostawy oprogramowania oraz protokół dostarczenia produktu informatycznego.

Głównym etapem projektu wdrożeniowego etapem jest wykonanie usług wdrożeniowych. Jest to etap, który wymaga szeregu bardzo istotnych dokumentów wśród których należy wymienić:

- dokument definiujący projektu,
- projekt rozwiązania,
- potwierdzenie dostarczenia i instalacji niezbędnego oprogramowania,
- specyfikację ewentualnych modyfikacji,
- protokoły odbioru usług, zatwierdzenia punktów kontrolnych i zakończenia.

Dokument definiujący projekt to jeden z głównych dokumentów określający cel główny i opis projektu, przedstawiający organizację projektu, metody śledzenie i kontrola projektu, specyfikację prac technicznych oraz zasady dokumentacji.

Zawarty w dokumencie definicji opis projektu to przede wszystkim określenie najważniejszych celów projektu, zakres funkcjonalny wdrożenia, podział wdrożenia na etapy oraz budżet. Jest to również dokument zawierający ramowy plan projektu, jego harmonogram, plan kamieni milowych jak również pożądane efekty poszczególnych etapów.

Nie mniej istotne są dokumenty dotyczące organizacji projektu. Dokumenty

te powinny przedstawić komitet sterujący: sponsora projektu, przedstawicieli klienta, przedstawicieli firmy wdrożeniowej, pozostałe osoby wchodzące w skład komitetu oraz zespoły wdrożeniowe (kierownicy, członkowie, konsultanci wiodący, pozostali konsultanci).

Do części organizacyjnej należą również dokumenty dotyczące śledzenia i kontroli projektu, które powinny zawierać podstawowe informacje dotyczące zarządzania projektem takie jak:

- terminarz okresowych spotkań komitetu sterującego,
- sposób koordynacji wewnętrznej w projekcie,
- metody i hierarchię komunikacji,
- sposób zatwierdzenie punktów kontrolnych i kamieni milowych,
- metody rozwiązywania konfliktów.

Śledzenie kontrola projektu powinny między innymi określać formę wszystkich dokumentów takich jak raporty ze spotkań, agendy, notatki spotkań. Dla dokumentów tych należy określić szablony zawierające przynajmniej następujące informacje:

- data,
- wersja,
- czas trwania,
- uczestnicy,
- cel,
- omówione zagadnienia i podjęte decyzje,
- termin następnego spotkania,
- podpisy ze strony firmy wdrożeniowej i klienta,
- oraz informacje o adresatach dokumentu.

Kolejna istotna grupa są dokumenty dotyczące projektu rozwiązania. Dokumenty te powinny dla każdego wydzielonego fragmentu systemu np. procesu przedstawiać:

- nazwę procesu lub usługi,
- cel,
- opis,
- scenariusz działań,
- mapę procesu, procesy poprzedzające, procesy korzystające,
- dane podstawowe dla obsługi procesu,
- raporty wykorzystywane w obsłudze procesu (standardowe lub dodatkowe),
- uprawnienia do obsługi procesu,
- zmiany organizacyjne niezbędne do implementacji procesu,
- wymagane modyfikacje i interfejsy z systemami zewnętrznymi,
- metody migracji danych.

W przypadku realizacji modyfikacji oprogramowania należy również wprowadzać szczegółowa dokumentację określającą projekt funkcjonalny zawierający specyfikację funkcjonalną (cel zakres, ograniczenia, załączniki) oraz uzgodnienia z klientem dotyczące wyceny i zatwierdzenia zakresu.

Tworząc rozwiązanie informatyczne należy zawsze mieć na uwadze cel

projektu, dokumentacja jest tylko jednym z narzędzi, wspomagających jego osiągnięcie i zapewniających określony poziom jakości. Należy również zauważyć, że jeśli firma wdrażająca system lub jej klient posiada certyfikaty jakości np. ISO, wtedy znaczna część dokumentacji wynika z wymaganych przez nie zapisów.

Wnioski

Z punktu widzenia technologii, wszystkie firmy mają dzisiaj praktycznie ten sam poziom wyposażenia sprzętowego, wydajność systemów projektowania oprogramowania jest porównywalna, a projektanci i programiści korzystają z tych samych narzędzi dostarczanych przez kilkadziesiąt wiodących na rynku firm. To co może różnicować firmy to czynnik ludzki, kultura organizacyjna, efektywny system procesów produkcyjnych oraz wdrożony i skuteczny proces ciągłego doskonalenia.

Zarządzanie jakością nie polega na korzystaniu ze specjalnych narzędzi, przeprowadzaniu przełomów organizacyjnych i zdobywaniu certyfikatów jakości. Jego istotą jest zmiana kultury organizacyjnej firmy. Zamiast koncentrować się na zysku - nastawiamy się na jakość produktów. Nawet, jeśli miałoby nas to więcej kosztować teraz, przyniesie większe zyski później. Aby to podejście zrozumieć i mieć odwagę wdrożyć w firmie potrzeba dojrzeć do pełnego i dorosłego widzenia korzyści, jakie niesie ono dla firmy w perspektywie strategicznej, a nie tylko krótkoterminowej. Jest to skomplikowane, ale realne.

Jakość powstaje nie w wyniku badania gotowego produktu - testów, ale w wyniku wbudowania jej w produkt na każdym etapie procesu wytwórczego. W sytuacji idealnej testowanie w ogóle przestaje być potrzebne – gwarantem jakości jest jakość systemu wytwórczego. Zmiana projakościowa systemu wytwórczego nie następuje jako wynik uporządkowania pojęć związanych z zarządzaniem jakością. Nie ogranicza się do ukonstytuowania pewnych reguł, zasad, systemów, szablonów. Jest procesem kształtowania kultury firmy. Każda firma przechodzi pewną ewolucję, dojrzewa do rozumienia nie tylko procesów rynkowych i wytwórczych, ale także tych, które zmierzają do zapewnienia właściwej jakości dostarczanych produktów. Skuteczne zarządzanie jakością, które ma prowadzić do wypracowania przewagi konkurencyjnej na rynku nie może obejść się bez przemyślanego określenia polityki jakości, zarządzania zasobami ludzkimi, środowiskiem pracy, wymaganiami klienta.. Jest to długotrwały proces, którego pozytywne skutki widoczne są dopiero po pewnym czasie i może dlatego tak trudno zdecydować się nań tym, którzy oczekują natychmiastowych wyników.

Analiza wymagań i testowanie są bardo ważnymi i kosztownymi etapami w projekcie powstawania czy wdrażania systemu informatycznego, dlatego też należy do nich podejść ze szczególną, inżynierską pieczołowitością i zastosować najefektywniejsze narzędzia do ich realizacji.

Literatura

1. Byzia T.: artykuł Zarządzanie jakością oprogramowania. Część 1: Jakość. Computer World nr 8, 1998.
2. Byzia T.: artykuł Zarządzanie jakością oprogramowania. Część 2: Terminologia. Computer World nr 9, 1998.
3. Byzia T.: artykuł Zarządzanie jakością oprogramowania. Część 3: Filozofia. Computer World nr 10, 1998.
4. Byzia T.: artykuł Zarządzanie jakością oprogramowania. Część 5: Kryteria. Computer World nr 12, 1998.
5. Byzia T.: artykuł Zarządzanie jakością oprogramowania. Część 6: Testowanie. Computer World nr 13, 1998.
6. Byzia T.: artykuł Zarządzanie jakością oprogramowania. Część 7: Narzędzia. Computer World nr 14, 1998.
7. Byzia T: Zarządzanie jakością w projekcie informatycznym. Materiały z warsztatów po XI Konferencji Górskiej Szkoły PTI „Jakość systemów informatycznych problemy i rozwiązania", Szczyrk, 1999.
8. Norma IEEE Std 730.1-1995. Wytyczne do tworzenia planu zapewnienia jakości.
9. Szyjewski Z.: Zarządzanie projektami informatycznymi. Metodyka tworzenia systemów informatycznych. Agencja wydawnicza Placet, Warszawa, 2001.

ROZDZIAŁ VII

ZARZĄDZANIE PROJEKTEM „MOVEX-ISO-FAMEG"

Krzysztof JASIOROWSKI, Jerzy Stanisław NOWAK, Remigiusz JASIOROWSKI

Wstęp

W artykule przedstawiono zarządzanie dużym projektem – wdrażanie zintegrowanego systemu informatycznego (ZSI) klasy MRPII/ERP oraz systemu zarządzania jakością (SZJ). ZSI to system „Movex" na platformie sprzętowej IBM AS/400 – produkt szwedzkiej firmy Intentia – dla 100 stanowisk; najpierw w wersji 9A (1999r.), później w wersji 11-tej (2003r.). Integralną częścią projektu Movex była metoda wdrożeniowa „Implex", w decydującym stopniu gwarantująca szybkie wdrożenie i zapewniająca możliwość dalszego rozwoju. Wdrożono 25 standardowych (spośród 42 oferowanych z wersji 9A).

Przed rozpoczęciem wdrażania projektu przyjęto dwa główne założenia:

- **jeżeli zastosowanie technik MRPII w sferze produkcji będzie nieskuteczne, wówczas wdrożenie kończy się niepowodzeniem,**
- **równolegle będzie również wdrażany SZJ – wówczas zgodny z PN-ISO 9001:1996 (EN-ISO 9001:1994) przez ten sam zespół wdrożeniowy; zadecydowały o tym dwa czynniki: procesowe podejście do wdrażania obydwóch systemów oraz maksymalne wykorzystanie wiedzy i umiejętności zakładowej kadry fachowców [10].**

Przedmiotowy projekt został wdrożony w Zakładach Mebli Giętych „Fameg" w Radomsku, jednym z największych w Europie producentów mebli giętych z drewna litego (głównie krzeseł, foteli, taboretów, stolików, wieszaków, itp.) – zaliczanych do grupy mebli szkieletowych. Wówczas asortyment produkcji obejmował 200 modeli podstawowych rocznie i około 1800 ich wariantów (gatunki drewna i ich wybarwienia, układy tapicerskie wraz z rodzajem i kolorystyką tkanin, sposoby montażu i demontażu dla celów transportowych). W roku 2001 wartość sprzedaży wynosiła około 130 mln zł rocznie przy zatrudnieniu 2500 pracownikach w trzech zakładach produkcyjnych zlokalizowanych w Radomsku [10].

Firma ta za skuteczne wyniki wdrożeń przedmiotowego projektu została dwukrotnie wyróżniona w finałach konkursów Computerworld – Lider Informatyki 1999 i Lider Informatyki 2003. Pierwsze wdrożenie Movexa (wersja 9A) trwało zgodnie z założeniami, 18-cie miesięcy; wdrożenie wersji11-tej zaangażowało 21 dniówek konsultacyjnych (dostawcy) i trwało jeden miesiąc.

Istotnym elementem organizacyjnym, sprzyjającym sprawnemu prowadzeniu projektu, było istnienie w firmie nabytego przez pracowników dość wysokiego poziomu kultury informatycznej w wyniku funkcjonowania

wieloletnich i wielorakich zastosowań informatyki, począwszy od 1965 roku na platformach: ZAM-2 Beta, Odra 1003, ZAM-41, ODRA 1304 i 1305, MERA 9150 (zapis danych źródłowych bezpośrednio na taśmach magnetycznych) oraz komputery personalne tzw. PC-ty, pracujące w sieciach Arcnet i Ethernet (wcześniej OA-LINK).

W 1995 roku funkcjonowało (na PC-tach) w firmie 19 różnych systemów, podsystemów i programów o bardzo niskim stopniu ich integracji [4]. Zasadniczym problemem (w oczekiwaniach kadry zarządzającej) był brak szybkiego i dosyć szczegółowego przenoszenia danych ze zbiorów „produkcyjnych" do zbiorów „ekonomiczno – księgowych". Fakt ten w zasadniczym stopniu ograniczał możliwości stosowania elementów rachunkowości zarządczej w sprawnym i efektywnym zarządzaniu firmą w zaostrzających się warunkach konkurencyjności na rynkach zagranicznych i krajowych.

1.Działania przedkontraktowe

1.1.Wybór systemu

W 1995 roku ogłoszony został przetarg na dostawę i wdrożenie zintegrowanego systemu informatycznego klasy MRPII. Przetarg ten nie został rozstrzygnięty; na dostawę i wdrożenie zintegrowanego systemu informatycznego klasy MRPII. Przetarg ten nie został rozstrzygnięty; na fakt ten miało wpływ stanowisko Funduszu Inwestycyjnego Nr 4, w którym „Fameg" był „okrętem flagowym". Niemniej jednak wyniki tego przetargu pozwoliły na przeprowadzenie przeglądu dostępnych w Polsce systemów klasy MRPII, które by spełniały oczekiwania Famegu. Wówczas sporządzono listę dostawców systemów zintegrowanych (również firm, tzw. „integratorów"), na której znalazło się 10 dostawców krajowych i zagranicznych akredytowanych w Polsce. Z tej liczby 7-miu dostawców oferowało produkty na platformie „unix-owej", zaś 3-ch na platformie IBM AS/400.

W 1997 roku ogłoszono drugi przetarg, do którego zaproszono trzy firmy z przedmiotowej listy. Wybrano system MOVEX (patrz pkt 1, 2.1).

1.2.Szkolenia

Od 1995 roku prowadzono szereg szkoleń zewnętrznych i wewnętrznych dla grup pracowników zatrudnionych na różnych poziomach zarządzania. Problematyka szkoleń obejmowała:

- zarządzanie przedsiębiorstwem,
- metody usprawniania zarządzania firmą,
- prezentacje funkcjonowania zintegrowanych systemów informatycznych klasy MRPII (u użytkowników eksploatujących systemy tej klasy),
- rachunkowość zarządcza, zarządzanie finansami, marketing,
- publikacje w przedmiotowych zakresach.

W szkoleniach tych uczestniczyli: członkowie zarządu, główni specjaliści, kierownicy działów zarządu, kierownicy i mistrzowie wydziałów produkcyjnych i pomocniczych. Szkolenia tego typu kontynuowano również w trakcie procesu wdrażania systemu zintegrowanego.

Podczas przedmiotowych szkoleń wszyscy ich uczestnicy nabierali coraz większego przekonania, że wdrożenie systemu klasy MRPII będzie wymagało określonych zmian organizacyjnych, zmniejszenia zatrudnienia w poszczególnych komórkach zarządu oraz zmiany struktury zatrudnienia w obszarze obsługi systemu informatycznego.

W zasadzie każdy uczestnik szkoleń zadawał sobie pytanie: jakie będzie moje miejsce w projekcie i jaką pracę będę w firmie wykonywał?

1.3.Konflikty organizacyjne we wdrażaniu systemu informatycznego

Najwyższe kierownictwo firmy miało pełną świadomość, że wdrażanie tak dużego projektu może powodować konflikty pomiędzy pracownikami.

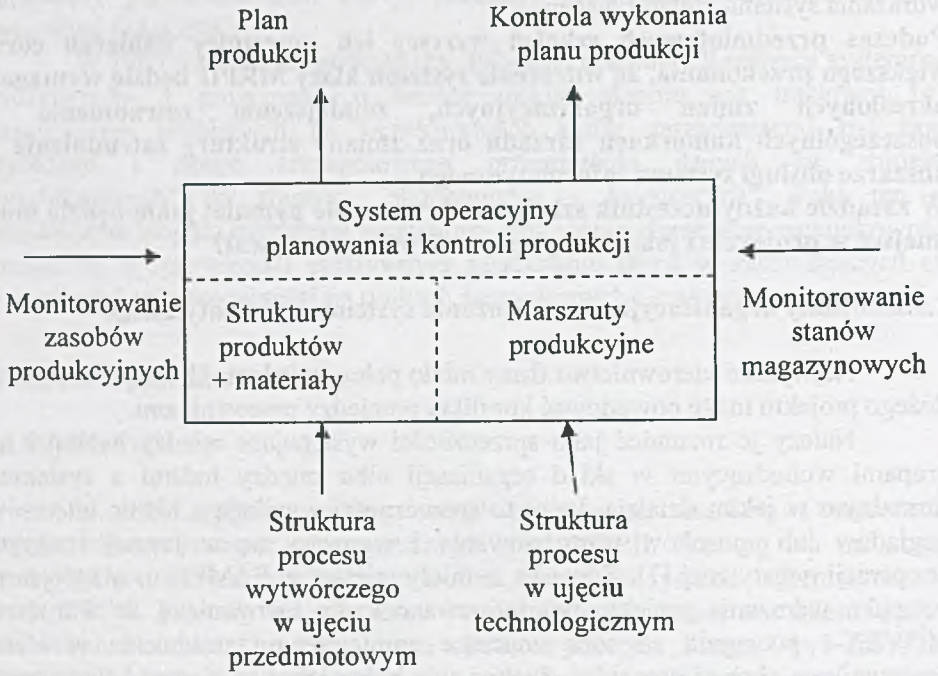
Należy je rozumieć jako sprzeczności występujące między ludźmi i ich grupami wchodzącymi w skład organizacji albo między ludźmi a systemem formalnym w jakim działają, które to sprzeczności wynikają z różnic interesów, poglądów lub sposobów wartościowania i wyrażają się w jawnej i ukrytej kooperacji negatywnej [1]. Zjawiska te miały miejsce w FAMEG-u, w którym na początku wdrażania projektu poinformowano kadrę kierowniczą, że wdrożenie MOVEX-a pociągnie za sobą znaczące zmniejszenie zatrudnienia w sferze pracowników obsługi procesów. Zachowania były różne: od postaw negatywnych do pozytywnej rywalizacji, co można było wcześniej przewidzieć. Natomiast w pewnym momencie pojawiło się niebezpieczne zjawisko w zespole wdrożeniowym; w gronie gospodarzy procesów zaczęto się wyraźnie „zastanawiać, który proces jest najważniejszy?". Były próby wpłynięcia na konfigurację systemu pod kątem „ustawienia się na przyszłość". W tej sytuacji, przyjęta wcześniej, struktura organizacji wdrożenia okazała się bardzo skuteczna.

1.4. Istota systemów MRPII/ERP i SZJ

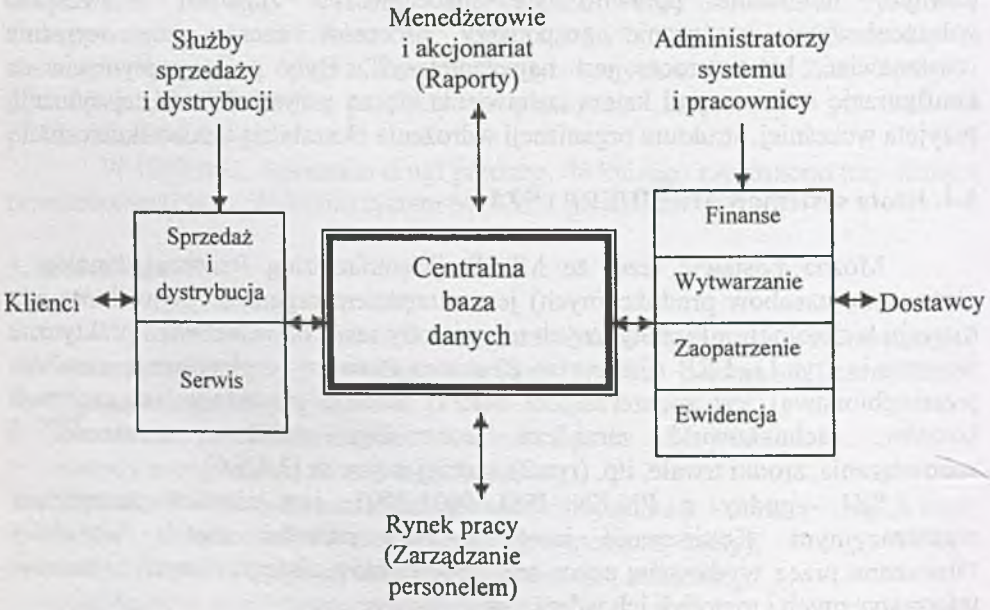
Można postawić tezę, że MRPII (Manufacturing Resorce Planning – planowanie zasobów produkcyjnych) jest narzędziem organizacyjnym, które bez rozwoju technologii informatycznych nie miałoby szans na wdrożenie i praktyczne stosowanie (rys.1). ERP (Enterprise Resource Planning – planowanie zasobów przedsiębiorstwa) jest rozszerzeniem MRPII o takie procedury jak: rachunek kosztów, rachunkowość zarządcza, kontroling, cashflow, należności i zobowiązania, środki trwałe, itp. (rys.2); szerzej o tym w [3,4,5,6].

SZJ, zgodny z PN-EN ISO 9001:2001, jest również narzędziem organizacyjnym. Konieczność integracji tych narzędzi została wcześniej zauważona przez wydawców norm oraz producentów zintegrowanych systemów informatycznych i metodyk ich wdrażania.

Najistotniejszą cechą tych narzędzi organizacyjnych, w przedmiocie omawianego tutaj zagadnienia jest procesowe podejście przy wdrażaniu systemów ERP i SZJ; szerzej o tym w [5,6].



Rys. 1. Klasyczny model operacyjnego planowania i kontroli produkcji np. MRPII [7].



Rys. 2. Schemat budowy systemu ERP- oprac. na podstawie [8].

2. Fazy wdrażania projektu

Przyjęto założenie, że wdrożenie projektu będzie obejmowało trzy fazy:

A. Przygotowanie
B. Realizacja procesu wdrożeniowego
C. Ustanowienie mechanizmu dalszego rozwoju systemu w powiązaniu z permanentnym usprawnianiem procesów gospodarczych [6].

2.1. Faza A – przygotowanie

Faza ta obejmowała okres od powstania pomysłu zastosowania systemu ERP do momentu rozpoczęcia procesu wdrożeniowego. Faza ta miała chyba najbardziej istotne znaczenie w całym przedsięwzięciu, chociaż nie zawsze doceniana przez wszystkich decydentów. W tej fazie bardzo łatwo popełnić brzemienne w skutkach błędy w niedoszacowaniu nakładów i przeszacowaniu oczekiwanych efektów. Dlatego też faza ta w Famegu rozpoczęła się odpowiednio wcześniej – od początku 1995 roku. Przygotowanie wdrożenia starano się przeprowadzić jak najsolidniej wykorzystując wszelkie dostępne wówczas środki: konsultacje, szkolenia dość szerokiej własnej kadry kierowniczej na różnych poziomach, prezentacje oferowanych w Polsce systemów, doświadczenia użytkowników systemów tej klasy, publikacje w menadżerskich i informatycznych wydawnictwach. System MOVEX, oferowany przez firmę INTENTIA Polska, wybrano w ograniczonym przetargu, do którego zaproszono, oprócz wymienionej, firmy oferujące również oprogramowanie na wybranej przez FAMEG platformie sprzętowej IBM AS/400: ISA (system BPCS) oraz JBA (system Biznes 400).
Faza A kończy się z chwilą podpisania kontraktu. Kontrakt zawarto w drugiej połowie 1997 roku z warszawską firmą VIMEX Spółka z o.o. (negocjującą wówczas warunki przekształceń kapitałowych i własnościowych z firmą INTENTIA Szwecja); obecnie INTENTIA Polska. Ze względu na wybraną platformę sprzętową, uzgodnienia przedkontraktowe prowadzone były przy współudziale przedstawicieli firmy IBM. Oprócz dostawy oprogramowania i sprzętu kontrakt obejmował:

- metodę wdrożeniową IMPLEX,
- opcję DEVEX gwarantującą bezpłatne dostarczenie przez dostawcę nowej – wybranej przez Fameg – dostępnej wersji oprogramowania w okresie 60 miesięcy, liczonego od chwili wdrożenia systemu,
- wykonanie okablowania strukturalnego, we wszystkich obiektach Famegu, niezbędnego do funkcjonowania wdrażanego systemu.

W tej fazie funkcjonował, powołany przez zarząd, zespół przygotowawczo-wdrożeniowy, którym kierował dyrektor ekonomiczny a w skład wchodzili specjaliści: kierownik działu informatyki, główny technolog, szef logistyki, szef produkcji, szef marketingu. Ustalono wówczas, że niezbędne będą

zmiany o charakterze organizacyjnym. Należało przemodelować strukturę w kierunku odejścia od pionowo ukształtowanych zależności z wyraźnie wyodrębnionymi działami funkcjonalnymi i wykonywanymi przez nie czynnościami, kierując się ku procesom, które przebiegają w układzie poziomym w poprzek organizacji. Już wówczas było wiadomo, że zarządzanie firmą zmierzać będzie w kierunku podejścia procesowego (zarządzanie przez pryzmat zachodzących procesów w firmie). Na takim bowiem podejściu bazuje wdrożenie, a później funkcjonowanie zintegrowanego systemu informatycznego. Fakt ten miał odzwierciedlenie w organizacji wdrożenia.

Informacje uzupełniające zawarte są w pkt. 1 i 2 niniejszego artykułu.

2.2. Faza B – realizacja procesu wdrożeniowego

Obejmowała ona okres od rozpoczęcia realizacji kontraktu do protokółu zamykającego wdrożenie i rozpoczynającego pracę na nowym systemie. Jest to faza , w której najistotniejsza jest współpraca między ludźmi z dwóch różnych firm o odmiennej kulturze [4]. Projekt "Movex" był też szczególnym spotkaniem ludzi, którzy choć na co dzień pracują w jednym przedsiębiorstwie, to jednak w dużej części przypadków nic nie robili razem. Ich wsparciem z jednej strony był przedstawiciel najwyższego kierownictwa, z drugiej – konsultanci, którzy byli źródłem wiedzy merytorycznej o nowym systemie. Zespół wdrożeniowy pracował zatem w środowisku nowym dla każdego uczestnika. Stąd przypadała nowa rola dla zarządzającego zespołem [4].

Współpraca z konsultantami zewnętrznymi nie była usłana różami. Trzeba sobie powiedzieć, że związanie się kontraktem wdrożeniowym z firmą i z jej produktem to jak małżeństwo z rozsądku z rozdzielnością majątkową, lepiej uważać na koszty. Możemy się nie kochać ale najważniejszy jest wytyczony cel – wdrożenie, dojść do niego pomaga kompromis [4].

Faza B została podzielona na etapy:

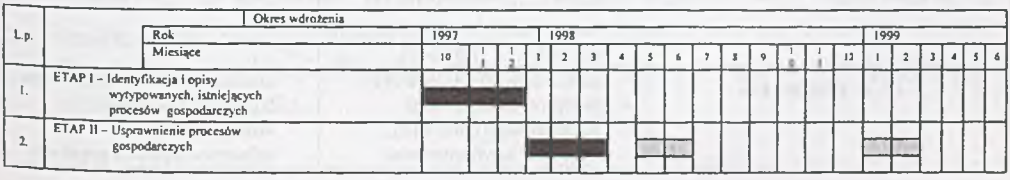
- **ETAP I** – wyodrębnienie i zdefiniowanie procesów oraz wyspecyfikowanie tych, które wymagały usprawnień. W realizacji tego zadania zastosowano metodę "burzy mózgów" w dwóch oddzielnie pracujących zespołach. Wyniki ich pracy zostały skonfrontowane w gronie połączonych zespołów i przy zastosowaniu konsensusu w efekcie powstała pierwsza wersja zdefiniowanych procesów: produkcyjnych (wykonawczych) i nieprodukcyjnych (częściej nazywanych biznesowymi lub gospodarczymi). W tym etapie nastąpiło zorganizowanie komitetu sterującego i zespołu wdrożeniowego oraz seria dodatkowych szkoleń dla określonych grup pracowników. Przed zakończeniem tego etapu przydzielono konsultantów do zespołu wdrożeniowego konsultantów zewnętrznych. Podpisano również umowę z audytorem zewnętrznym projektu. Zdefiniowane procesy podzielono na podprocesy, podprocesy na czynności, czynności na kroki. Zdefiniowano szczegółowo cele całego projektu, przeprowadzono analizę środowiska biznesowego, ustalono struktury wspierające poszczególne procesy (gospodarze procesów + zakładowi specjaliści), opracowano specyfikacje wykorzystania

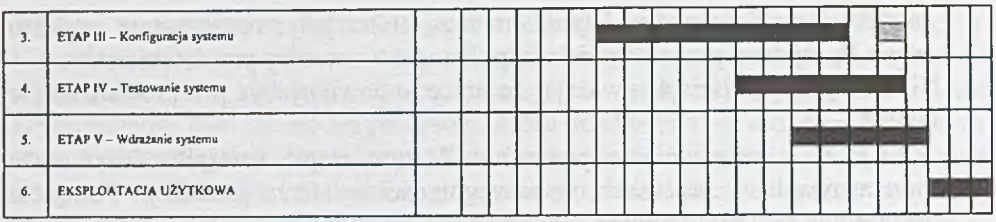
przydzielonych zasobów. Wynikiem tego etapu były: opracowanie projektu etapu II, raport o procesach i ich mapy, model biznesowy przedsiębiorstwa. Niektórzy specjaliści [4] uważają, że prace te powinny być przeprowadzane w fazie A.

- **ETAP II** – usprawnianie procesów. W tym etapie nastąpiło dosyć duże przetasowanie w procesach, w szczególności w sferze produkcji. Podejście procesowe stanowiło novum, podobnie jak cała koncepcja restrukturyzacji. Od gospodarzy procesów wymagano większej wiedzy niż ta, którą początkowo dysponowali; konieczne były dalsze gruntowne szkolenia. Wynikiem tego etapu były: uaktualniona definicja projektu, plan etapu III, projekt techniczny instalacji komputerowej, przebiegu i opisy usprawnionych procesów.
- **ETAP III** – konfiguracja systemu; na mapę procesów nałożono funkcje systemu MOVEX (programów standardowych). Rozpoczęto zapełnianie produkcyjnej bazy danych (PBD) oraz zainstalowano moduły jakości QA. Wynikiem tego etapu były: model rozwiązania i jego opis, opracowanie podręcznika dla administratora w zakresie NAWIGACJI systemu, zaktualizowany raport o procesach, wstępne instrukcje pracy dla przyszłych użytkowników, opracowanie projektu etapu IV.
- **ETAP IV** – testowanie systemu na częściowo zapełnionej PBD i kontynuacja dalszego jej zapełniania, ustalanie automatycznych księgowań oraz "wygładzanie konfiguracji". Rozpoczęto szkolenia użytkowników przez gospodarzy procesów w oparciu o docelowe instrukcje, przeprowadzono testy: pilotażowy i pełnoskalowy. Wynikiem tego etapu były: zaktualizowany opis konfiguracji, uzupełnienie i aktualizacja wszystkich instrukcji, plan etapu V.
- **ETAP V** – wdrożenie i rozpoczęcie eksploatacji użytkowej systemu. Najwrażliwszym elementem wdrożenia było opanowanie modułów MRP II. Wcześniej kadra kierownicza nie miała doświadczenia w stosowaniu tych mechanizmów w postaci narzędzi informatycznych. Eksploatacja użytkowa odbywała się już na zapełnionej PBD. Przejście firmy na nowy system informatyczny nastąpiło z pominięciem pracy równoległej na starym oprogramowaniu i nowym systemie. O braku etapu przejściowego zadecydowały rozmiary firmy [6,10]. Wynikiem tego etapu były: wdrożone i zatwierdzone rozwiązanie, wstępne zarysowanie mechanizmów dalszego usprawniania procesów i systemu.

Powyższy podział fazy B na etapy dokonany został w oparciu o metodę wdrożeniową IMPLEX (tablica 1). Podział na etapy jest też częściowo zdeterminowany wielkością projektu oraz specyfiką firmy.

Harmonogram etapowego wdrażania systemu MOVEX w Fameg S.A. przedstawia rys.3.





LEGENDA
- prace podstawowe
- prace uzupełniające

Rys. 3. Źródło [6]

Tablica 1. Syntetyczne ujęcie metody wdrożeniowej IMPLEX, z odstępstwami od standardu, zastosowanej w FAMEG S.A. (procedury metody IMP...).

Etap	Wejście (We)	Czynności projektowe	Wyjście (Wy)
1	2	3	4
I	– umowa typu IMPLEX zawarta z klientem, – specyfikacja dysponowanych zasobów	– przeprowadzenie spotkania otwierającego (IMP141), – analiza środowiska biznesowego (IMP142), – wyodrębnienie istniejących procesów biznesowych BI (IMP143),	– definicja projektu, – wstępny raport o procesach biznesowych, – model biznesowy, – mapa BI, – plan etapu II.
II	WY etapu I	– wprowadzenie stosowanej metody BPR (Business Process Reengineeringu) - IMP241, – udokumentowanie istniejących BI (IMP242), – zdefiniowanie BI dla projektu (IMP243), – przeprojektowanie BI (IMP244).	– uaktualniona definicja i plan projektu, – raport o BI, (przebiegi, opisy), – projekt techniczny instalacji komputerowej – realizowany według oddzielnego harmonogramu, – plan etapu III.
III	WY etapu II	– opis konfiguracji (IMP341), – zainstalowanie środowiska technicznego (IMP342), – ustanowienie środowiska operacyjnego (IMP343), – konfigurowanie systemu MOVEX (IMP344), – szkolenie członków projektu (IMP345), – stwierdzenie ważności konfiguracji (IMP346), – zapełnianie PBD (IMP441), – ustanowienie instrukcji pracy początkowej (IMP347).	– skonfigurowane rozwiązanie – opis konfiguracji, – instrukcja pracy początkowej, – uaktualniony raport o procesach, – plan etapu IV.
IV	WY etapu III	– zapełnianie PBD (IMP441), – przeprowadzenie testu pilotażowego (IMP442), – szkolenie użytkowników	– analiza wyników testów wraz z podjęciem działań korygujących i zabezpieczających poprawne

		końcowych (IMP443), – przeprowadzenie testu pełnoskalowego (IMP444), – uzupełnienie i aktualizacja instrukcji stanowiskowych (IMP443).	wdrożenie, – zaktualizowany opis konfiguracji , – sprawozdanie o funkcjonowaniu BI, – plan etapu V.
V	WY etapu IV	– przygotowanie eksploatacji (IMP541), – rozpoczęcie eksploatacji (IMP542), – wsparcie eksploatacji (IMP543), – utworzenie wstępnego planu usprawnień (IMP544).	– zatwierdzenie wdrożenia rozwiązań, – wstępny plan usprawnień.
-	EKSPLOATACJA UŻYTKOWA WY etapu V	– procedury śledzenia funkcjonowania wdrożenia w odniesieniu do czasu i kosztów projektu (narzędzie MS Exel) , – listy kontrolne (narzędzie – Intentia), – dokumentacja (zgodna ze standardami Implex), – ANALIZA I OCENA JAKOŚCI WDROŻENIA PROJEKTU.	– systematyczne tworzenie docelowego planu usprawnień.

Opracowanie : autorzy.

2.3. Faza C – ustanowienie mechanizmu rozwoju

Obejmuje wypracowanie stałej procedury i mianowanie osób odpowiedzialnych za stałe dostosowywanie procesów i wspomagających narzędzi ERP do zmieniających się potrzeb przedsiębiorstwa. Rozpoczyna się ona z chwilą opracowanie przy zakończeniu fazy B (etap V) "listy dalszych usprawnień", a w zasadzie nigdy się nie kończy. Jest to mechanizm cyklicznych mini-projektów usprawnień rozpoczynanych po "uzbieraniu" i zatwierdzanych nowych list. W omawianym tutaj projekcie mechanizm ten otrzymał nazwę "konsolidacja". W praktyce konsolidacja sprowadza się do działań w następujących obszarach:

- wykorzystanie większej ilości funkcji (programów) we wdrożonych już modułach,
- wdrożenie nowych modułów we wdrożonej już wersji systemu,
- wdrożenie nowej wersji stosowanego systemu,
- wymiana systemu na inny; jednak taka decyzja może mieć miejsce kiedy kosztowne fazy A i B zakończyły się niepowodzeniem a ich złe przeprowadzenie z pewnością wtedy może być potraktowane jako działanie na szkodę firmy jeżeli przyczyny takiej sytuacji nie noszą znamion zawartych w kontrakcie w rozdziale "siła wyższa" (należy zawsze mieć świadomość w realizacji tak dużego przedsięwzięcia, że "informatyka to nie zabawka" [6,9]). Szczegółowe działania Famegu w tej fazie zostaną omówione w dalszej części niniejszego artykułu .

3.Struktura organizacji projektu

Jej schemat przedstawiono na rys.4. Ponieważ decyzję o wdrażaniu projektu "MOVEX" podjęto już w warunkach zaawansowanego wdrożenia systemu zarządzania jakością (SZJ) zgodnego z normą PN-ISO 9001:1996 (ISO 9001:1994) zdecydowano, że obydwa duże projekty będzie wdrażał ten sam zespół [6]. Podstawą decyzji zarządu o jednoczesnym prowadzeniu dwóch dużych projektów – MOVEX i SZJ – było procesowe podejście do ich wdrażania. Modelowanie procesów gospodarczych oraz projektowanie i dokumentowanie procedur jakościowych wymaga wykonania wielu prac o podobnym charakterze w tych samych obszarach. Powierzenie temu samemu zespołowi wdrożeniowemu tych dwóch projektów zapewnia możliwie maksymalne wykorzystanie MOVEX-a do informatycznego wsparcia SZJ oraz efektywne wykorzystanie wiedzy i umiejętności kadry fachowców.

Wdrożenia nadzorował Komitet Sterujący, w którego skład wchodzili:

- ze strony Famegu: dyrektor naczelny-prezes zarządu (sponsor projektów), trzech członków zarządu, szef informatyki-pełnomocnik zarządu ds. MOVEX i SZJ (Project Managament), audytor zewnętrzny; odpowiedzialnymi za nadzór ze strony zarządu byli dwaj jego członkowie – dyrektor ds. ekonomicznych i dyrektor ds. technicznych,
- ze strony Intentia (MOVEX): dwóch członków zarządu i lider projektu (szef zespołu przydzielonych konsultantów),
- ze strony Zetom Katowice (SZJ): szef przydzielonych konsultantów.

Audytor zewnętrzny funkcjonował tylko w części projektu dot. ERP.

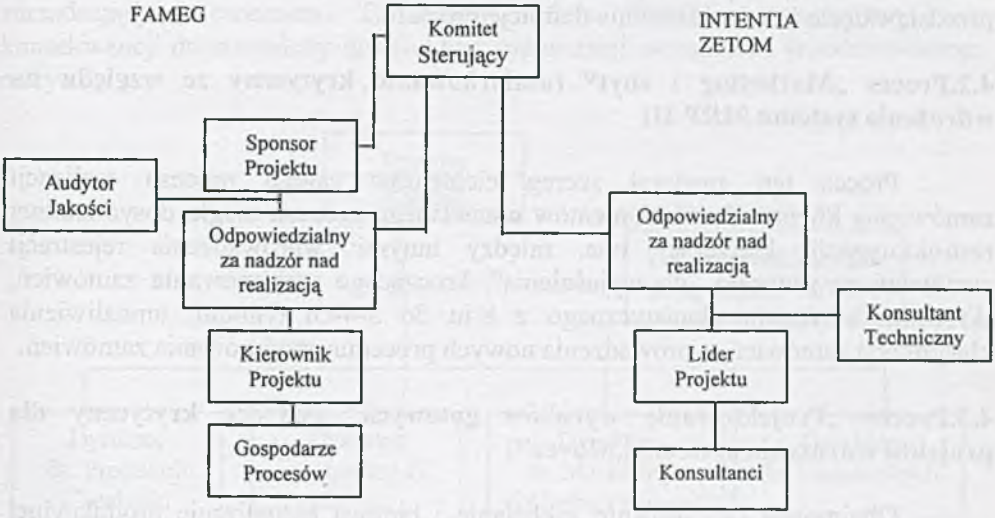
W skład Zespołu Wdrożeniowego wchodzili:

- ze strony Famegu: szef informatyki-pełnomocnik zarządu ...(kierownik zespołu), pięciu gospodarzy procesów, kierownik działu informatyki (odpowiedzialny za infrastrukturę informatyczną),
- ze strony Intentia: dyrektor organizacyjny, lider projektu, przydzieleni konsultanci (pięciu – jeden konsultant dla każdego procesu), konsultant techniczny,
- ze strony Zetom: konsultanci.

Kierownikowi zespołu wdrożeniowego podlegali gospodarze procesów, którzy w miarę pojawiających się potrzeb włączali do projektów podległych im pracowników. Gospodarze procesów byli również odpowiedzialni za tworzenie dokumentacji ERP i SZJ właściwych swoim funkcjom części systemów, a także za przeszkolenie użytkowników końcowych [6]. Żaden z pracowników Famegu przydzielonych do projektów nie był zwolniony z dotychczas pełnionych obowiązków; za prace wdrożeniowe pracownicy ci otrzymywali dodatkowe wynagrodzenie za "nieograniczony czas pracy".

Gospodarz procesu – to specyficzna dla metody Implex nazwa właściciela procesu (process owner), za jego obsługę podczas wdrożenia systemu i w jego eksploatacji

użytkowej (w ramach nowej, zorientowanej na procesy struktury organizacyjnej firmy).



Rys. 4. Struktura projektu MOVEX v.9A
(pierwsze wdrożenie)

4. Procesy gospodarcze

Wdrażając projekty MOVEX i SZJ wyodrębniono 5 podstawowych procesów gospodarczych: produkcji, marketingu i zbytu, projektowanie wyrobów gotowych (wraz z konstrukcyjną bazą danych – PBD), zaopatrzenia, finansów i księgowości (wraz z kosztami utrzymania ruchu). Każdy z procesów podzielono na podprocesy, te z kolei na czynności, które podzielono na kroki [10,4].

4.1.Proces „Produkcja" (najistotniejszy proces z punktu widzenia osiągnięcia wdrożenia).

Dla tego procesu opracowane zostały dwa schematy diagrametryczne [10] dla zakładów produkcyjnych nr 1 i nr 3, które zlokalizowane są w różnych częściach miasta. Schematy te nie były klasycznymi mapami procesów. a stanowiły opis przepływu materiałów i części konstrukcyjnych oraz definiowały ich inspekcję w wydziałach produkcyjnych, co było uzasadnione specyfiką procesu technologicznego. Powstał tutaj problem dokumentów generowanych przez system [4,6,10]. Wynikał on ze specyfiki „trudnej" technologii w Famegu. Emitowanie jednej karty zlecenia dla całego procesu technologicznego – wydawało się pierwotnie, że zajdzie konieczność zastosowania tutaj rozwiązań poza

standardowych. Jednak zorganizowanie magazynów buforowych pozwoliło zastosować programy standardowe w systemowej dokumentacji produkcyjnej. Był to moment krytyczny, który mógł w znacznym stopniu skomplikować całe przedsięwzięcie i zmienić istotnie definicję projektu.

4.2.Proces „Marketing i zbyt" (umiarkowanie krytyczny ze względu na wdrożenie systemu MRP II)

Proces ten zawierał szereg elementów całego procesu realizacji zamówienia klienta. Część elementów omawianego procesu uległo dosyć istotnej restrukturyzacji. Dotyczyły one, między innymi: wprowadzenia rejestracji zamówień ze statusem „do wyjaśnienia", kroczącego przyjmowania zamówień, skrócenia horyzontu planistycznego z 8-iu do 3-4-ch tygodni, umożliwienia klasyfikacji zamówień, wprowadzenia nowych procedur przyjmowania zamówień.

4.3.Proces „Projektowanie wyrobów gotowych" (wysoce krytyczny dla projektu wdrożenia systemu „Movex")

Obejmował on głównie zakładanie i bieżącą aktualizację produkcyjnej bazy danych – PBD [4,10] wraz z całym systemem kodowania „rzeczy i czynności". Tutaj trudnościami były głównie problemy organizacyjne, które znalazły się w gestii kierownika projektu, między innymi zagospodarowywania „obszarów ziemi niczyjej".

4.4.Proces „Zaopatrzenie" (kluczowy dla sukcesu przedsięwzięcia jako całości).

W procesie tym dokonano pełnej charakterystyki obszaru tak z punktu widzenia Business Process Reengineering (BPR), jak i modelu planowania i kontroli produkcji. Tutaj projekt restrukturyzacji był najbardziej głęboki. „Główny harmonogram Produkcji" stanowił podstawę dla konstrukcji „Planu Budżetu Zakupów Surowców Drzewnych". Równolegle tworzony jest „Plan Finansowy" na bazie m.in. „Głównego Planu Popytu", z którego wynika „Plan Budżetu Zakupów Materiałów Pomocniczych i Technicznych". W tym procesie zdefiniowane zostały elementy należące do obszaru gospodarki magazynowej.

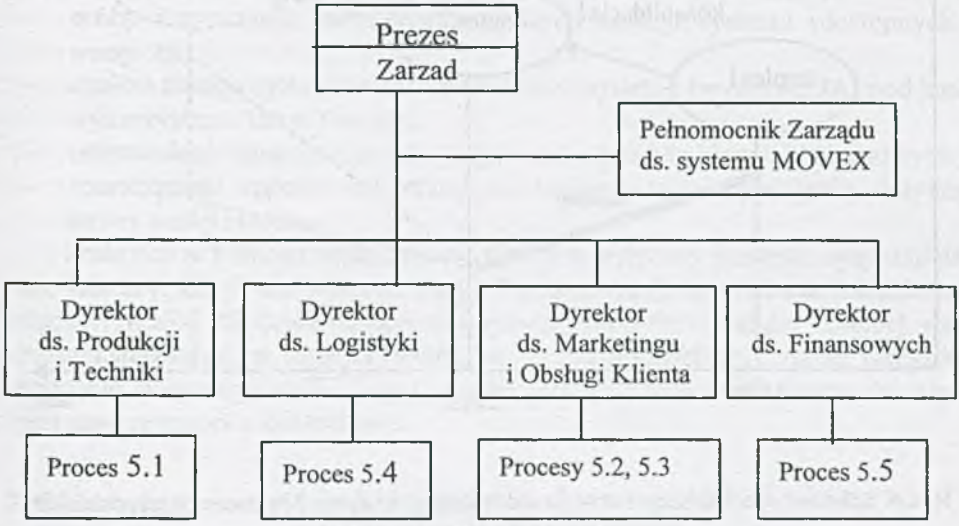
4.5.Proces „Rozwój Firmy" (niekrytyczny dla wdrożenia systemu MOVEX).

W sensie BPR jest grupą podprocesów. Restrukturyzacji podlegały podprocesy: zarządzanie finansami, tworzenie polityki cenowej, utrzymanie zdolności produkcyjnych.

5. Zmiany w organizacji zarządzania firmą.

Wdrożenie systemu MOVEX v.9A w ujęciu procesowym stworzyło warunki do wprowadzenia usprawnień w organizacji zarządzania firmą. Stopniowo

odchodzono od tradycyjnej struktury zarządzania kierując się ku procesowemu zarządzaniu. Zarząd firmy zmniejszył się do dwóch osób. Natomiast dotychczasowi gospodarze procesów zostali mianowani dyrektorami zarządzającymi procesem. Te zmiany miały charakter ewolucyjny i w konsekwencji doprowadziły do struktury organizacji zarządzania przedstawionego na rys.5.



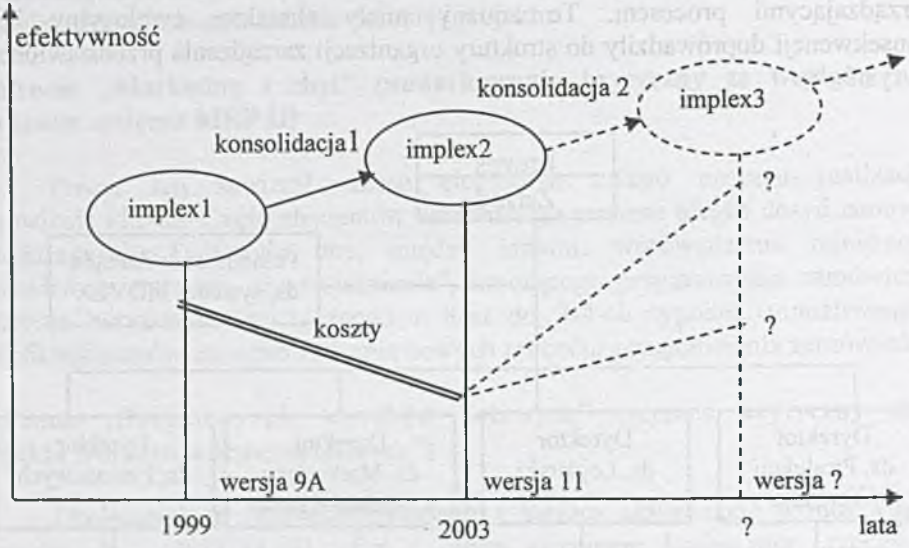
Rys. 5. Aktualny ogólny schemat organizacyjny FAMEG S.A.
(organizacja procesowa)

Taka organizacja jest elastyczna i pozwala zmieniać , w razie potrzeby, podporządkowanie procesów poszczególnym dyrektorom. Już w trakcie wdrażania systemu okazało się , że celowe jest podporządkowanie działu konstrukcyjno-technologicznego dyrektorowi ds. marketingu i obsługi klientów , co na początku spotkało się z dużym zdziwieniem i niedowierzaniem. Jednak taka zmiana organizacyjna pozwoliła na skrócenie realizacji zlecenia produkcyjnego , a tym samym zamówienia klienta z 8-miu do 3-4-ch tygodni.

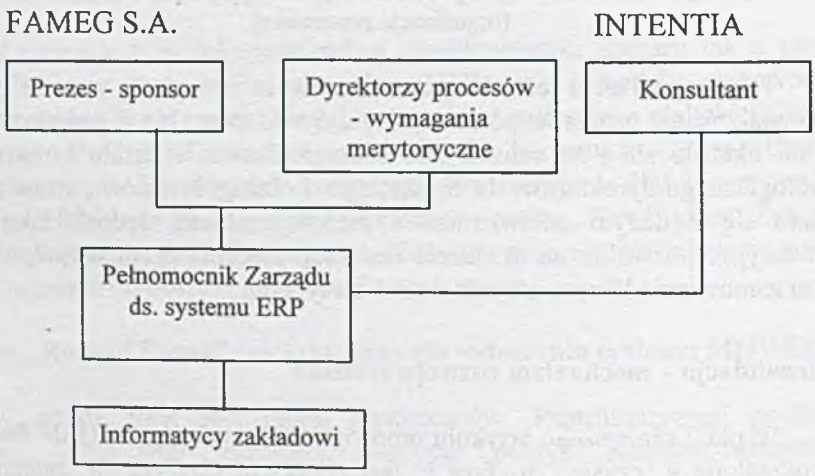
6. Konsolidacja – mechanizm rozwoju systemu

W pkt.2 niniejszego artykułu omówiono fazy wdrożenia. O ile fazy A i B były określone w czasie , to faza C jest procesem ciągłym. W zależności od merytorycznej zawartości wyników konsolidacji (fazy C), fazy A i B będą w niej realizowane cyklicznie w mniejszym lub większym stopniu. Konsultanci wdrożeniowi (dostawcy systemu) są niezbędni przy pierwszym cyklu ABC. W następnych cyklach odgrywają coraz mniejszą rolę wraz z czasem działania mechanizmu rozwoju systemu, choć prawdopodobnie nie da się z nich zrezygnować [6]. Rys. 6 przedstawia mechanizm rozwoju systemu w Famegu,

który zakończył się wdrożeniem wersji 11-tej Movexa; organizację tego wdrożenia przedstawia rys.7.



Rys.6. Schemat mechanizmu rozwoju wdrożonego systemu Movex z uwzględnieniem tendencji kształtowania się kosztów



Rys. 7. Struktura organizacji wdrożenia wersji 11-tej systemu MOVEX (drugie wdrożenie)

W Famegu ustanowienie mechanizmu rozwoju systemu Movex przewidziano już na etapie podpisania z Intentia-Polska (d.Intentia-Vimex) kontraktu głównego gwarantującego korzystanie z opcją DEVEX (patrz pkt.2.1). W ramach fazy konsolidacji zakładowi informatycy mieli możliwość zapoznania z wdrożeniami w Polsce różnych wersji systemu Movex (wyższych od wersji 9A). W znacznym stopniu ułatwiło to wybór wersji 11-tej. W fazie konsolidacji przeprowadzono szereg różnych działań. Do najważniejszych należały :

- próby i symulacja zastosowania nowych funkcji systemu (dostępnych w wersji 9A),
- analiza funkcji systemu w innych wersjach systemu (wyżej od 9A) pod kątem wykorzystania ich w Famegu,
- usprawnienie funkcjonujących już procesów produkcyjnych i biznesowych,
- sporządzenie raportu zawierającego merytoryczne wymagania dotyczące nowej wersji systemu.

Praktyka w Famegu wykazała, że 60-cio miesięczny kontrakt opcji DEVEX nie jest zbyt długi, ale wystarczający. Zaangażowanie konsultanta Intentii w tej fazie wynosiło 20 dniówek konsultacyjnych. Wdrożenie każdej kolejnej wersji systemu wymaga, w mniejszym lub większym stopniu, wykonania wszystkich procedur poszczególnych (5-ciu) etapów metody Implex. Wejściem do etapu I jest zawsze raport z konsolidacji.

7. Efektywność zastosowań informatyki

7.1. Uwarunkowania wdrożeń i eksploatacji systemów informatycznych

a. Od 1971 roku systemy były wdrażane i przetwarzane w ZETO-Łódź [4]. Początkowo tam też przygotowywane były maszynowe nośniki danych (mnd). Odległość pomiędzy Famegiem i ZETO-Łódź wynosiła ok.100 km. W ZETO-Łódź wykonywano również oprogramowanie wielu systemów.
b. W miarę rozwoju zastosowań informatyki w Fameg zorganizowano zdecentralizowane stacje tworzenia mnd wyposażone w: dalekopisy z perforatorami, dziurkarki i sprawdzarki kart, automaty obrachunkowe z perforatorami taśmy papierowej, automaty organizacyjne z perforatorami i czytnikami taśmy papierowej; szczegóły w [4].
c. Wdrożenie niektórych systemów było obligatoryjne (decydenci: Zjednoczenie Przemysłu Meblarskiego, Ministerstwo Leśnictwa i Przemysłu Drzewnego oraz Ministerstwo Przemysły Chemicznego (KTM-y)[4].
d. Od 1980 roku przetwarzanie systemów odbywało się w filialnym ośrodku obliczeniowym ZETO-Łódź w Radomsku (teren Fameg) wyposażonym w ODRA 1304, później ODRA 1305 i MERA 9150 [4]. Ośrodek funkcjonował do końca czerwca 1993 roku.
e. W latach 1991-1994 wszystkie systemy eksploatowane na ODRA 1305 były stopniowo przenoszone na PC-ty.

f. W latach 1997-1999 zrealizowano dużą inwestycję informatyczną – wdrożenie zintegrowanego systemu informatycznego klasy MRPII/ERP „Movex" na platformie sprzętowej IBM AS/400 wraz z pełną infrastrukturą (serwerownia, biura mistrzów, okablowanie strukturalne).
g. W 2003 roku wdrożono nową (11-tą) wersję systemu Movex.

7.2.Ocena efektywności

Wymierne efekty ekonomiczne zestawiono z ponoszonymi nakładami ekonomicznymi stosując formuły A.Kierczyńskiego. Jedną z zastosowanych formuł jest porównanie kosztów eksploatacji systemów i kosztów techniki tradycyjnej [2,8]. W efekcie uzyskano wskaźnik efektywności „E_k" pokazujący, ile razy „tradycyjna" technika obliczeniowa jest droższa niż komputerowa. W przypadku, gdy wprowadzenie techniki komputerowej pozwala na zwiększenie zasobu przetwarzanych informacji i powoduje dodatkowe efekty ekonomiczne, należy dodać do kosztów „starej" techniki (w celu uzyskania porównywalności obu technik) tak zwany **koszt hipotetyczny** , przez który należy rozumieć koszt , jaki należałoby ponosić, gdyby daną pracę wykonywaną obecnie przez komputer trzeba było wykonać tradycyjną techniką. Drugim wskaźnikiem jest okres zwrotu nakładów „T", stosowany jest powszechnie przy ocenie efektywności dużych projektów inwestycyjnych. Kształtowanie się tych wskaźników w latach 1971-2004 przedstawiono w tablicy 2.

Tablica 2

Wskaźnik	Okresy	1971-1976	1977-1978	1980	1981-1985	1986-1990	1991-1995	1999-2001	1999-2003	2002-2003	2004
T	X	0,89	0,81	0,71	0,93	0,84	1,1	2,7	0,74	0,45	0,24
E_k	X	1,15	1,22	1,41	1,07	1,19	0,91	0,37	1,35	2,22	4,1

7.3.Efekty niewymierne

W Famegu do efektów organizacyjno-technicznych zaliczono: usprawnienie systemu informacyjnego, zmiany w strukturze zatrudnienia pracowników administracyjno-biurowych, ujednolicenie i unifikacja dokumentów źródłowych wykorzystywanych w różnych komórkach funkcjonalnych, znaczne skrócenie czasu wykonywanych obliczeń, wyeliminowanie szeregu żmudnych powtarzalnych obliczeń w systemie tradycyjnym, usprawnienie ewidencji zasobów gospodarczych, polepszenie czytelności i rytmiczności obiegu dokumentów. Po wdrożeniu Movex-a wystąpiły nowe efekty – skrócony został czas realizacji zamówienia klienta z 8 do 3-4 tygodni. Nie spowodował tego wzrost zdolności produkcyjnych; efekty wystąpiły z lepszego harmonogramowania produkcji i lepszej realizacji zleceń produkcyjnych. W konsekwencji rośnie zadowolenie klienta i konkurencyjność firmy. Czy te efekty da się oszacować jakimś obiektywnym wskaźnikiem ekonomicznym?

8.Wspomaganie SZJ przez systemy ERP

Systemy ERP posiadają zintegrowane moduły jakości co zwiększa ich atrakcyjność poprzez uzyskiwanie szybkiego dostępu („na bieżąco") do systemu o modułowej budowie do planowania i kierowania procesami związanymi z SZJ; szerzej o tym w [10].

Na rynku krajowym kilka firm oferuje oprogramowanie będące narzędziami służącymi do budowania modeli biznesowych. Są to zbiory narzędzi łączących modelowanie opisowe i analityczne. Nie są one na ogół zintegrowane z procesami biznesowymi. Pozwalają jednak je nadzorować i doskonalić oraz sterować nimi. Stanowią one skuteczne narzędzia do tworzenia, publikowania oraz nadzorowania dokumentacji SZJ, niezależnie od stopnia integracji funkcjonujących w firmach systemów informatycznych. Tradycyjna dokumentacja SZJ składa się z księgi jakości, szeregu procedur, instrukcji oraz załączników. SZJ wymaga, aby komórki organizacyjne miały dostęp do dokumentacji niezbędnej do zapewnienia prawidłowej realizacji zadań [10].

Praktyka firmy meblarskiej, w której wdrożono system MRPII/ERP „MOVEX", dowiodła, że w dużym przedsiębiorstwie produkcyjnym wdrażanie i funkcjonowanie SZJ okazuje się bardzo trudne bez informatycznego wsparcia. Choć „MOVEX" nie zapewnia wszystkich potrzeb w tym zakresie, to jego funkcjonowanie w zasadniczy sposób upraszcza dokumentowanie SZJ przy wykorzystaniu standardowych narzędzi systemu informatycznego. Wsparcie to jest możliwe na kilku poziomach.

Poziom I – operacja technologiczna (patrz rys.1) w produkcyjne bazie danych (PBD). Wykorzystano tutaj standardowe pola tekstowe dostępne na poziomie operacji na każdym stanowisku pracy, dla każdego elementu struktury produktu (materiał, części składowe, podzespół, produkt-wyrób, komplet wyrobów). W polach tych o pojemności 2-4 wierszy ekranowych zamieszczane są mierzalne parametry dla operacji; chodzi o parametry podlegające inspekcji dozoru technicznego i kontrolerów jakości.

Poziom II – narzędzie „notes", który może być „podpięty" do każdego identyfikatora elementu struktury w PBD (pojemność od 900 wierszy ekranowych). To narzędzie umożliwia wprowadzenie do PBD niezbędnych instrukcji, procedur, wymagań klienta, itp.

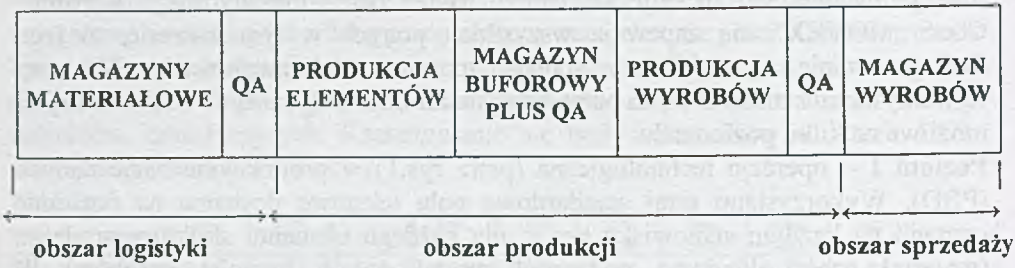
Poziom III – stosowanie standardowych modułów jakości QA. Do procedur związanych z podprocesem QA należą: obsługa specyfikacji kontroli, tworzenie żądań, raportowanie, raportowanie wyników, monitorowanie świadectw jakości partii i specyfikacji, śledzenie partii, tworzenie statystyk. Dla każdego elementu struktury w PBD użytkownik ma możliwość swobodnego definiowania „klas jakości", tzn. cech poddawanych inspekcji; szerzej o tym w [10,3].

Poziomy I i II mają zastosowanie do wprowadzania danych i opisów **względnie stałych** do PBD. Informacje te są dostępne dla wszystkich użytkowników systemów ERP i SZJ bez możliwości dokonywania przez nich modyfikacji. Przy takim rozwiązaniu możliwa jest szybka aktualizacja tych informacji przez osoby uprawnione u wszystkich użytkowników funkcjonujących w tych systemach.

Poziom III jest tylko systemowo „zakotwiczony" w PBD. Sztywne powiązanie modułów QA z PBD grozi odczuwalnym spowolnieniem lub blokowaniem systemu ERP. Jednak zwolnienie partii do dalszego procesu produkcji jest możliwe tylko po pozytywnym wyniku inspekcji. **Na etapie konfiguracji systemu ERP należy przewidzieć lokalizacje inspekcji (np. magazyny materiałowe, magazyny międzyoperacyjne – międzywydziałowe, wybrane – ważne stanowiska pracy, itp.). Nie wszystkie systemowe QA muszą być aktywne; są uaktywniane w miarę potrzeb przez administratorów systemu.**

Te narzędzia systemowe stanowią ważny element w SZJ. Umiejętne ich stosowanie umożliwia skonfigurowanie systemu ERP, który pozwala skutecznie, systemowo zapewnić funkcjonowanie SZJ w obszarach tutaj omówionych (tzw. „wymuszenie przez system").

OGÓLNY SCHEMAT ORGANIZACJI PROCESU WYTWÓRCZEGO PO WDROŻENIU SYSTEMU „MOVEX" W FAMEG S.A.



Rys. 8. Ogólny schemat procesu wytwórczego z modułami jakości QA

Instalacja modułów QA na etapie konfiguracji systemu ma bardzo istotne znaczenie. Zostawienie tego problemu „na później" spowoduje konieczność ponownej, kosztownej konfiguracji systemu.

Tablica 3. Możliwości wspomagania SZJ w ERP (przykłady)

Lp	Obszar SZJ	Moduły ERP	PN-EN ISO 9001:2001	Uwagi
1	2	3	4	5
1.	Parametry produkcji wyrobów podlegające inspekcji (operacje technologiczne)	MRPII – PBD (poziomy I, II, III)	Ust. 7.5.1, 7.5.2, 7.6, 8.2.3, 8.2.4, 8.1	
2.	Identyfikacja elementów struktur wyrobów w PBD	MRPII – Zaopatrzenie	Ust. 7.5.3	
3.	Zabezpieczenie wyrobu	Notes (PBD-poziom II)	Ust. 7.5.5	
4.	Wynik oceny dostawcy	Zaopatrzenie	Ust. 7.4.1, 7.5.3	Spełnione wymaganie

				„zapis"
5.	Polityka jakości	Notes – poziom II w PBD	Ust. 5.1	Specjalne struktury dla dokumentacji SZJ w PBD
6.	Procedury	Notes – poziom II w PBD	Ust. 4.2.3, 4.2.4, 8.2.2, 8.3, 8.5.2, 8.5.3, 6.1, 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 8	
7.	Księga Jakości + opis procesów	Notes – poziom II w PBD, interfejs graficzny	4.2.2	Konieczny interfejs graficzny
8.	Statystyki	PBD - poziom III	8	

Niecelowym działaniem jest wprowadzanie wymaganych „zapisów" SZJ w systemie ERP. Chodzi o to, aby nadmiernie nie „rozdmuchiwać rozmiarów PBD". Pozycje 5,6,7 w tablicy 3 należą do tzw. danych względnie stałych i w żaden sposób nie wpływają na spowolnienie działania systemu ERP. Aktualizacja części podstawowej dokumentacji SZJ jest wykonywana przez uprawnionego pracownika w tym samym czasie u wszystkich użytkowników SZJ na wszystkich workstation.

Pozostałe elementy dokumentacji SZJ mogą być prowadzone przy zastosowaniu dostępnego na rynku oprogramowania do publikowania i nadzorowania lub opracowanego w firmie we własnym zakresie oprogramowania przy zastosowaniu Word i Exel.

Zakończenie

Omówienie wszystkich aspektów zarządzania tak dużym projektem nie jest możliwe w ramach jednego artykułu. Dlatego też w tekście zamieszczono stosunkowo dużą ilość odsyłaczy do poszczególnych pozycji w wykazie literatury stanowiących integralną część niniejszego artykułu. W szczególności dotyczy to tych pozycji, które są zbiorowym efektem prac wykonanych podczas XV, XVI i XVII Górskich Szkół PTI w Szczyrku.

Literatura

1. Askanas W. – Konflikty organizacyjne przy wdrażaniu eto. PWN 1978.
2. Jasiorowski K.,Koch J. – Efektywność zastosowania epd w Zakładach Przemysłu Meblarskiego w Radomsku. Przemysł Drzewny Nr 4/1979.
3. Jasiorowski K., Jasiorowski R. – Wdrożenie zintegrowanego systemu MRPII/ERP „Movex" w Fameg S.A. z uwzględnieniem wspomagania systemu zarządzania jakością (SZJ) zgodnego z normami ISO 9000. PTI – Systemy informatyczne. Zastosowanie i wdrożenia 2003 pod redakcją J.K.Grabary i J.S.Nowaka. WNT, Warszawa- Szczyrk 2003, Tom III, cz.1.

4. Jasiorowski K., Jasiorowski R., Nowak J.S. – Informatyka w FAMEG S.A. w Radomsku w latach 1965 – 2004. Efektywność zastosowań informatycznych 2004 pod redakcją Janusza K.Grabary i Jerzego S.Nowaka. WNT, Warszawa 2004.
5. Jasiorowski K., Nowak J.S., Jasiorowski R. – Efektywność informatyki w Fameg S.A. w Radomsku. Efektywność zastosowań systemów informatycznych 2004 pod redakcją J.K.Grabary i J.S.Nowaka. WNT, Warszawa 2004.
6. Jasiorowski K., Nowak J.S., Jasiorowski R. – Organizacja wdrożeń systemu ERP w Fameg S.A. w Radomsku. Informatyka we współczesnym zarządzaniu pod redakcją J.Kisielnickiego, J. S.Nowaka. i J.K.Grabary. WNT, Warszawa 2004.
7. Jasiorowski K., Nowak J.S., Jasiorowski R. – Wspomaganie systemów zarządzania jakością (wg PN-EN ISO 9001:2001). Informatyka i współczesne zarządzanie pod redakcją J.Kisielnickiego, J.K.Grabary i J. S.Nowaka.. PTI – Oddział Górnośląski, Katowice 2005.
8. Kierczyński A. – Efektywność komputeryzacji. PWE, 1975.
9. Maciejec L. – Informatyka w przedsiębiorstwie to nie zabawka. UIK, Warszawa 1995.
10. Zintegrowany system informatyczny klasy MRPII/ERP „MOVEX" w Fameg S.A. w Radomsku. Materiały wdrożeniowe, Radomsko luty 2001.

ROZDZIAŁ VIII

EFEKTYWNE ZARZĄDZANIE RYZYKIEM W PROJEKTACH WDROŻEŃ ZINTEGROWANYCH SYSTEMÓW INFORMATYCZNYCH

Anna KACZOROWSKA

W dziedzinie takiej jak realizacja bardzo dużych projektów informatycznych oszacowanie ich złożoności jest kluczem do rozwiązywania pojawiających się zagrożeń i ograniczania strat ekonomicznych, ale także społecznych i politycznych, jakie przynoszą nieudane wdrożenia systemów informatycznych.

Wdrożenie ZSI typu ERP rozbudowanego o specjalistyczne dziedzinowe systemy z konkretnej branży jest zamierzeniem implikującym wykonanie wielu różnorodnych działań. Dla zmniejszenia lub wyeliminowania trudności, po pierwsze proponuje się metodykę efektywnie „obsługującą" ryzyko związane z informatyzacją danej organizacji. Została ona przez autorkę nazwana Metodyką Rezultatów Kontrolnych (MRK), ponieważ określone i dające się sprawdzić rezultaty odnośnie zakresu prac winny kończyć każdy wyróżniony etap projektu.

1. Metodyczne wdrożenie zintegrowanego systemu informatycznego

Wdrożenie ZSI rozbudowanego o specjalistyczne dziedzinowe systemy wymaga metodycznego podejścia. Metodyka wdrożenia jest to sformalizowany, szczegółowy opis w podziale na poszczególne etapy i grupy czynności wykonywane podczas implementacji. Powinna obejmować wszelkie działania począwszy od etapu przygotowania projektu aż po fazę postimplementacyjnego testowania wdrożonego systemu. W przedsięwzięciu wdrażania ZSI największą wartość mają wiedza i doświadczenie osób zaangażowanych we wdrażanie, metodyka jest zaś tym narzędziem, które wspomaga i syntetyzuje ich pracę. Uporządkowanie prac jest szczególnie ważne w przypadku, gdy kadra zarządzająca i informatycy mają małe doświadczenie w pracy z systemami zintegrowanymi.

Podstawową zasadą prawidłowego zarządzania projektem wdrożenia ZSI jest wydzielenie go z rutynowych zadań w istniejących pionowych strukturach na uczelni (wraz z nadaniem mu dostatecznie wysokiego priorytetu na okres jego trwania) i jasne, zrozumiałe wyróżnienie wymiernych rezultatów, które etapowa realizacja projektu ma zapewnić. Uzyskane rezultaty będą, bowiem miernikiem osiągnięcia określonych celów projektowych, czy to szybszej odpowiedzi na zapytanie o aktualny obraz sytuacji ekonomicznej uczelni czy też wyższy poziom bezpieczeństwa informacji przetwarzanych w systemie przy wzrastającym wolumenie zbiorów danych. Określone i mierzalne rezultaty odnośnie do zakresu prac i odpowiedniej dokumentacji winny kończyć każdy wyróżniony etap. Stąd też,

autorka nazwała metodykę - **Metodyką Rezultatów Kontrolnych** (MRK). Rezultaty kontrolne stanowią najistotniejszy element zarządzania na taktycznym poziomie zarządzania projektem, ponieważ określają one efekty osiągane w kolejnych krokach działań. Przy czym, wyniki wszystkich etapów projektu wdrożenia ZSI, powinny być specyfikowane ze szczególną starannością, ponieważ w przypadku przerwania działań w dowolnym momencie, prowadzona na bieżąco dokumentacja da kompleksowy obraz zrealizowanych prac i może być podstawą do ich wznowienia przez innych wykonawców.

Dla przedstawianej metodyki zarządzania przedsięwzięciem wdrożeniowym ZSI zakłada się dodatkowo wspomaganie procesu zarządzania projektem pakietami informatycznymi typu Ms Project Professional czy Project + (zapewniają one, bowiem, kontrolę stworzonego wcześniej za ich pomocą kompleksowego planu przedsięwzięcia; monitorują na bieżąco przebieg działań, identyfikują odchylenia od planu i na ich podstawie umożliwiają podejmowanie działań korygujących).

Proponowana MRK zakłada podział przedsięwzięcia wdrożeniowego na następujące etapy:

1. Bilans dotychczasowego stanu komputeryzacji i elektronizacji w informatyzowanej uczelni (rezultatami tego etapu powinien być raport zawierający opis funkcjonujących systemów informatycznych lub tradycyjnych procedur przetwarzania danych, opis procesów, zakres koniecznej restrukturyzacji obiektu).
2. Zdefiniowanie zadania projektowego
   2.1. Utworzenie specjalistycznej – projektowej struktury organizacyjnej (na początek ramowej struktury organizacyjnej bez dostawcy.
   2.2. Określenie mierzalnych celów przedsięwzięcia (określane są tu: funkcjonalny zakres przyszłego ZSI, organizacja i zbiory danych, terytorialne zorganizowanie systemu, sprzęt).
   2.3. Opis wszystkich zauważonych uwarunkowań wykonania prac, możliwych do uwzględnienia na tym początkowym etapie przedsięwzięcia (specyfikacja czynników krytycznych dla projektu).
3. Modyfikacje utworzonej wcześniej projektowej struktury organizacyjnej z uwzględnieniem w niej ścisłej współpracy klienta i dostawcy.
4. Przygotowanie planu prac wdrożeniowych
   4.1. Analiza przedwdrożeniowa systemu przeprowadzana przez dostawcę (opisuje jak wygląda obecna struktura organizacyjna jednostki, jak kształtują się procesy uczelni, jak wyróżnione procesy zostaną zamodelowane w systemie przez dostawcę; powstaje tutaj, ewentualnie, opis wymaganej infrastruktury technicznej na potrzeby wdrażanego sytemu).
   4.2. Przygotowanie planu prac wdrożeniowych i szczegółowa kalkulacja budżetu przedsięwzięcia w informatycznym systemie wspomagającym zarządzanie projektami, np. Ms Project Professional 2003 oraz planu zapewnienia jakości (może on zawierać m.in. przegląd specyfikacji wymagań, projekt techniczny oprogramowania aplikacyjnego, wstępny i ostateczny – krytyczny przegląd projektu technicznego); zakłada się

kilkuetapowe omówienie projektu wstępnego dla ewentualnego usunięcia błędów, sprawdzenia uwzględnienia dotychczasowych wymagań i przedstawienia dodatkowych, zgłoszenia ewentualnych modyfikacji rozwiązań przez dostawcę lub ujawnienie przez niego niespójności w wymaganiach klienta; szczególnie dokładne zaplanowanie przejścia ze starych systemów na nowe).

4.3. Oszacowanie wartości ryzyka na podstawie listy priorytetowych zagrożeń projektu i ewentualne przekształcenie niektórych zagrożeń w koszty stałe (poprzez ubezpieczenie, kwotę gwarancyjną, umowy z podwykonawcami na zasadzie płatności o dzieło, zakup licencji).

5. Wdrażanie wybranego ZSI i ciągły monitoring realizacji planu pod względem czasu i kosztów (w Ms Project).

5.1. Przeprowadzenie koniecznej restrukturyzacji w obiekcie (określenie nowej lub modyfikacja istniejącej strategii działania obiektu; wyodrębnienie procesów podstawowych i pomocniczych, którymi należy zarządzać ze szczególnym uwzględnieniem mierzenia wyników obiektu z punktu widzenia jego klientów, ukształtowanie lub zmiany w systemie jakości jednostki; określenie sposobów rekrutacji i motywowania pracowników; *przeprowadzenie koniecznych zmian właśnie w tym momencie odpowiada praktyce większości wdrożeń ZSI w tych organizacjach; zmiany te powinny być już brane pod uwagę na etapie określania celów przedsięwzięcia (punkt 2.2) i szczegółowo przemyślane podczas analizy przedwdrożeniowej systemu (punkt 4.1)).*

5.2. Kompletacja i organizacja dostaw sprzętu i oprogramowania,

5.3. Instalacja systemu wraz z migracją danych ze starych systemów (skonfigurowanie szeregu parametrów zainstalowanego systemu zgodnie z określonymi wymaganiami uczelni; uruchomienie systemu na próbnej *bazie testowej*; opracowanie niezbędnych interfejsów i programów do migracji danych z dotychczas eksploatowanych u klienta systemów; migracja danych ze starych systemów; sprawdzenie próbne translacji istniejących baz danych),

5.4. Szkolenia pracowników uczelni z zespołów roboczych (projektowych), odpowiedzialnych za wdrożenia poszczególnych modułów funkcjonalnych systemu.

5.5. Szkolenia zorientowane na kadrę zarządzającą uczelnią (w pierwszej kolejności zarząd; dla każdego członka zarządu min 1 dzień; szkolenia nie przy komputerze, ale obejmujące logikę przetwarzania i przepływu informacji w systemie; następnie Dziekani wydziałów – użytkownicy kluczowi razem z użytkownikami modułów).

5.6. Szkolenia (systemowe i aplikacyjne) zorientowane na użytkowników sytemu (szczegółowe szkolenia dla użytkowników poszczególnych modułów; szkolenie administratorów systemu - 1 dzień).

5.7. Wdrożenie i testowanie oprogramowania aplikacyjnego (*należy określić zakres testowania w planie przedsięwzięcia*).

6. Przeprowadzenie odbioru formalnego (np. z określeniem okresu wstępnego).

7. Bieżąca eksploatacja i rozwój systemu (*ustalenie zasad rozwoju systemu oraz zakresu obsługi poprzez gwarancje i umowy serwisowe w kontrakcie na projekt*).

2. Użyteczne tabele punktów funkcyjnych

Opracowana Metodyka Rezultatów Kontrolnych zawiera autorskie studium wykonalności systemu informatycznego, w którym szacowana jest wielkość badanego rozwiązania za pomocą Metody Punktów Funkcyjnych. Zastosowanie tej metody pozwala na jednorodną ocenę systemów informatycznych różnych klas zastosowań pod wieloma kontami, takimi jak:

- złożoność systemów,
- wydajność zespołów wykonawczych,
- pracochłonność niezbędna do wykonania systemu,
- czas realizacji projektu.

Niezależnie jednak od zastosowanej metodyki Klient, aby dobrze wynegocjować kontrakt na projekt powinien umieć oszacować rozmiary przedsięwzięcia i posiadać kompetencje kontrolne odnośnie stanu zaawansowania prac. Firma, która wstępnie oszacuje rozmiary systemu informatycznego na miarę swoich potrzeb, za pomocą proponowanego w MRK studium opartego na Metodzie Punktów Funkcyjnych, będzie dysponowała jednorodną miarą, którą będzie mogła przeliczać na różne inne wielkości przypadające na punkt funkcyjny.

Po wyznaczeniu podstawowych parametrów projektu informatycznego, w pierwszym etapie studium, autorka zachęca do skorzystania z wniosków, do których dochodzi w czasie analizy uniwersalnych tabel punktów funkcyjnych, w drugim etapie studium wykonalności SI. Firma,, który wykorzysta informacje z obydwu etapów studium będzie mógłał stosunkowo precyzyjnie skonstruować założenia kontraktowe i pewniej prowadzić negocjacje z dostawcą wybranego systemu informatycznego.

I tak, dane przedstawione w tabeli 1 umożliwiają wyznaczenie szacowanej pracochłonności w zależności od wielkości ocenianego systemu wyrażonej w punktach funkcyjnych. Na podstawie danych z tej tabeli można również ocenić wydajność zespołów wykonawczych pracujących nad różnymi systemami informatycznymi. Oszacowana pracochłonność może stanowić podstawę do podejmowania konkretnych decyzji wykonawczych takich jak: określenie przez kierownika projektu profilu zatrudnienia w projekcie, wyliczenie globalnego czasu realizacji projektu z odpowiednim rozdziałem czasu produkcyjnego i nieprodukcyjnego.

Tabela 1. Średnia produktywność wyrażona w punktach funkcyjnych na osobomiesiąc i ilość godzin potrzebnych na wyprodukowanie jednego punktu funkcyjnego dla działań podejmowanych w różnych fazach projektu

Lp.	Czynności projektowe	Punkty funkcyjne/miesiąc			godziny pracy/1 punkt funkcyjny		
		MIN	MOD	MAX	MAX	MOD[7]	MIN
1	Wymagania (requirements)	50	175	350	2,64	0,75	0,38
2	Prototypowanie (prototyping)	25	150	250	5,28	0,88	0,53
3	Architektura (architecture)	100	300	500	1,32	0,44	0,26
4	Planowanie projektu (project plans)	200	500	1500	0,66	0,26	0,09
5	Projektowanie początkowe(initial design)	50	175	400	2,64	0,75	0,33
6	Projekt szczegółowy (detailed design)	25	150	300	5,28	0,88	0,44
7	Przeglądy projektu (design reviews)	75	225	400	1,76	0,59	0,33
8	Kodowanie (coding)	15	50	200	8,80	2,64	0,66
9	Pozyskanie el. powtarzalnych (reuse asquistion)	400	600	2000	0,33	0,22	0,07
10	Zakup pakietów (package purchase)	350	400	1500	0,38	0,33	0,09
11	Inspekcje kodu (code inspections)	75	150	300	1,76	0,88	0,44
12	Niezależne weryfikacja i walidacja (independent V & V)	75	125	200	1,76	1,06	0,66
13	Zarządzanie konfiguracją (configuration management)	1000	1750	3000	0,13	0,08	0,04

[7] MOD, to wartość modalna, określająca położenie najwyższego punktu krzywej rozkładu β

14	Formalna integracja (formal integration)	150	250	500	0,88	0,53	0,26

Tabela 1 cd. Średnia produktywność wyrażona w punktach funkcyjnych na osobomiesiąc i ilość godzin potrzebnych na wyprodukowanie jednego punktu funkcyjnego dla działań podejmowanych w różnych fazach projektu

15	Dokumentacja użytkownika (user documentation)	20	70	100	6,60	1,89	1,32
16	Testowanie jednostek (function testing)	25	150	300	1,89	0,88	0,33
17	Testowanie funkcjonalne (function testing)	25	150	300	5,28	0,88	0,44
18	Testowanie integracyjne (integration testing)	75	175	400	1,76	0,75	0,33
19	Testowanie systemowe (system testing)	100	200	500	1,32	0,66	0,26
20	Testowanie instalacji (field testing)	75	225	500	1,76	0,59	0,26
21	Testowanie akceptacyjne (acceptace testing)	75	350	600	1,76	0,38	0,22
22	Testowanie niezależne – beta testowanie (independent testing)	100	200	300	1,32	0,66	0,44
23	Zapewnienie jakości (quality assurance)	30	150	300	4,40	0,88	0,44
24	Instalacja/Szkolenie (installation/training)	150	350	600	0,88	0,38	0,22
25	Zarządzanie projektem (project management)	15	100	200	8,80	1,32	0,66
	Wartości kumulowane	1,90	6,75	13,88	69,38	19,55	9,51
	Średnie arytmetyczne	133	284,8	624	2,78	0,78	0,38

Źródło: na podstawie [34]

Biorąc za podstawę rozmiar systemu informatycznego wyrażony w punktach funkcyjnych Jones Capers podzielił systemy na następujące, uwzględnione w tabeli klasy wielkości:

Tabela 14. Klasa inwestycji informatycznej w zależności od jej rozmiaru wyrażonego w PF

Rozmiar systemu w PF	Przeliczenie rozmiaru w PF na rozmiar w liniach kodu języka „C"	Opis systemu	Klasa inwestycji informatycznej
1 PF	125 C	pomniejsze osobiste aplikacje i małe programy stanowiące podstawę do rozbudowy na większe; *harmonogram realizacji dla takich programów to zwykle okres od 1 dnia do tygodnia*	małe programy

Tabela 14 cd. Klasa inwestycji informatycznej w zależności od jej rozmiaru wyrażonego w PF

10 PF	1250 C	typowa wielkość aplikacji dla użytkownika końcowego oraz tymczasowych aplikacji stworzonych z zamiarem integracji z istniejącym oprogramowaniem w organizacji; *harmonogram dla programów o takim rozmiarze wynosi zwykle mniej niż jeden miesiąc*	aplikacje użytkownika
100 PF	12 500 C	aplikacje tej wielkość, to praktycznie górna granica dla programów użytkowych; w tej grupie aplikacji przeważają programy dopisywane do istniejącego w firmie oprogramowania; odnotowano mało samodzielnych programów takiego rozmiaru w 1998 r. w USA; natomiast w 1988 roku aplikacje takie były stosunkowo liczne dla środowiska DOS – u, jak.	duże aplikacje użytkownika

		np. wczesne interpretery BASIC – a; *harmonogram obejmuje okres mniejszy niż 6 miesięcy; indywidualni programiści mogą stworzyć takie aplikacje, chociaż inni specjaliści mogą być również angażowani*	
1000 PF	125 000 C	rozmiar tych programów odpowiada aplikacjom komercyjnym i wielu wewnętrznym programom środowiska WINDOWS ale również jest to typowy rozmiar dla wewnętrznych aplikacji klient – serwer; Harmonogram jest w tym przepadku dłuższy niż 12 miesięcy; dla tej grupy programów opracowanie specyfikacji wymagań i dokumentacji użytkownika ma znaczący udział w koszcie tego oprogramowania; kontrola jakości jest również elementem zasadniczym dla takich aplikacji; indywidualni programiści nie są w stanie samodzielnie stworzyć tak dużych programów; zalecana liczebność zespołu projektowego to do 10 osób; specjaliści tacy jak administratorzy baz danych, programiści techniczni (w określonym języku) czy osoby odpowiedzialne za zagwarantowanie jakości powinni być planowani w profilu zatrudnienia podczas realizacji projektów; wraz ze wzrostem liczebności zespołu wzrasta złożoność	programy komercyjne

		komunikacji pomiędzy jego członkami	
10 000 FP	1 250 000 C	aplikacje tej wielkości są zwykle nazywane jako systemowe, ponieważ są one za duże jako indywidualne programy, budowa takich programów często implikuje problemy, powodujące przekraczanie budżetu czy harmonogramu, czy nawet wstrzymanie przedsięwzięcia, liczebność zespołu projektowego oscyluje wokół liczby 100; harmonogram zakłada realizację projektu w czasie od 3 do 5 lat; praca „papierowa" jest tak duża, że produkcja dokumentów jest niekiedy kosztowniejsza niż programowanie; ze względu na wzrost liczby defektów wraz ze wzrostem aplikacji, formalna kontrola jakości obejmująca testy wstępne jest niezbędna, zarządzanie konfiguracją i zmianami jest również wymagane	systemy informatyczne
100 000 FP	12 500 000 C	aplikacje takiego rozmiaru znajdują się pośród najtrudniejszych przedsięwzięć minionego wieku; odpowiadają one swoim rozmiarem produktom takim jak Microsoft Windows 95 czy systemowi operacyjnemu MVS firmy IBM; jest to również rozmiar odpowiadający wielkości informatycznych wojskowych systemów; harmonogram tutaj to	duże systemy informatyczne

		zwykle od 5 do 8 lat; zespól projektowy liczy sobie setki osób, działających często w dużym rozproszeniu terytorialnym; problem zarządzania komunikacją osiąga bardzo duże rozmiary; zużycie papieru i liczba defektów mogą pociągać ogromne koszty; formalne zarządzanie konfiguracją i zmianami jest wymagane i kosztowne	

Źródło: opracowanie na podstawie [2]

W tabelach 2, 3 zamieszczono dodatkowo podział systemów informatycznych pod względem obszarów zastosowania. Z danych zamieszczonych w tabeli 2 wynika, że systemy powyżej 10 000 punktów funkcyjnych; duże i skomplikowane, wykonywane przez zespół własny użytkownika nie mają żadnych szans na powodzenie, ponieważ zostały w 100% wstrzymane.

Tabela 2 Średnia liczebność zespołów (w USA) realizujących projekty informatyczne różnych typów i o różnym rozmiarze wyrażonym w punktach funkcyjnych

Rozmiar w PF	Typ oprogramowania						
	Użytko-wnika	MIS	Kontra-ktowe	Kome-rcyjne	Syste-mowe	Wojskowe	Średnio
1 PF	0,25	0,50	0,50	0,50	0,60	0,75	0,52
10 PF	0,50	1,00	1,00	1,25	1,50	1,75	1,17
100 PF	1,00	1,25	1,15	1,75	2,00	3,00	1,69
1000 PF	-	9,00	8,00	10,00	11,00	15,00	10,60
10 000 PF	-	70,00	65,00	80,00	90,00	120,00	85,00
100 000 PF	-	750,00	700,00	775,00	900,00	1300,00	885,00

Źródło: na podstawie [5]

Znając średnie liczebności zespołów (z tabeli powyżej) można wyznaczyć średni czas realizacji projektów informatycznych w miesiącach, w zależności od typu aplikacji. Z danych zamieszczonych w tabeli 2 można wysunąć wniosek, że systemy powyżej 10 000 punktów funkcyjnych realizowane przez zespoły zewnętrzne mają już większą szansę powodzenia. Przedstawione dane, w przypadku dużych systemów wykonywanych na potrzeby komercyjne, wskazują na bardzo duże ryzyko niepowodzenia inwestycji tego typu.

Tabela 3 Średni czas, w miesiącach, realizacji projektów informatycznych w zależności od wielkości przedsięwzięcia wyrażonej w punktach funkcyjnych

Rozmiar w PF	Typ oprogramowania						
	Użytko-wnika	MIS	Kontra-ktowe	Kome-rcyjne	Syste-mowe	Wojskowe	Średnio
1 PF	0,05	0,10	0,10	0,20	0,20	0,30	0,16
10 PF	0,50	0,75	0,90	1,00	1,25	2,00	1,07
100 PF	3,50	9,00	9,50	11,00	12,00	15,00	10,00
1000 PF	-	24,00	22,00	24,00	28,00	38,00	27,20
10 000 PF	-	48,00	44,00	46,00	47,00	64,00	49,80
100 000 PF	-	72,00	68,00	66,00	78,00	85,00	73,80
ŚREDNIO	0,68	25,64	24,08	24,70	27,74	34,05	27,00

Źródło: na podstawie [5]

Z kolei tabela 4[8], tzw. „wymieralności projektów" przedstawia statystykę powodzenia (lub niepowodzenia) projektów w zależności od rozmiaru wyrażonego w punktach funkcyjnych. Może być ona, zdaniem autorki, używana do szacowania ryzyka podejmowania przedsięwzięć informatycznych, a co za tym idzie, dokonywania niezbędnych ustaleń z klientem, co do czasu trwania przedsięwzięcia, bezpiecznego marginesu kosztów itp. . Analizując informacje z tej tabeli można przypuszczać, że małe projekty kończą się z sukcesem w przeważającej liczbie przypadków, ale ryzyko i prawdopodobieństwo wstrzymania przedsięwzięcia lub jego opóźnienia wzrasta wraz ze wzrostem rozmiaru aplikacji. W rzeczywistości, realizacja systemów informatycznych o wielkości 10 000 punktów funkcyjnych jest jednym z najbardziej ryzykownych przedsięwzięć we współczesnym świecie. Przyjmując jako miarę złożoności systemu punkty funkcyjne klient powinien wielkość z zakresu 1000 – 1500 PF postrzegać jako próg złożoności aplikacji dających się stosunkowo bezpiecznie realizować. Fakt występowania takiej bariery nie należy jednak interpretować jako zakaz realizacji dużych systemów informatycznych. Należy jednak tak planować duże projekty, aby część systemu jednorazowo realizowana i przekazywana do eksploatacji nie przekraczała 1500 punktów funkcyjnych. Stan taki można uzyskać dzieląc odpowiednio duży system informatyczny i obarczony z tego względu dużym prawdopodobieństwem niepowodzenia na podsystemy realizowane w oddzielnych projektach, bazujące na tej samej koncepcji, opracowanej na etapie analizy i przygotowania założeń.

[8] Dane zamieszczone w tabeli są wynikiem analizy 6700projektów informatycznych realizowanych na całym świecie

Tabela 4. Tabela ukazująca procentowy rozkład prawdopodobieństwa zakończenia projektu w zależności od wielkości projektu wyrażonej w PF

Wielkość projektu w PF	Charakterystyka projektu	Wcześniej niż planowano	Zgodnie z harmonogramem	Opóźniony	Wstrzymany
1 PF	Programy małe	14,68%	83,16%	1,92%	0,25%
10 PF	Aplikacje użytkownika	11,08%	81,25%	5,67%	2,00%
100 PF	Duże aplikacje użytkownika	6,06%	74,77%	11,83%	7,33%
1000 PF	Programy tworzone przez firmy	1,24%	60,76%	17,67%	20,33%
10 000 PF	Systemy informatyczne	0,14%	28,03%	23,83%	48,00%
100 000 PF	Duże systemy informatyczne	0%	13,67%	21,33%	65,00%
ŚREDNIO		5,53%	56,94%	13,71%	23,82%

Źródło: na podstawie [4]

Kolejna tabela 5 pokazuje rozbieżność pomiędzy estymowaną a rzeczywistą wielkością projektu wyrażoną w punktach funkcyjnych. Rozbieżności te rosną wyraźnie z rozmiarem projektu. Powodów tych różnic należy upatrywać w:

- błędach estymacji, wynikających przede wszystkim z nadmiernego optymizmu lub względów politycznych danej firmy,
- rosnących wymagań w trakcie projektu ze względu na zwiększającą się ilość zmian.

Tabela 5. Punkty funkcyjne a defekty w poszczególnych etapach realizacji projektu informatycznego

Pochodzenie defektu	Potencjalny poziom wystąpienia/1PF	Sprawność usunięcia defektu	Defekty pozostałe w systemie
Wymagania	1,00	77%	0,23
Projektowanie	1,25	85%	0,19
Kodowanie	1,75	95%	0,09
Dokumentacja	0,60	80%	0,12
Błędy w poprawkach	0,40	70%	0,12
RAZEM	**5,00**	**85%**	**0,75**

Źródło: na podstawie [5]

Tabela 6 przedstawia statystyki prawdopodobieństwa ukończenia projektów, w zależności od ich wielkości wyrażonej w punktach funkcyjnych, z rozbieżnościami większymi niż 25% w stosunku do harmonogramu. Przy czy, podając za Capers Jones ([5]) 5% „poślizg" jest zauważalny, większy niż 10% jest boleśnie kosztowny, a 50% należy uważać za katastrofę. Wraz ze wzrostem wielkości projektu opóźnienia w stosunku do harmonogramu i w ogóle wstrzymania realizacji przedsięwzięć są coraz powszechniejsze. Czynnik zarządzania komunikacją, zarówno w stosunku do opóźnień i wstrzymań (cancel) projektów rośnie wraz ze wzrostem wymagań użytkownika. Średni wzrost nieplanowanych, niewyspecyfikowanych wymagań użytkownika wynosi około 1% do 2% na miesiąc podczas fazy projektowania i kodowania typowego projektu informatycznego. Rzeczywisty zakres zmian w wymaganiach użytkownika może jednak wzrosnąć aż do 10% w przeciągu pojedynczego miesiąca.

Tabela 6. Prawdopodobieństwo „poślizgu" większego niż 25% w stosunku do harmonogramu w zależności od wielkości projektu w punktach funkcyjnych

Rozmiar w PF	Typ oprogramowania						
	Użytko-wnika końcowego	MIS	Kontra-ktowe	Kome-rcyjne	Syste-mowe	Wojsko-we	Średnio
1 PF	4,00%	1,90%	98,00%	0,90%	0,90%	1,90%	1,92%
10 PF	20,00%	2,00%	97,00%	0,00%	3,00%	5,00%	5,67%
100 PF	35,00%	6.00%	88,00%	6,00%	7,00%	9,00%	11,83%
1000 PF	30,00%	12,00%	74,00%	16,00%	13,00%	20,00%	17,67%
10 000 PF	0,00%	13,00%	47,00%	20,00%	21,00%	29,00%	11,20%
100 000 PF	0,00%	31,00%	24,00%	20,00%	32,00%	30,00%	16,60%
Średnio	14,83%	10,98%	71,33%	10,48%	12,82%	15,82%	10,81%

Źródło: na podstawie [5]

Jedną z najnowszych metryk punktów funkcyjnych jest tabela 7, uwzględniająca kompletność różnego rodzaju narzędzi, technik, metod, które mogą być używane do zarządzania projektami informatycznymi. Tabela ta ukazuje kilka podstawowych różnic pomiędzy projektami informatycznymi zakończonymi z sukcesem i tymi w realizacji, których zdarzyły się niepowodzenia.. Podaje się ([5]), że najlepsze w branży informatycznej organizacje wykorzystują narzędzia o dziesięciokrotnie lepszej jakości i trzydzieści razy lepsze narzędzia do zarządzania projektami, aniżeli te, które do tej grupy nie należą. Firmy softwarowe odnoszące sukcesy jak i te, którym się to nie udaje wykorzystują narzędzia, których wielkość oscyluje w zakresie od 30 000 do 50 000 punktów funkcyjnych. Dane zawarte w tabeli 7 pokazują, że stosunek liczby używanych narzędzi w projektach wiodących i opóźnionych wyraża się ułamkiem $6/1$. Projekty zakończone z sukcesem wykazują tendencję do zastosowania w nich 16 razy więcej narzędzi, metod i

technik zarządzania (wyrażonych w punktach funkcyjnych) niż przedsięwzięcia opóźnione. Wykorzystywanie lub nie zestawu narzędzi do zarządzania projektami nie jest samo w sobie głównym czynnikiem różnic pomiędzy jednymi i drugimi projektami. *Kierownicy wiodących projektów, wykorzystujący duży zestaw narzędzi są po prostu lepiej przygotowani do rozpoznawania złożoności (rozmiaru) projektu i posiadają większą umiejętność doboru odpowiednich metod, technik czy narzędzi wspomagających zarządzanie.*

Tabela 7. Liczba i zakres stosowania narzędzi, metod i technik project management –u (zakres wyrażony w punktach funkcyjnych) w opóźnionych i wiodących przedsięwzięciach

Lp.	Narzędzia, metody i techniki zarządzania projektem informatycznym	Projekty opóźnione	ŚREDNIO	Projekty wiodące (zrealizowane z sukcesem)
1	Planowanie	1000	1250	3000
2	Estymacja kosztów			3000
3	Analiza statystyczna			3000
4	Metodologia zarządzania		750	3000
5	Analiza 2000 – go roku			2000
6	Zarządzanie jakością			2000
7	Assessment support		500	2000
8	Rozmiar projektu			1750
9	Analiza portfolio			1500
10	Zarządzanie ryzykiem			1500
11	Analiza obciążenia zasobów	300	750	1500
12	Zarządzanie kosztami		350	1250
13	Raportowanie rozbieżności kosztów		500	1000
14	Personnel support	500	500	750
15	Analiza trendu kamieni milowych		250	750
16	Budget support		250	750
17	Analiza punktów funkcyjnych		250	750
18	Przeliczanie LOC na punkty funkcyjne			750
	Ogółem punkty funkcyjne	1800	5350	30250
	Liczba zastosowanych narzędzi	3	10	18

Źródło: na podstawie [5]

3 Statystyki czynników niezwiązanych z wielkością projektu ograniczających ryzyko niepowodzenia przedsięwzięcia wdrożeniowego ZSI

Ważnym czynnikiem, który wpływa na efekt z jakim projekt zostanie zakończony jest poziom doświadczenia zespołu realizatorów, którymi są kierownicy projektu (po stronie klienta i dostawcy), wykonawcy i konsultanci. Typ ryzyka związany z doświadczeniem zespołu realizatorów nazywany jest „socjologią" projektu (tab. 8). Zależy ono od doświadczenia kierowników projektu (doświadczeni – niedoświadczeni), klienta (zwłaszcza komitetu sterującego), kwalifikacji zawodowych członków zespołu projektowego, doświadczenia zespołów technicznych ([8]).

Tabela 8. „Socjologia" projektu a prawdopodobieństwo sukcesu projektu

Socjologia projektu		Stan realizacji projektu			
dobra	zła	przed terminem	w terminie	opóźnione	zaniechane
tak	nie	18%	79%	20%	10%
nie	tak	0%	10%	45%	45%

Źródło: na podstawie [8]

Do czynników technicznych ograniczających ryzyko niepowodzenie projektu zalicza się:

- dobór zastosowanej technologii informatycznej do specyfiki projektu (efektywna – nieefektywna),
- dobór narzędzi programistycznych (adekwatne – nieadekwatne),
- możliwość ponownego wykorzystania komponentów (duża – mała),
- nadzór nad wymaganiami i zakresem projektu (stabilność wymagań – „pełzanie" [8]).

Tabela 9. Zastosowana technologia a prawdopodobieństwo sukcesu projektu

Technologia projektu		Stan realizacji projektu			
efektywna	nieefektywna	przed terminem	w terminie	opóźnione	zaniechane
Tak	nie	15%	62%	22%	10%
Nie	tak	0%	10%	75%	15%

Źródło: na podstawie [8]

Nie tylko wielkość projektu ma decydujący wpływ na *powodzenia realizacji przedsięwzięcia informatycznego, bardzo istotny jest tutaj również poziom i profesjonalizm zarządzania* (tab. 10). Czynnik ten obejmuje stosowane metodyki, metody oraz narzędzia wspomagające zarządzanie, a także zagadnienia z zakresu zapewnienia i kontroli jakości.

O profesjonalizmie zarządzania projektem przesądzają: szacowanie pracochłonności projektu (ręczne - automatyczne), planowanie (ręczne -

automatyczne), monitorowanie stopnia zaawansowania prac (formalne – nieformalne), określanie nakładów na zapewnienie i kontrolę jakości (minimalne – optymalne).

Tabela 10. Poziom i profesjonalizm zarządzania a prawdopodobieństwo sukcesu projektu

Zarządzanie projektem		Stan realizacji projektu			
dobre	złe	przed terminem	w terminie	opóźnione	zaniechane
Tak	nie	19%	78%	20%	10%
Nie	tak	0%	15%	45%	40%

Źródło: na podstawie [8]

Wnioski

Mając na uwadze znaczenie metody punktów funkcyjnych w ocenie ryzyka projektów informatycznych oraz brak w dostępnej, krajowej literaturze informacji na temat praktycznego zastosowania tej metody w szacowaniu rozmiarów systemów informatycznych, autorka zaimplementowała tę metodę w unikalnym studium wykonalności SI proponowanym w Metodyce Rezultatów Kontrolnych.

Stworzone, dwuetapowe studium wykonalności badanego systemu informatycznego jednoznacznie odróżnia zaproponowaną Metodykę Rezultatów Kontrolnych od innych metodyk przedstawianych w krajowej i zagranicznej literaturze przedmiotu. Studium to w pierwszym etapie oferuje bowiem, w efekcie skorzystania z autorskich programów, uzyskanie najważniejszych informacji o:

- wielkości systemu na miarę potrzeb klienta,
- szacunkowym czasie trwania projektu,
- liczebności zespołów projektowych,
- wartości niezbędnego budżetu na realizację przedsięwzięcia.

Ze względu na fakt, iż każde wdrożenie ZSI cechuje pewna specyfika, wynikająca m.in. z funkcjonowania obiektu w określonym otoczeniu, jego doświadczeń w zakresie stosowania technologii informatycznej, wykorzystywanej infrastruktury sprzętowej, zaangażowania użytkowników systemu i kierownictwa informatyzowanej firmy, *nie jest możliwe statyczne zastosowanie proponowanej metodyki zarządzania takim projektem w różnych jednostkach*, ale zawsze można określić pewną sekwencję etapów i uszczegóławiać je o potrzebne rezultaty kontrolne i zadania.

Autorka uważa, że na sukces z pozoru podobnych projektów mają wpływ stosowane metodyki, techniki i metody szczegółowe oraz wspomaganie informatycznymi systemami do zarządzania przedsięwzięciami. Stąd też w artykule proponowana jest Metodyka Rezultatów Kontrolnych, od której zastosowania oczekuje się zmniejszenia ryzyka nieosiągnięcia dodatnich efektów wynikających z wdrożenia ZSI w konkretnej firmie.

Literatura

1. Kaczorowska A.: Efektywność wdrażania systemów informatycznych kasy EMS na przykładzie krajowych zakładów energetycznych. Rozprawa doktorska, Łódź, 2003.
2. Guidelines for Software Measurement. IFPUG, 1994.
3. Jones C.: Applied Software Measurement – Assuring Productivity and Quality. McGraw Hill, New York, 1996.
4. Jones C.: Patterns of Software Systems Failure and Success. International Thomson Computer Press, 1996.
5. Jones C.: Project Management Tools and Software Failures and Successes. CROSSTALK The Journal of Defense Software Engineering, luty 1998, str. 13 – 17.
6. Jones C.: Revitalizing Software Projects. American Programmer, June 1994.
7. Adamczewski P.: Ku systemom zintegrowanym klasy MRP II. [w:] Kubiak B., Korowicki A. . Human Computer Interaction 2001. Wyd. Akwila – Roman Młodkowski, Gdańsk, 2001.
8. Adamczewski P.: Wdrażanie systemu zintegrowanego jako złożone przedsięwzięcie informatyczne. Informatyka, Wydanie specjalne, 1998.

ROZDZIAŁ IX

SZCZYPTA PSYCHOLOGII DLA KIEROWNIKÓW PROJEKTÓW

Ewa MIZERSKA

Do planowania projektu podchodzimy zwykle zgodnie z metodyką. Po pierwsze musi być sformułowany cel projektu, dobrze zdefiniowany produkt i kryteria odbioru/sukcesu oraz podstawowe ramy (czas i koszty). Można wtedy przystąpić do harmonogramowania, w efekcie czego otrzymujemy (między innymi) plan zasobów: liczbę potrzebnych pracowników, ich przybliżone specjalności (architekt, programista, analityk, ekspert w danej dziedzinie, itp.), wreszcie zadania dla każdego z nich. Zebranie odpowiednich ludzi i przydzielenie im zadań, wydaje się wszystkim, co należy uczynić, aby projekt pomyślnie wystartował i szczęśliwie dobiegł końca.

Uważamy za naturalne i oczywiste, że ludzie będą pracować z entuzjazmem, będą żądni sukcesu, w współpraca układać się będzie zgodnie i bezproblemowo.

Zdajemy sobie bowiem sprawę, że od postawy pracowników, ich zaangażowania, a także zadowolenia z pracy zależy w efekcie jakość produktu i ostateczny sukces projektu. Jeśli coś w tej sferze szwankuje, skłonni jesteśmy uważać, że przyczyną są nieodpowiedni ludzie: niedostatecznie wykwalifikowani, sfrustrowani, albo po prostu z natury kłótliwi. Czy słusznie?

Rzeczywiście, badania potwierdzają, że zadowolony z pracy pracownik częściej i łatwiej osiąga sukces, zarówno osobisty jak i zespołowy. Okazuje się jednak, że zaangażowanie i satysfakcja z pracy tylko w około 10-30% wiąże się z czynnikami osobowościowymi pracowników. Największa korelacja, aż 40-60%, występuje pomiędzy satysfakcją z pracy a szeroko pojętymi warunkami pracy, czy tez „czynnikami sytuacyjnymi". (Interakcja obu rodzajów czynników odpowiada za 10-20% wariancji). Najważniejszym z czynników warunkujących osiąganie zawodowej satysfakcji okazuje się autonomia wykonywania zadań, czyli zezwolenie na pewną samodzielność i zaufanie ze strony bezpośredniego przełożonego.

Szczególny negatywny wpływ ma konflikt bądź niejasność roli pracownika. O jaką rolę chodzi? Czy tylko tę, związana z kompetencjami, a więc rodzajem przydzielanych zadań? Okazuje się, że nie.

Mówiąc o roli trzeba także brać pod uwagę cechy osobowości danego pracownika, odnosząc je do cech (a więc i ról) pozostałych członków zespołu.

Paradoksalnie okazuje się, że pracownik mający wszystkie wymagane kwalifikacje, a przy tym bezkonfliktowy, świetny kolega – może być pierwszym który opuści nasz projekt, zwerbowany do ciekawszej lub lepiej płatnej pracy. Zadania, które mu stawialiśmy, nie były dla niego niczym nowym, nie stanowiły wyzwania, a więc powodowały wrażenie zawodowego zastoju.

Z kolei człowiek o wysokich kompetencjach, znający swoją wartość, jeżeli nie „dogada się" z resztą zespołu, może więcej zaszkodzić niż pomóc. Skupi się bardziej na demonstracyjnym udowadnianiu swojej niezbędności, niż na współpracy dla wspólnego sukcesu.

Najlepszymi pracownikami okazują się ludzie, z którymi „dobrze się współpracuje", „dopasowani do zespołu", jeżeli przy tym zadania stanowią dla nich okazję do rozwoju, zdobycia nowej wiedzy i umiejętności.

Czym więc jest owo dopasowanie do zespołu, skoro, jak widać, nie chodzi tu o rolę wynikającą wprost z kompetencji pracownika?

R. Meredith Balbin, autor książki „Twoja rola w zespole" wyróżnia 9 ról i twierdzi, że każda z nich musi być reprezentowana, aby zespół był rzeczywiście zrównoważony, w przez to efektywny. Nie znaczy to oczywiście, że zespoły powinny liczyć dokładnie po 9 osób. Role te mogą być, i często są, łączone. Ponadto, jak zauważymy za chwilę, zwykle wystarczy jeden Myśliciel, podczas gdy Realizatorów trzeba kilku, czasem kilkunastu.

Balbin nazywa role, omawia zalety osób w te role wchodzących, pomaga rozpoznać preferencje poprzez uważne wsłuchanie się w wypowiedzi ludzi. Omawia także wady i podpowiada, do jakiego stopnia można je tolerować, a kiedy niepożądane zachowania wychodzą poza granice dopuszczalności.

Rola	Zalety	Wady	Niedopuszczalne	Wypowiedzi
Myśliciel	twórczy, z wyobraźnią, postępowy, rozwiązuje trudne problemy.	Pomija szczegóły. Zbyt zajęty, by efektywnie; się porozumiewać.	Obstaje przy swoich indywidualnych pomysłach, podczas gdy współpraca mogłaby dać lepsze efekty	1. KLedy problem wydaje się nie do pokonania, podejdź do niego od innej strony. 2. Nie ma problemów nierozwiązywalnych. 3. Im większy problem, tym większe wyzwanie. 4. Nie przeszkadzaj, geniusz pracuje. 5. Dobre pomysły zawsze na początku wydają się dziwne. 6. Pomysły biorą się z marzeń. 7. Bez nieustannych innowacji nie ma przetrwania.
Poszukiwacz źródeł	komunikatywny entuzjastyczny komunikatywny. Bada możliwości, Nawiązuje kontakty	Zbył optymistyczny. Traci zainteresowanie z chwilą, gdy mija pierwszy entuzjazm.	Zapomina o zobowiązaniach, terminach	1. Możemy zbić na tym fortunę. 2. Pomysły należy kraść z dumą. 3. Nie wyważaj otwartych drzwi. 4. Okazja rodzi się na skutek błędu innej osoby. 5. Chyba możemy to wykorzystać? 6. Żeby się dowiedzieć, możesz zatelefonować. 7. Czas przeznaczony na rozpoznanie rzadko bywa stracony.
Koordynator:	dojrzały, pewny siebie, dobry przewodniczący. Określa cele, zachęca do podejmowania decyzji, trafnie przydziela zadania.	Może być postrzegany jako manipulator, W swojej pracy wyręcza się innymi.	Uważa, że jemu należą się pochwały za wysiłek całego zespołu	1. Nie traćmy sprzed oczu istoty sprawy. 2. Czy ktoś ma jeszcze coś do dodania? 3. Chcielibyśmy osiągnąć porozumienie, zanim pójdziemy dalej. 4. Nigdy nie zakładaj, że cisza oznacza aprobatę. 5. Uważam, że powinniśmy dać szansę komuś innemu. 6. Właściwe rozdzielenie obowiązków to prawdziwa sztuka. 7. Zarządzanie to sztuka nakłaniania innych do wykonania całej pracy.
Lokomotywa:	stawia przed zespołem wyzwania, dynamiczny. potrzebuje presji. Jest odważny i zdeterminowany w pokonywaniu przeszkód.	Może prowokować innych i ranić ich uczucia.	Nie umie ratować sytuacji dobrym humorem	1. Po prostu zrób to! 2. Powiedz "nie", a potem negocjuj. 3. Jeśli się tego podejmiesz, oczekuję, że to będzie zrobione. 4. Nie jestem zadowolony, że osiągamy wszystko, co jest w naszej mocy. 5. Może jestem nudny, ale przynajmniej rzeczowy. 6. Nadam bieg sprawom. 7. Kiedy wyrzucają cię przez drzwi, próbuj wejść oknem.

Krytyk Wartościujący:	rzeczowy, wnikliwy, ma talent strategiczny Dostrzega różnorodne opcje. Zdolny do obiektywnej oceny.	Mato energiczny, nie umie inspirować innych Nadmiernie krytyczny..	Popada w cynizm bez logicznego uzasadnienia	1. Przemyślę **to** i **jutro** poinformuję cię, jaka jest ostateczna decyzja. 2. Czy wyczerpaliśmy już wszystkie możliwości? 3. Jeśli to jest sprzeczne z logiką, nie warto się za to zabierać. 4. Lepiej jest pomału podejmować słuszną decyzję niż w pośpiechu podjąć błędną. 5. Po chwili namysłu uważam, że to najlepsza możliwość. 6. Rozważmy alternatywne rozwiązania. 7. Nie powinno się podejmować decyzji na fali czystego entuzjazmu.
Dusza Zespoiu:	współpracujący, łagodny, uważny i dyplomatyczny. Słucha, buduje, zapobiega tarciom, wprowadza spokój.	Niezdecydowany w nagłych sytuacjach. Ulegający wpływom.	Unika sytuacji związanych z presją	1. Uprzejmość nic nie kosztuje. 2. Bardzo mnie zainteresował twój punkt widzenia. 3. Jeśli tobie to odpowiada, ja też jestem z tego zadowolony. 4. Każdy ma jakąś dobrą stronę, do której można się odwołać. 5. Gdyby ludzie lepiej siebie słuchali, mniej by gadali. 6. Zawsze możesz odczuć w pracy dobrą atmosferę. 7. Staram się być wszechstronny.
Realizator:	zdyscyplinowany, godny zaufania, Konserwatywny : wydajny. Przekształca pomysły w działania.	Mato elastyczny. W nowych sytuacjach reaguje z opóźnieniem.	Utrudnia zmiany	1. Jeśli to jest wykonalne, zrobimy *to.* 2. Jeden gram działania jest wart tyle, co kilogram teorii. 3. Ciężka praca jeszcze nikogo nie zabiła. 4. Jeśli to jest trudne, zrobimy to natychmiast. Jeśli niemożliwe, zabierze nam to trochę więcej czasu. 5. Błądzić jest rzeczą ludzką, wybaczać nie mieści się w polityce firmy. 6. Zabierzmy się za *to,* co mamy zrobić. 7. Firma może liczyć na moje pełne poparcie.
Skrupulatny Wykonawca:	pracowity, sumienny, niespokojny. Szuka błędów i zaniedbań. Punktualny,	Ma skłonność do zbytniego zamartwienia się. Niechętnie przydziela zadania innym. Bywa drobiazgowy.	Ma skłonność do zachowań obsesyjnych	1. To coś, czemu musimy poświęcić całą naszą uwagę. 2. Zawsze trzeba czytać to, co jest napisane drobnym druczkiem. 3. "Jeśli coś może się nie udać, to się na pewno nie uda" i "Murphy był optymistą". 4. Nie ma żadnego usprawiedliwienia na to, że nie jest się doskonałym. 5. Perfekcja to coś normalnego. 6. Zrób dzisiaj to, co możesz zrobić jutro. 7. Czy to zostało sprawdzone?

Specjalista:	samodzielny, z inicjatywą. Potrafi wyznaczyć sobie jeden nadrzędny cel, skłonny do poświęceń. Posiada rzadko spotykaną wiedzę i umiejętności.	Działa w wąskim zakresie. Koncentruje się na szczegółach, nie dostrzega całościowego obrazu.	Ignoruje czynniki spoza obszaru swojej specjalności	1. W tej pracy nigdy nie przestajesz się uczyć. 2. Jeśli wybierzesz pracę, którą kochasz, nigdy już nie będziesz musiał pracować. 3. Prawdziwy profesjonalizm sam w sobie jest nagrodą. 4. Fascynuje mnie to, co robię. 5. Im więcej wiesz, tym więcej musisz się jeszcze dowiedzieć. 6. Lepiej jest wiedzieć wiele na jeden temat niż trochę na wiele tematów. 7. Komisja to dwanaście osób wykonujących pracę jednego człowieka.

Ludzie, którzy są członkami scementowanego zespołu, odczuwają dumę z tego powodu. Chętnie podkreślają ten fakt przy różnych nieformalnych okazjach. Czują, że przynależą do „lepszej klasy", „elity". To, co tworzą (produkt projektu) traktują jako swoją własność, budują tak, jakby budowali dla siebie. Zdarza się słyszeć od nich: „ten system to moje dziecko". Nie tylko samo zadanie, ale również praca w takim właśnie gronie osób daje im zadowolenie. Innymi słowy pracują nie tylko dla zleceniodawcy (który chce otrzymać produkt), nie tylko dla siebie samych (aby zrobić dobrą robotę, nauczyć się czegoś nowego i móc to wpisać do cv), ale także dla siebie nawzajem: koledzy czekają na moją cząstkę pracy, nie mogę ich zawieść.

To wszystko sprawia, że zespół, który osiągnie ten etap rozwoju raczej nie zmienia składu osobowego dopóki nie zakończy pracy – a zakończy z sukcesem.

Niestety, nie ma jednoznacznej recepty na uzyskanie tego, tak pożądanego, stanu rzeczy. Tom DeMarco, Timothy Lister piszą wprost: „Nie można zmusić ludzi, by stali się zgranym zespołem. Można mieć nadzieję, że tak się stanie. Można trzymać kciuki. Można działać tak, żeby zwiększyć szanse pomyślnego kształtowania zespołu. Ale nie można tego spowodować. Jest to zbyt delikatny proces, by można go było kontrolować."

W szczególności błędne jest założenie, że grupa osób natychmiast po utworzeniu i przyjęciu zadań do wykonania – zaczyna działać jako zespół w swoim ostatecznym kształcie. Zazwyczaj na początku to kierownik nie tylko wyznacza cel, ale także określa sposób funkcjonowania zespołu: formalną strukturę, obszary odpowiedzialności, zasady przydzielania zadań i odbioru produktów, plan komunikacji, itp.. Praktyka pokazuje, że te założenia pozostają w mocy dość krótko. Członkowie zespołu muszą się do siebie nawzajem dopasować, ustalić wzajemne stosunki i oczekiwania. Muszą także wypróbować w praktyce wszystkie formalne ustalenia, sprawdzając, jak daleko może sięgać „twórczy nieład". Dopiero po takim „okresie sztormowania" przyjmowane są – nieformalnie, zwyczajowo – normy i zasady, które rzeczywiście będą obowiązywać. Można je wtedy spisać, czyniąc zadość wymaganiu formalnego prowadzenia projektu.

Nie ma przepisu na stworzenie zgranego zespołu, natomiast łatwo można podać przepis na atmosferę antyzespołową, szereg czynników, które z pewnością utrudniają, a często wręcz uniemożliwiają pracę w pełni zespołową.

Najważniejszy z nich Tom DeMarco i Tim Lister nazywają defensywnym zarządzaniem. Chodzi o taką postawę kierownika, kiedy próbuje się on, w widoczny sposób, zabezpieczyć przed błędami popełnianymi przez pracowników. Jest to odczytywane jako brak zaufania i zakaz popełniania błędów, a więc wymaganie doskonałości. Dla znakomitej większości ludzi jest to zbyt duża presja, która w efekcie działa demotywacyjnie.

Rozdrobnienie obowiązków pracowników występuje wtedy, gdy jedna osoba pracuje równolegle nad kilkoma projektami. Z punktu widzenia szefa działu lub prezesa firmy bywa to uzasadnione. Jednak w oczywisty sposób obniża jakość i wydłuża czas pracy nad poszczególnymi zadaniami. Uniemożliwia także pełne wejście pracownika do zespołu, gdyż zawsze część jego czasu będzie poświęcana na inne sprawy. Podobnie działa nadmiar biurokracji.

Obniżanie jakości produktu może, moim zdaniem, mieć miejsce jedynie wtedy, gdy następuje w wyniku rzeczywiście niezależnych działań osób (grup, firm) niezwiązanych w żaden sposób z zespołem. Może to być rozszerzenie zakresu projektu bez zmiany czasu i kosztu, albo jakaś wyjątkowa zmiana w otoczeniu projektu.

Czynnikiem zdecydowanie demoralizującym jest wyznaczanie jawnie nierealnych terminów wykonania prac. Niestety, w warunkach zwanych „biznesowymi" zdarza się to dość często. Jawnie nierealny termin sprawia, że próby zachęcenia ludzi do szybszej, sprawniejszej pracy skazane będą na porażkę. Ludzie nie są robotami i będą o tym coraz natarczywiej przypominać.

Wreszcie drażliwą sprawą jest zakończenie pracy nad projektem. Czy równocześnie musi nastąpić rozwiązanie zespołu i dołączenie poszczególnych osób do różnych grup? W wielu firmach tak właśnie się dzieje. Jest to wręcz uznawane za właściwe funkcjonowanie tzw. organizacji macierzowej. Szkoda, bo potencjał zgranego zespołu mógłby być z powodzeniem wykorzystany w nowym projekcie. Jeżeli jednak nieodwołalnie trzeba zespół rozwiązać, pamiętajmy aby uczynić z tego małe święto. Trzeba ogłosić sukces wszem i wobec, głośno podziękować każdemu za jego wkład we wspólne zwycięstwo. Pomoże to przenieść część entuzjazmu i zapału do nowych zespołów.

Literatura

1. DeMarco T., Lister T., *Czynnik ludzki. Skuteczne przedsięwzięcia i wydajne zespoły.*
2. Edward Yourdon, *Marsz ku śmierci. Poradnik projektanta systemów.*
3. R. Meredith Balbin, *Twoja rola w zespole.*
4. Kossowska Małgorzata , *Ocena i rozwój umiejętności pracowniczych.*
5. Peter Malkin, Cary Cooper, Charles Cox, *Organizacje a kontrakt psychologiczny. Zarządzanie ludźmi w pracy.*
6. Magdalena Sękowska, Ewa Szymanowska, *Analiza transakcyjna w zarządzaniu.*

CZĘŚĆ 2

ROZDZIAŁ X

PROJECT SERVANT – ZINTEGROWANE ŚRODOWISKO WSPOMAGANIA WYTWARZANIA OPROGRAMOWANIA W FIRMIE INFORMATYCZNEJ

Marek SUJDAK, Jan WEREWKA, Dawid PYTKA

1. Wstęp

Wobec systemów zintegrowanych wspomagających zarządzanie wytwarzania oprogramowania można zdefiniować kilka nadrzędnych celi, są to:

- Utrzymywanie danych wspólnych dla wszystkich uczestników projektu. Uczestnicy projektu potrzebują centralnego repozytorium, gdzie można uzyskać lub odszukać informację projektową, cele projektowe, aktualny zakres prac do wykonania, status projektu
- Ujednolicenie podstawowych procesów zarządzania projektami informatycznymi.
- Gromadzenie danych o projektach dla wyznaczenia aktualnych stanów projektów oraz dla dalszych potrzeb analizy.
- Określenie ram do realizacji projektów. System zintegrowany powinien pełnić rolę szkieletu (ang. framework) dla projektów.
- Pełnienie roli biblioteki procesów. Wypełnienie luki pomiędzy zdefiniowaniem procesu a jego zastosowaniem w projekcie.

Geneza powstania PSV jest podobna tak jak w przypadku powstawania innych narzędzi. Poszukiwano prostego narzędzia do planowania i kontroli czasu projektu. Szeroko przeprowadzona analiza wykazała, że nie było takiego narzędzia spełniającego następujące kryteria:

- Wspólny dostęp do danych dla wszystkich uczestników projektu.
- Możliwość zbierania danych wspomagających kontrolę pracy.
- Niski koszt w przeliczeniu na użytkownika.
- Prostota użytkowania systemu, zrozumiała dla osób nie uczestniczących bezpośrednio w projektach.
- Możliwość łatwej integracji z innymi systemami.

Na bazie tych potrzeb powstał system zwany Time Servant [5], który był istotnym udogodnieniem wspomagającym realizacje projektów. Postanowiono, ze system ten ma wspomagać realizacje projektów w szerszym zakresie, a proponowane rozwiązania mają wspomagać uznane standardy. W tym celu dokonano analiz możliwych rozwiązań [4,6].

Prezentowany system znalazł zastosowanie w firmach informatycznych, tym nie mniej mógłby być on także interesujący dla firm realizujących inne procesy wytwórcze.

2. Struktura systemu, opis systemu

Project Servant jest narzędziem, które z założenia wspiera cały cykl życia projektu w wielu dziedzinach:

Planowanie. Umożliwia harmonogramowanie faz, iteracji i zadań oraz zarządzanie, modyfikowanie i analizę planów projektowych. Pomocne w tym są dostępne różne widoki na plan projektu. Możliwe jest oglądanie planu z widokiem na fazy, iteracje i zadania (ang. WBS), w tym przypadku mamy informacje na temat ich statusów, terminów wykonania, priorytetów oraz celów i założeń wstępnych. W innym widoku możemy spojrzeć na plan projektu pod względem zaangażowania zespołu projektowego w poszczególnych etapach realizacji. Dostępne są widoki, które dostarczają informacji, kiedy zadanie było lub będzie realizowane i ile czasu zaplanowano, a ile czasu zajęło wykonanie zadania oraz kto uczestniczył i kto odpowiadał za wykonanie zadania. Do widoków tych dodano filtry, które ułatwiają pobranie istotnych informacji.

Produkt. System dostarcza wsparcie dla rozwoju i cyklu życia produktu. Pozwala na zarządzanie wersjami oraz komponentami produktu. Z wersjami mogą być związane sprawy projektowe takie jak wykryte błędy lub wymagania, które należy zaimplementować.

Zarządzanie wymaganiami. Istnieje możliwość dodawania, usuwania, modyfikowania wymagań projektowych. Z wymaganiem związanie są zadania, jakie należy wykonać oraz jest ono powiązane z wersją produktu, w której powinno być zaimplementowane.

Zarządzanie ryzykiem. System pozwala na dodawanie i zarządzanie wykrytymi ryzykami projektowymi. Z ryzykami możemy powiązać akcje, jakie należy wykonać w celu ich złagodzenia.

Zarządzanie błędami. Testerzy i developerzy mogą raportować wykryte w produkcie błędy, przypisując je do wersji, w której zostały znalezione i w której powinny być usunięte. Z błędami mogą być powiązane zadania jakie należy wykonać w celu ich usunięcia.

Zarządzanie dokumentami. System pozwala na przechowywanie dokumentów projektowych w formie plików lub linków, które mogą być związane z zadaniami, wymaganiami, błędami lub ryzykami. To ułatwia sprawne przeszukiwanie i zarządzanie nimi.

Uczestnicy. Project Servant pozwala na definiowanie grup projektowych, które mogą być podstawą organizacyjną firmy, z przypisanymi do nich użytkownikami systemu. W projekcie natomiast określa się team projektu wybierając użytkowników ze zdefiniowanych grup określając ich role w projekcie i stopień zaangażowania. Uczestnicy logują wykonywaną przez nich pracc w projekcie.

Opracowany został również mechanizm powiadamiania na zdarzenia projektowe. Pozwana on na automatyczne wysyłanie informacji o dodaniu, modyfikacji zadań, błędów, wymagań lub ryzyk do osób zainteresowanych oraz innych zdarzeniach. Rodzaj wysyłanych zdarzeń jest elementem konfiguracji projektu.

Project Servant jest zintegrowany z narzędziami do planowanie,

definiowania procesów, środowiskami developerskimi, systemami do zarządzania kwestiami, konfiguracją oraz środowiskami do analizy i projektowania. Wtyczki do EA oraz Eclipse są w fazie wersji beta, wtyczka do MS Project'a została zakończona. Planowany jest integracja z wszystkimi narzędziami istotnymi w procesie wytwarzania i utrzymywania oprogramowania.

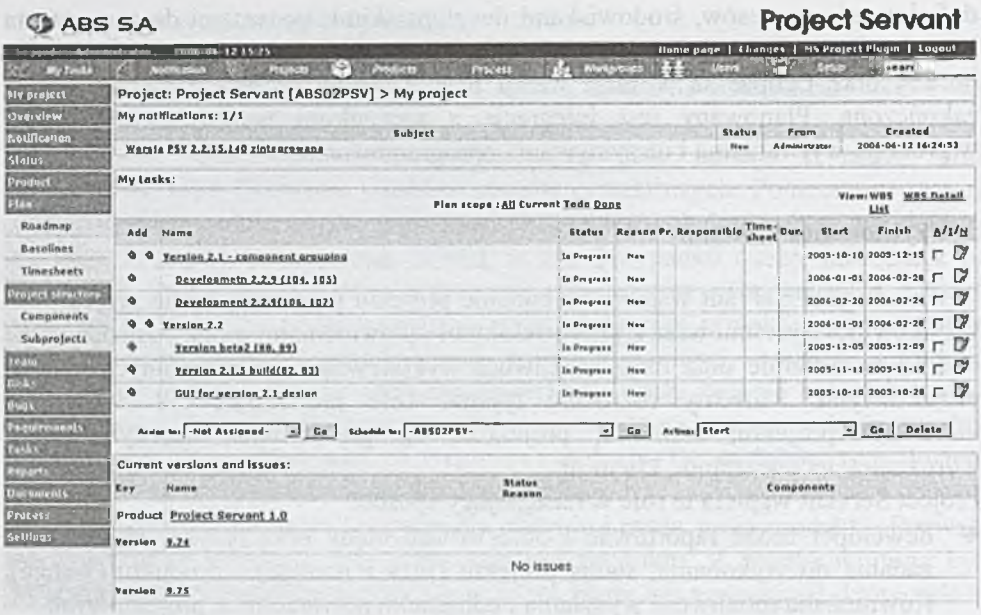
3. Użytkownicy systemu

Project Servant wspiera wykonanie projektu przez różne role projektowe. Użytkownicy systemu wraz z przydzielonymi uprawnieniami mogą posiadać różne widoki w systemie oraz mają możliwość wykonywania poszczególnych działań projektowych. Możemy rozróżnić typowe role projektowe jak deweloper, kierownik projektu, architekt, projektant oprogramowania, analityk, tester, wdrożeniowiec, uczestnik, klient itp.
Project Servant wspiera te role w następujący sposób:

- deweloper może raportować i obserwować błędy oraz ryzyka, obserwować zadania do wykonania, status projektu (fazy i iteracje) i produktu (wersje). Również ma możliwość wysyłania i odbierania powiadomień projektowych.
- kierownik projektu planuje fazy, iteracje i zadania, obserwuje i analizuje plany i raporty. Zarządza produktami, wersjami oraz komponentami produktowymi i projektowymi. Tworzy i organizuje grupę projektową. Zarządza ryzykiem i błędami.
- architekt, projektant, analityk może definiować i zarządzać wymaganiami oraz komponentami produktowymi.
- tester może logować błędy do systemu wykryte w różnych wersjach produktu i obserwować ich status.
- klient, uczestnik ma możliwość obserwowania mapy drogowej i statusu projektu, wypuszczonych wersji oraz zbierania wymagań projektowych.
- Inne role takie jak np. księgowość na podstawie raportów może obliczać koszty oraz czas realizacji zadań.

Korzystając z mechanizmu komponentów uczestnicy mogą uzyskać spojrzenie na sprawy projektu z różnych perspektyw. Widok dyscyplinowy daje spojrzenie na charakter, stan oraz plany związane z poszczególnymi dyscyplinami. Przykładowo tester przy pomocy komponentu dyscyplinowego „Test" może analizować sprawy. Połączenie z komponentami funkcjonalnymi daje spojrzenie na testy w kontekście określonej grupy wymagań.

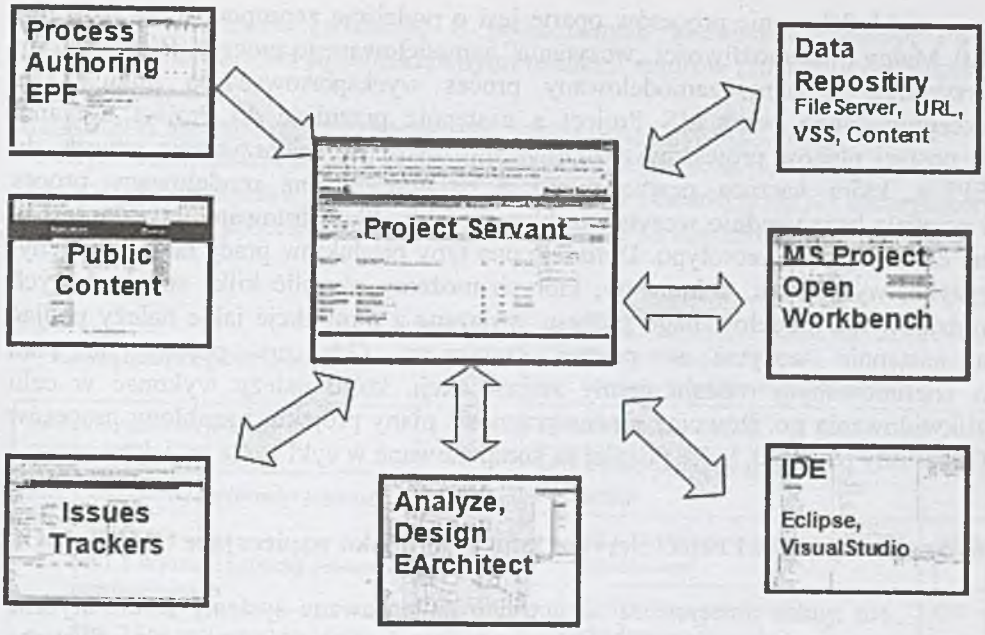
Komponenty pozwalają stworzyć wiele widoków projektowych, które są z kolei obiektem zainteresowań dla poszczególnych uczestników.



Rys. 1. Główna strona PSV

4. Integracja PSV z innymi narzędziami

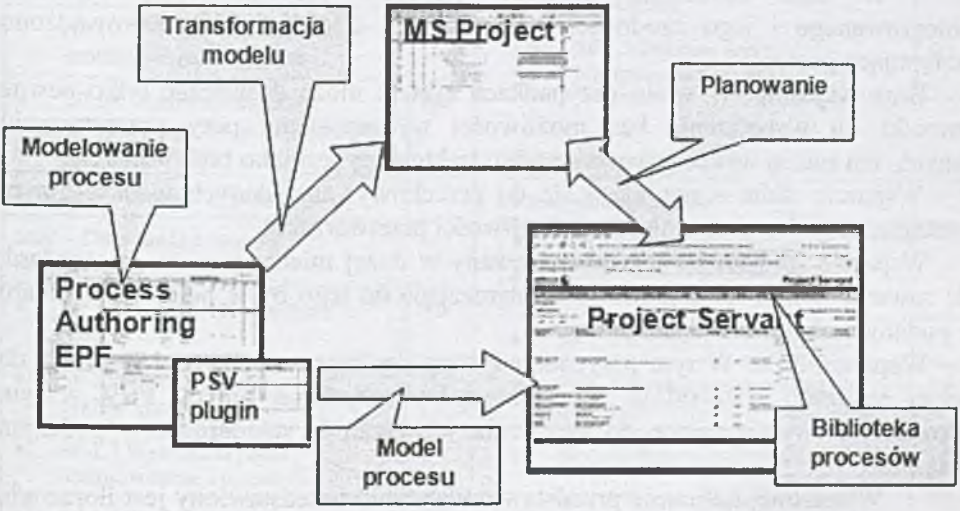
Project Servant jest zintegrowany z różnymi narzędziami. Współpraca z MS Projectem umożliwia wymianę informacji związanych z planowaniem, harmonogramowaniem projektu. W środowiskach deweloperskich takich jak Eclipse, Visual Studio dostępne są informacje na temat wykonywanych zadań, istnieje możliwość zmian ich statusów oraz raportowanie postępów prac. Enterprise Architect jest narzędziem, z którego mogą być pobierane wymagania projektowe w celu ich dalszej analizy. Możliwa jest również zbieranie informacji z innych systemów ogólnego przeznaczenia (ang. Issue Tracker) jak na przykład Jira. Z Project Servant'a możliwy jest dostęp do zasobów z danymi takich jak serwery plików, repozytoria, systemy zarządzania treścią. Project Servant może również współpracować z systemem do opisu procesów jakim jest Eclipse Proces Framework.



Rys. 2. Schemat integracji PSV z innymi narzędziami

Modelowanie, wdrażanie procesów

Istotną cechą Project Servant'a jest współpraca z systemem do modelowania procesów Eclipse Process Framework EPF.



Rys. 3. Zasady modelowania, wdrażania modeli do PSV

Modelowanie procesów oparte jest o podejście zaproponowane w SPEM 2.0. Mamy dwie możliwości „wczytania" zamodelowanego procesu. W pierwszym przypadku możemy zamodelowany proces wyeksportować do pliku XML akceptowalnego przez MS Project a następnie przenieść do Project Servanta w postaci planów projektowych. Druga możliwość to wykorzystanie wtyczki do EPF'a, która narzuca pewne reguły i ograniczenia na modelowany proces i pozwala bezpośrednio wczytać szablon projektu. Skonstruowana wtyczka jest w tej chwili w fazie prototypu. Definiuje ona typy produktów pracy takie jak błędy, ryzyka, wymagania, dokumenty, których możemy określić kilka standardowych rodzajów dla modelowanego procesu, związane z nimi akcje jakie należy podjąć a następnie wczytać w postaci szablonów. Gdy np. pojawi się błąd o zdefiniowanym rodzaju mamy zestaw akcji, które należy wykonać w celu zlikwidowania go. Również można przenosić plany projektu i szablony procesów (capability patterns), które później są komponowane w cykl życia projektu.

5. Analiza systemu Project Servant jako środowisko wspierające CMMI, ISO

Na rynku dostępne są różnorodne zintegrowane systemy informatyczne wspomagające procesy zarządzania wytwarzaniem oprogramowania. Systemy takie mogą być oceniane ze względu na:

- koszt produktu w przeliczeniu na stanowisko pracy,
- łatwość obsługi,
- realizowaną funkcjonalność.

W celu porównania potrzebnej funkcjonalności, a nie balastowej, która mało komu jest potrzebna, proponuje się porównanie podobnych narzędzi względem uznanych standardów, jakimi są CMMI i ISO.

W celu określenia poziomu wsparcia przedstawionego systemu zintegrowanego i jego zgodności z standardami CMMI i ISO, wprowadzono następujące poziomy:

0 – Brak wsparcia. W wielu przypadkach system może dostarczać tylko pewne formatki do wypełnienia bez możliwości wykorzystania przy przetwarzaniu danych, ten rodzaj wykorzystania systemu traktowany jest jako brak wsparcia.

1 – Wsparcie słabe – ogranicza się do przechowywania danych w określonym formacie, przy dosyć ograniczonej możliwości przetwarzania.

2 – Wsparcie średnie. System zintegrowany w dużej mierze wspomaga standard, ale zawarte w nim funkcje nie są wystarczające do tego by w pełni ten standard w podanym obszarze był spełniony.

3 – Wsparcie duże. W tym przypadku system dostarcza możliwości wsparcia do pełnej realizacji standardów. Prawidłowe wykorzystanie funkcji PSV w tym zakresie jest wystarczające do spełnienia wymagań w standardów w podanym obszarze.

W ostatniej kolumnie przedstawionych tabel przedstawiony jest iloraz a/b, gdzie a oznacza poziom wsparcia realizowany przez PSV, natomiast b jest możliwym poziomem wsparcia dostarczanym przez system zintegrowany. Różnica

poziomów a-b może świadczyć o potencjalnych kierunkach rozwoju PSV. Przedstawione wartości są subiektywnymi ocenami autorów i maja tylko znaczenie orientacyjne.

Rzutowanie systemu zintegrowanego na praktyki szczegółowe CMMI

Pewnym wskaźnikiem jakości systemu zintegrowanego może być poziom wsparcia dla praktyk szczegółowych CMMI [3]. W tabelach zawarto praktyki szczegółowe związane tylko z procesami dotyczącymi poziomu 2 i 3 dojrzałości.

Zarządzanie procesowe

Zorientowanie na procesy w organizacji	
SG1 – Określenie możliwości poprawy procesów	
• SP 1.1 Wyznacz potrzeby procesów organizacyjnych	0/0
• SP 1.2 Dokonaj oceny procesów organizacyjnych	0/0
• SP 1.3 Zidentyfikuj możliwości poprawy procesów organizacyjnych	0/0
SG2 – Zaplanuj i zaimplementuj działania poprawy procesów	
• SP 1.1 Wyznacz plany działań	1/1
• SP 1.2 Zaimplementuj plany działań procesów	0/0
• SP 1.3 Dokonaj wdrożenia ocen procesów organizacyjnych	0/0
• SP 1.4 Wprowadź uzyskane doświadczenia do zasobu procesów organizacyjnych	0/0
Definicja procesów organizacji	
SG1 – Ustal zasób procesów organizacyjnych	
• SP 1.1 Wyznacz standardowe procesy	1/2
• SP 1.2 Wyznacz opisy modeli cyklów życia	1/2
• SP 1.3 Wyznacz kryteria dostosowawcze i instrukcje	0/0
• SP 1.4 Wyznacz repozytorium miar	0/0
organizacji	
• SP 1.5 Wyznacz bibliotekę zasobów procesów organizacyjnych	1/2
Szkolenie w organizacji	
SG1 – Ustal potencjał szkoleniowy organizacji	
• SP 1.1 Ustal strategiczne potrzeby szkoleniowe	0/0
• SP 1.2 Ustal, jakie potrzeby szkoleniowe stanowią odpowiedzialność organizacji	0/0
• SP 1.3 Ustal plan taktyczny szkolenia w organizacji	1/1
• SP 1.4 Ustal potencjał szkoleniowy	0/0
SG2 – Dostarcz koniecznego szkolenia	
• SP 1.1 Dostarcz szkolenia	0/0
• SP 1.2 Przechowaj dane o szkoleniach	0/3
• SP 1.3 Dokonaj oceny efektywności szkoleń	0/3

Zarządzanie projektem

Planowanie projektu	
SG1 – Wyznaczenie oszacowań parametrów planów projektu i ich utrzymywanie	
• SP 1.1 Wyznacz zakres projektu	1/1
• SP 1.2 Wyznacz oszacowania dla produktów pracy i atrybutów zadań	3/3
• SP 1.3 Zdefiniuj cykl życia projektu	3/3
• SP 1.4 Wyznacz oszacowania dla wysiłków i kosztów potrzebnych dla	3/3

wykonania produktów pracy i zadań	
SG2 – Opracowanie planu projektu	
• SP 2.1 Wyznacz budżet i harmonogram	2/3
• SP 2.2 Zidentyfikuj ryzyka projektu	2/2
• SP 2.3 Planuj zarządzania danymi projektu	3/3
• SP 2.4 Planuj zasoby projektu	2/3
• SP 2.5 Planuj potrzebną wiedzę i umiejętności	0/0
• SP 2.6 Planuj zaangażowanie (włączenia) udziałowców	1/2
• SP 2.7 Określ plany projektu	3/3
SG3 – Uzyskanie akceptacji planu	
• SP 3.1 – Dokonaj przeglądu planów, które mają wpływ na projekt w celu zrozumienia uzgodnień projektowych	2/2
• SP 3.2 – Dokonaj przeglądu planu projektu, który określa dostępne i wymagane zasoby	2/2
• SP 3.3 – Uzyskaj akceptację planu (odpowiednich udziałowców wykonujących i wspomagających wykonanie projektu)	0/0
Monitorowanie i kontrola projektu	
SG1 – Monitorowanie projektu	
• SP 1.1 – Monitoruj parametry planu projektu	2/3
• SP 1.2 – Monitoruj uzgodnienia	0/0
• SP 1.3 – Monitoruj ryzyka projektowe	3/3
• SP 1.4 – Monitoruj zarządzanie danymi projektowymi	2/2
• SP 1.5 – Monitoruj zaangażowanie udziałowców	1/1
• SP 1.6 – Dokonuj przeglądu postępu prac	2/2
• SP 1.7 – Dokonuj przeglądu kamieni milowych	2/2
SG2 – Zarządzanie akcjami naprawczym:	
• SP 2.1 Dokonuj analizy problemów	0/0
• SP 2.2 Podejmuj działania naprawcze (wobec powstałych problemów)	1/1
• SP 2.3 Zarządzaj akcjami naprawczymi	2/2
Zarządzanie uzgodnieniami z dostawcami	
SG1 – Ustalenie uzgodnień z dostawcami:	
• SP 1.1 – Określ sposoby nabycia	0/0
• SP 1.2 – Dokonaj wyboru dostawców	0/0
• SP 1.3 – Ustal uzgodnienia z dostawcami	0/0
SG2 – Spełnienie wymagań dostaw:	
• SP 2.1 Dokonaj przeglądu produktów gotowych (COTS)	0/0
• SP 2.2 Wykonaj uzgodnienia z dostawcami	0/0
• SP 2.3 Dokonaj akceptacji nabytego produktu	1/1
• SP 2.4 Przekaz nabyty produkt do projektu	0/2
Zintegrowane zarządzanie projektem (dla IPPD – Integrated Product and Process Development)	
SG1 – Korzystaj z procesów zdefiniowanych dla projektu:	
• SP 1.1 – Ustal zdefiniowany proces projektu	2/2
• SP 1.2 – Stosuj bazę procesów organizacji dla planowania działań projektowych	1/1
• SP 1.3 – Zintegruj plany	2/3
• SP 1.4 – Zarządzaj projektem stosując zintegrowane plany	2/3
• SP 1.5 – Dokonaj wkładu do bazy procesów organizacji	0/1
SG2 – Koordynuj współpracę z odpowiednimi udziałowcami	
• SP 2.1 Zarządzaj zaangażowaniem udziałowców	1/1

• SP 2.2 Zarządzaj zależnościami	0/0
• SP 2.3 Rozwiązuj problemy koordynacyjne	1/2
SG3 – Stosuj wspólną wizję projektu	
• SP 3.1 Zdefiniuj otoczenie wspólnej wizji projektu	3/3
• SP 3.2 Ustal wspólną wizję projektu	3/3
SG4 – Zorganizuj zintegrowane zespoły dla IPPD	
• SP 4.1 Określ zintegrowaną strukturę zespołu dla projektu	3/3
• SP 4.2 Opracuj wstępny przydział wymagań do zintegrowanego zespołu	3/3
• SP 4.3 Ustal zintegrowany zespół	3/3
Zarządzanie ryzykiem	
SG1 – Przygotowanie do zarządzania ryzykiem:	
• SP 1.1 Określ źródła ryzyk i kategorii	1/1
• SP 1.2 Zdefiniuj parametry ryzyk	2/3
• SP 1.3 Określ strategię zarządzania ryzykiem	1/1
SG2 – Identyfikacja i analiza ryzyk:	
• SP 2.1 Zidentyfikuj ryzyka	1/1
• SP 2.2 Oszacuj, klasyfikuj i nadaj priorytety	3/3
SG3 – Zapobieganie ryzykom:	
• SP 3.1 Opracuj plany przeciwdziałającym ryzykom	3/3
• SP 3.2 Wdróż plany przeciwdziałającym ryzykom	1/1
Zintegrowanie zespołów (grup)	
SG1 – Ustal skład zespołów	
• SP 1.1 Zidentyfikuj zadania zespołu	3/3
• SP 1.2 Zdefiniuj wymagane kompetencje i umiejętności	0/2
• SP 1.3 Przydziel odpowiednich członków zespołu	2/2

SG2 – Kierowanie operacjami załogi	
• SP 2.1 Ustal wspólna wizję	3/3
• SP 2.2 Ustal kartę (kontrakt) zespołu	0/2
• SP 2.3 Zdefiniuj role i odpowiedzialności	3/3
• SP 2.4 Ustal procedury operacyjne	1/1
• SP 2.5 Współpracuj między sprzęgniętymi zespołami	3/3
Zintegrowane zarządzanie uzgodnieniami z dostawcami	
SG1 – Wyspecyfikuj ogólne cele	
• SP 1.1 Analizuj potencjalne źródła produktów	0/0
• SP 1.2 Dokonaj oceny i określ źródła produktów	0/2
SG2 – Skoordynowanie współpracy z dostawcami	
• SP 2.1 Dokonaj oceny wybranych produktów pracy dostawców	0/2
• SP 2.2 Zrewiduj uzgodnienia z dostawcami i relacje między nimi	0/2

Inżynieria oprogramowania

Zarządzanie wymaganiami	
SG1 – Zarządzanie wymaganiami:	
• SP 1.1 Uzyskaj zrozumienia znaczenia wymagań	0/1
• SP 1.2 Uzyskaj akceptację wymagań (przez uczestników projektu)	0/1
• SP 1.3 Zarządzaj zmianami wymagań	2/3
• SP 1.4 Zarządzaj wzajemną zależnością pomiędzy wymaganiami, a planami projektu i produktami pracy	1/3
• SP 1.5 Zidentyfikuj braku spójności pomiędzy produktami pracy, a wymaganiami	1/1
Rozwój wymagań	

SG1 – Rozwijanie wymagań klienta:	
• SP 1.1 Wyjaw potrzeby (udziałowców, a także ich oczekiwań, ograniczeń i interfejsów)	1/1
• SP 1.2 Rozwijaj wymagania klienta (transformacja wymagań udziałowców na wymagania klienta)	1/3
SG2 – Rozwijanie wymagań wobec produktu:	
• SP 2.1 Wyznacz wymagania wobec produktu i jego komponentów	1/3
• SP 2.2 Przydziel wymagania do każdego produktu	1/3
• SP 2.3 Zidentyfikuj wymagania wobec interfejsu	0/1
SG3 – Analiza i ocena wymagań:	
• SP 3.1Określ zasady operacyjne i scenariusze	0/3
• SP 3.2 Określ definicje wymaganej funkcjonalności	0/1
• SP 3.3 Dokonaj analizy wymagań	1/3
• SP 3.4 Dokonaj analiza wymagań pod kątem wyważenia potrzeb udziałowców i możliwości	1/2
• SP 3.5 Dokonaj walidacji wymagań za pomocą zrozumiałych metod	0/1
Rozwiązania technologiczne	
SG1 – Wybór rozwiązań dla komponentów produktu:	
• SP 1.1 Opracowanie szczegółowych rozwiązań alternatywnych i kryteriów wyboru	0/0
• SP 1.2 Opracowanie koncepcji operacyjnej, scenariuszy	0/0
• SP 1.3 Wybór rozwiązań dla komponentów produktu	0/0
SG2 – Rozwijanie projektów dla produktu i komponentów:	
• SP 2.1 Opracowanie projektu produktu i jego komponentów	0/0
• SP 2.2 Ustalenie pakietu danych o produkcie	0/0
• SP 2.3 Opracuj interfejsy produktu według zadanych kryteriów	0/0
• SP 2.4 Wyznacz opisy interfejsów	0/0
• SP 2.3 Dokonaj analizy wobec produktu względem, zakupu, rozwoju lub ponownego użycia	0/1
SG3 – Implementacja projektu produktu:	
• SP 3.1Implementuj projekt	0/1
• SP 3.2 Opracuj dokumentację wspomagającej produkt	0/0
Integracja produktu	
SG1 – Przygotowanie produktu do integracji	
• SP 1.1 Wyznacz kolejność integracji	0/0
• SP 1.2 Ustal środowisko integracji produktu	0/0
• SP 1.3 Ustal procedury i kryteria integracji produktu	0/0
SG2 – Zapewnij zgodność sprzęgów (interfejsów) produktu	
• SP 2.1 Dokonaj przeglądu interfejsów ze względu na ich kompletność	0/1
• SP 2.2 Zarządzaj interfejsami (definicjami, projektami i zmianami)	0/1
SG3 – Scalanie komponentów produktu i dostawa produktu	
• SP 3.1 Potwierdzenie gotowości komponentów produktu dla integracji	0/3
• SP 3.2 Zintegruj komponenty produktu	0/3
• SP 3.3 Dokonaj oceny scalonych komponentów produktu (ze względu na sprzęgi)	0/3
• SP 3.4 Dokonaj pakowania i dostawy produktu lub komponentu produktu	1/3
Weryfikacja	
SG1 – Przygotowanie weryfikacji	
• SP 1.1 Wybierz produkty pracy dla weryfikacji	0/2

• SP 1.2 Ustal środowisko weryfikacji	0/2
• SP 1.3 Ustal procedury i kryteria weryfikacji	0/3
SG2 – Dokonywanie przeglądów w postaci oglądu (peer review)	
• SP 2.1 Przygotuj przegląd	0/3
• SP 2.2 Przeprowadź przegląd	1/3
• SP 2.3 Dokonaj analizę danych z przeglądu	0/1
SG3 – Weryfikacja wybranych produktów pracy	
• SP 3.1 Dokonaj weryfikacji	1/3
• SP 3.2 Dokonaj analizy wyników weryfikacji i identyfikacja akcji naprawczych	1/3
Walidacja	
SG1 – Przygotowanie walidacji	
• SP 1.1 Przygotuj produkt do walidacji	0/1
• SP 1.2 Ustal środowisko do walidacji	0/1
• SP 1.3 Ustal procedury i kryteria walidacji	0/3
SG2 – Walidacja produktu lub komponentów produktu	
• SP 2.1 Przeprowadź walidację	0/1
• SP 2.2 Dokonaj analizy wyników walidacji	0/2

Zarządzanie pomocnicze

Zarządzanie konfiguracją	
SG1 – Ustal bazy produktów (baselines)	
• SP 1.1 Zidentyfikuj jednostki konfiguracji	1/3
• SP 1.2 Opracuj system zarządzania konfiguracją	1/3
• SP 1.3 Utwórz bazę produktów wydawanych	1/3
SG2 – Śledź i kontroluj zmiany	
• SP 2.1 Śledź żądania zmian	1/3
• SP 2.2 Kontroluj zmiany jednostek konfiguracji	1/3
SG1 – Określ spójność (integralność)	
• SP 3.1 Ustal i utrzymuj dane dotyczące zarządzania konfiguracją	0/3
• SP 3.2 Przeprowadź audyty konfiguracji	0/3
Zapewnienie jakości procesu i produktu	
SG1 – W sposób obiektywny dokonaj oceny procesów i produktów pracy	
• SP 1.1 Dokonaj obiektywnej oceny procesów	0/1
• SP 1.2 Dokonaj obiektywnej oceny produktów pracy i usług	0/1
SG1 – Dostarcz obiektywny wgląd	
• SP 2.1 Poinformuj o problemach związanych z niezgodnością i o rozwiązaniach tych problemów	1/2
• SP 2.2 Ustal i utrzymuj dane dotyczące aktywności zapewnienia jakości	1/2
Pomiary i analiza	
SG1 – Dostosuj działania związane z pomiarami i analizą (do potrzeb i celów)	
• SP 1.1 Określ cele pomiarowe	1/2
• SP 1.2 Wyspecyfikuj miary	1/2
• SP 1.3 Wyspecyfikuj sposób zbierania danych i procedury przechowywania	1/2
• SP 1.4 Wyspecyfikuj procedury analiz	1/2
SG2 – Dostarcz wyniki pomiarów	
• SP 2.1 Zbieraj dane pomiarowe	1/2
• SP 2.2 Analizuj dane pomiarowe	1/2
• SP 2.3 Przechowywuj dane i wyniki	1/2
• SP 2.4 Poinformuj o wynikach (odpowiednich udziałowców)	1/2
Analiza decyzyjna i rozwiązywanie problemów	

SG1 – Dokonaj oceny rozwiązań alternatywnych	
• SP 1.1 Określ instrukcje dla analiz decyzyjnych	0/0
• SP 1.2 Ustal kryteria oceny	0/0
• SP 1.3 Zidentyfikuj alternatywne rozwiązania	0/0
• SP 1.4 Wybierz metody oceny	0/0
• SP 1.5 Oceń rozwiązania alternatywne	0/0
• SP 1.6 Wybierz rozwiązania	0/0
Środowisko organizacyjne do integracji	
SG1 – Dostarcz infrastrukturę do integracji współpracy(IPPD)	
• SP 1.1 Określ wspólną wizję	2/3
organizacji	
• SP 1.2 Ustal zintegrowane środowisko pracy	2/3
• SP 1.3 Ustal wymagane umiejętności związane z infrastrukturą integracyjną	0/0
SG2 – Zarządzanie zasobami dla celów integracji	
• SP 2.1 Określ zasady przywództwa	0/0
• SP 2.2 Określ bodźce dla integracji	0/0
• SP 2.3 Określ mechanizm równoważący odpowiedzialności grupy i całej organizacji	0/0

Zamieszczone oceny poziomu wsparcia mają charakter jedynie orientacyjny i wskazują raczej na metodykę ocen proponowaną względem zintegrowanych systemów informatycznych wspomagających procesy zarządzania wytwarzaniem oprogramowania Zdaniem autorów ten sposób podejścia jest słuszny, lecz będzie wymagał dopracowania.

Pierwsza tabela dotyczy zarządzania procesowego. Dla tego typów procesów wsparcie systemu zintegrowanego, będzie się głównie odnosić do praktyk związanych z definicją procesów zarządzania.

W drugiej tabeli zamieszczono oceny dotyczące praktyk związanych z zarządzaniem projektem. Ogólnie wsparcie tych praktyk przez system zintegrowany powinno być wysokie, w szczególności system zintegrowany powinien oferować wspólną wizję projektu, wspomagać integrację zespołów. Wspomaganie planowania i monitorowania realizacji projektów jest oczywistą koniecznością systemu zintegrowanego.

Tabela trzecia dotyczy procesów zarządzania związanych z inżynierią oprogramowania. Oczywiście w przypadku procesów wytwórczych związanych z inżynierią oprogramowania istnieje wiele narzędzi wspomagające te procesy. Na poziomie zarządzania ważne jest wspomaganie zbierania i rozwoju wymagań oraz wspomaganie weryfikacji i walidacji.

Praktyki dla procesów pomocniczych opisanych w tabeli czwartej odnoszą się do praktyk z procesów opisanych wcześniej. Poziom wsparcia zależy w głównej mierze od referencyjnych procesów.

Rzutowanie systemu zintegrowanego na obszary systemu jakości ISO 9000

Ze względu na popularność standardu ISO [7, 8] i stosunkowo niskie koszty certyfikacji oraz ze względu na wysoką zgodność standardu z CMMI warto dokonać porównania wsparcia narzędzi zintegrowanych dla obszarów standardu ISO.

System zarządzania jakością

4.1. Wymagania ogólne	
4.2. Wymagania dotyczące dokumentacji	
• 4.2.1 Wymagania ogólne	0/0
• 4.2.2 Księga jakości	0/0
• 4.2.3 Nadzór nad dokumentami	0/0
• 4.2.3 Nadzór nad zapisami	0/0

Odpowiedzialność kierownictwa

5.1. Zaangażowanie kierownictwa	1/1
5.2. Orientacja na klienta	1/1
5.3. Polityka jakości	2/2
5.4. Planowanie	
• 5.4.1 Cele dotyczące jakości	0/0
• 5.4.2 Planowanie systemu zarządzania jakością	2/2
5.5. Odpowiedzialność, uprawnienia i komunikacja	
• 5.5.1 Odpowiedzialność i uprawnienia	2/2
• 5.5.2 Przedstawiciel kierownictwa	0/0
• 5.5.3 Komunikacja wewnętrzna	2/2
5.6. Przegląd zarządzania	
• 5.6.1 Postanowienia ogólne	0/1
• 5.6.2 Dane wejściowe dla przeglądu	0/1
• 5.6.3 Dane wyjściowe z przeglądu	0/1

Zarządzanie zasobami

6.1. Zapewnienie zasobów	0/0
6.2. Zasoby ludzkie	
• 6.2.1 Postanowienia ogólne	0/1
• 6.2.2 Kompetencje, świadomość i szkolenie	0/3
6.3. Infrastruktura	0/0
6.4. Środowisko pracy	1/2

Realizacja wyrobu

7.1. Planowanie realizacji wyrobu	2/3
7.2. Procesy związane z klientem	
• 7.2.1 Określenie wymagań wobec wyrobu	2/3
• 7.2.2 Przegląd wymagań dotyczących wyrobu	2/3
• 7.2.3 Komunikacja z klientem	2/3
7.3. Projektowanie i rozwój	
• 7.3.1 Planowanie projektowania i rozwoju	2/3
• 7.3.2 Dane wejściowe do projektowania i rozwoju	1/3
• 7.3.3 Dane wyjściowe do projektowania i rozwoju	1/3
• 7.3.4 Przegląd projektowania i rozwoju	1/3
• 7.3.5 Weryfikacja planowania i rozwoju	1/2
• 7.3.6 Walidacja w projektowaniu i rozwoju	1/2
• 7.3.7 Nadzorowanie zmian w projektowaniu i rozwoju	2/3
7.4. Zakupy	
• 7.4.1 Proces zakupu	0/2
• 7.4.2 Informacje dotyczące zakupów	0/2
• 7.4.3 Weryfikacja zakupionego wyrobu	0/2
7.5. Produkcja i dostarczanie usługi	
• 7.5.1 Nadzorowanie produkcji i dostarczenie usługi	1/2
• 7.5.2 Walidacja procesów produkcji	1/2

i dostarczenia usługi	
• 7.5.3 Identyfikacja i identyfikowalność	1/2
• 7.5.4 Własność klienta	0/3
• 7.5.5 Zabezpieczenie wyrobu	0/2
7.6. Nadzorowanie wyposażenia do monitorowania i pomiarów	1/2

Pomiary, analizy i doskonalenie

8.1. Postanowienia ogólne	0/0
8.2. Monitorowanie i pomiary	
• 8.2.1 Zadowolenie klienta	0/1
• 8.2.2 Audit wewnętrzny	0/1
• 8.2.3 Monitorowanie i pomiary procesów	1/2
8.3. Nadzór nad wyrobem niezgodnym	0/3
8.4. Analiza danych	1/3
8.5. Doskonalenie	
• 8.5.1 Ciągłe doskonalenie	0/0
• 8.5.2 Działania korygujące	0/0
• 8.5.2 Działania zapobiegawcze	0/0

Z przedstawionych powyżej tabel wynika inne podejście standardu ISO niż to jest w CMMI wynikające z tego, że standard ISO jest standardem ogólnym. Tym nie mniej istnieją porównania tych standardów świadczące o ich dużym powiązaniu. Podobieństwa te by się znacznie uwydatniły, gdyby uwzględnić trzeci poziom szczegółowości standardu, na co z kolei nie pozwala ograniczona objętość opracowania. Największe wsparcie system zintegrowany dostarczać może dla obszaru realizacji wyrobu.

Tak jak to mówiono wcześniej zamieszczone wartości wsparcia są subiektywnymi ocenami autorów i mają tylko znaczenie orientacyjne. Tym nie mniej przy ocenie innych systemów zintegrowanych warto mieć podobne odniesienie referencyjne. Celem autorów jest także doskonalenie miar porównawczych, tak by zmniejszać wielkość oceny subiektywnej na korzyść oceny obiektywnej.

Z przedstawionych w dużej mierze subiektywnych wyników widać, że poziom wsparcia do zarządzania wytwarzaniem oprogramowania przez proponowane oprogramowanie Project Servant jest wysokie, a sam system zyskuje na coraz większej popularności.

6. Wnioski

Przedstawiony w opracowaniu zintegrowany system informatyczny o nazwie Project Servant (PSV) wspomaga procesy zarządzania wytwarzaniem oprogramowania w sposób znaczący. System ten nadaje się bardzo dobrze zarówno dla firm informatycznych zlokalizowanych w jednym miejscu jak i rozproszonych geograficznie. W wielu przypadkach jest konieczne i celowe by zlecający mógł monitorować postęp i rodzaje wykonywanych czynności. PSV jest właśnie takim narzędziem umożliwiającym wgląd osób nadzorujących projekt w przebieg realizowanych prac. PSV może być również dedykowany dla pojedynczych projektów, jakimi są na przykład granty, czy projekty unijne, w których

uczestniczą różne rozproszone geograficznie ośrodki.

Przedstawione subiektywne oceny w sposób orientacyjny wskazują na potencjalne możliwości systemu oraz na ewentualne kierunki rozwoju. PSV w dużej mierze wspomaga standardy jakości, ułatwia osiągnięcie zmian jakościowych i wydajnościowych.

System ten sprawdził się w informatycznych firmach krajowych i zagranicznych przy różnej liczbie rzeczywistych użytkowników systemu sięgających ponad 100 osób. System ten nie posiada ograniczeń na liczbę użytkowników i jest dobrze skalowalny.

Literatura

1. Boris Mutafelija, Harvey Stromberg. ISO 9001:2000 CMMI v1.1 Mappings, 2003.
2. Boris Mutafelija, Harvey Stromberg. Exploring CMMI-ISO 9001:2000 Synergy when Developing a Process Improvement Strategy. SEPG 2003 Conference.
3. Capability Maturity Model Integration (CMMI) Version 1.1, CMMI for System Engineering, Software Eng., Integrated Product and Process Development and Supplier Sourcing. Carnegie Mellon SEI, CMMI Product Team, 2002, p. 724.
4. Jan Werewka, Marek Sujdak: Standaryzacja analiz wykonywanych prac w projektach informatycznych, XII Konferencja Sieci i Systemy Informatyczne, Łódź 2004, Politechnika Łódzka, pp. 593-602.
5. J. Werewka, M. Piwowarczyk: Zagadnienie kontroli pracy w procesie wytwarzania oprogramowania. Problemy i metody inżynierii oprogramowania. Wydawnictwa Naukowo-Techniczne, Strony 567-576.
6. Marek Sujdak, Jan Werewka: Zagadnienie tworzenia szablonów w projektach informatycznych, XII Konferencja Sieci i Systemy Informatyczne, Łódź 2004, Politechnika Łódzka, pp. 583-592.
7. Przewodnik ISO 9000. Materiały informacyjne nt. wdrażania systemu zarządzania jakością wg norm ISO serii 9000. GIG.
8. System zarządzania jakością. Wytyczne doskonalenia funkcjonowania. Polska Norma PN-EN ISO 9004, PKN, 2001.

ROZDZIAŁ XI

METODY ZAPEWNIENIA UŻYTECZNOŚCI OPROGRAMOWANIA

Beata HYSA

1. Jakość użytkowa produktu informatycznego

Sukces rynkowy współczesnego produktu informatycznego jest uzależniony od spełnienia jakościowych oczekiwań klienta nie tylko w zakresie technicznym, ale przede wszystkim w warstwie użytkowej. Można powiedzieć, że jakość produktu informatycznego jest dzisiaj kombinacją trzech składowych [SIKO05]:

- jakości technicznej (konstrukcyjnej), wyrażanej najczęściej przez jakość kodu źródłowego oraz zgodność z specyficznymi wymaganiami (w tym bezpieczeństwa),
- jakości ergonomicznej, zwykle rozumianej jako wygoda obsługi produktu, która wiąże się ze zgodnością interfejsu użytkownika z zaleceniami ergonomicznymi,
- jakości użytkowej, rozumianej jako praktyczna użyteczność produktu, która jest bezpośrednio odczuwana i weryfikowana przez użytkownika w rzeczywistych zadaniach roboczych.

Norma ISO 9126-1 podaje, że użyteczność (ang. usability) jest jedną z sześciu podstawowych charakterystyk jakości produktu: funkcjonalności, niezawodności, użyteczności, efektywności, pielęgnowalności (łatwości konserwacji), przenośności. Można powiedzieć, że użyteczność oznacza stopień dopasowania rozwiązań informatycznych (produktów interaktywnych) do potrzeb użytkowników, mierzony łatwością nauki i obsługi, skutecznością i efektywnością uzyskanych wyników pracy, oraz odczuwaną satysfakcją z łatwego uzyskania oczekiwanego wyniku. [SIKO97]

Praca z produktem o wysokiej użyteczności jest bardziej efektywna, łatwiejsza, przyjemniejsza, oraz nie powoduje uczucia frustracji oraz zagubienia.

2. Metody zapewnienia użyteczności

Istnieje wiele sposobów zapewnienia użyteczności. W tabeli 1 przedstawiono najbardziej popularne metody podzielone ze względu na etapy projektu, w którym są najczęściej stosowane.

Tabela 1

Metody zapewnienia użyteczności	Poszczególne etapy projektowania				
	Planowanie	**Analiza**	**Projektowanie**	**Testowanie i ocena**	**Użytkowanie, doskonalenie**
	Planowanie użyteczności	Wywiady	Prototypowanie	Ocena heurystyczna	Badanie opini
	Analiza kontekstu użytkownika	Analiza zadań użytkownika	Normy	Inspekcje – listy kontrolne	
	Analiza konkurencji	Obserwacja	Przewodniki stylu	Testy z udziałem użytkowników	
		Sortowanie kart	Zalecenia projektowe	Kwestionariusze ankietowe	
		Scenariusze użycia	Czarnoksiężnik z Oz		
		Persony	Prototypowanie wideo		

Tabela1 Metody zapewnienia użytkowania oprogramowania
Źródło: Opracowanie własne na podstawie[[SBWW06]

Poniżej zostały scharakteryzowane niektóre z metod zapewnienia użyteczności oraz korzyści z ich zastosowania.

2.1. Planowanie użyteczności

Planowanie jest podstawowym czynnikiem zapewnienia użyteczności w projekcie informatycznym. Planowanie polega przede wszystkim na opracowaniu planu użyteczności. Dokument „Plan zapewnienia użyteczności" określa miejsce jakości użytkowej w zarządzaniu jakością w projekcie informatycznym. Typowy plan użyteczności zawiera:

- uzasadnienie potrzeby doskonalenia użyteczności produktu, w tym: analizę kosztów i korzyści, określenie wpływu podwyższonej użyteczności na atrakcyjność rynkową produktu i wyniki handlowe (zwykle na tle analizy rozwiązań konkurencyjnych),
- plan prac a w nim: harmonogram działań, osoby odpowiedzialne za poszczególne zadania, podział zasobów na poszczególne prace,

- opis potrzebnej infrastruktury technicznej oraz organizacyjnej, wymaganych do zrealizowania planu.

Zatwierdzony przez kierownictwo projektu plan użyteczności zapewnia, że prace nad doskonaleniem użyteczności będą prowadzone w planowy sposób i koordynowane przez osobę odpowiedzialną nie tylko za jakość finalnego produktu, ale także za współpracę z docelowymi użytkownikami na wszystkich etapach realizacji projektu.

Plan zapewnienia użyteczności zwykle pozostaje w bezpośredniej relacji do polityki jakości firmy, co ułatwia włączenie go do już funkcjonujących w firmie planów jakości i systemów zarządzania jakością [SBWW06].

2.2. Analiza kontekstu użytkowania,

Analiza kontekstu użytkowania (ang. context of use analysis) to jeden z pierwszych etapów zapewniania użyteczności. Analiza kontekstu użytkowania polega na zebraniu szczegółowych informacji dotyczących:

- charakterystyk przyszłych użytkowników,
- opisu środowiska organizacyjnego,
- opisu infrastruktury technicznej, w której planowany system ma funkcjonować.

Informacje te są niezbędne w celu poprawnego określenia wymagań użytkowych dla planowanego systemu oraz określenia pozatechnicznych czynników ryzyka dla planowanego przedsięwzięcia projektowo-wdrożeniowego. Informacje te zbiera się wstępnie podczas fazy planowania, natomiast zostają one uzupełnione i uszczegółowione na etapie analizy wymagań.

2.3. Testy z udziałem użytkowników

Testowanie produktu (prototypu) przez użytkowników jest skuteczną metodą wykrywania problemów z użytecznością systemu. Projektanci są bardziej skłonni do różnych zmian widząc użytkowników zmagających się z trudnościami w obsłudze systemu. Testowanie systemu z udziałem użytkowników pozwala stwierdzić, co im utrudnia pracę oraz określić, w jakich obszarach konieczne są udoskonalenia w sposobie obsługi produktu. Testowanie użyteczności polega na tym, że użytkownicy zaproszeni do udziału w badaniach wykonują z pomocą systemu typowe zadania robocze w ściśle zaplanowany sposób. Przebieg pracy użytkowników jest zazwyczaj rejestrowany na wideo. Prowadzący badanie sporządzają raport, w którym podaje się listę zaobserwowanych problemów z użytecznością systemu oraz zakres zalecanych zmian dotyczących interfejsu użytkownika.

Aby testy z udziałem użytkowników (zwane także testami użytkowymi i testami użyteczności, ang. user testing) były efektywne, powinny być z organizowane w przemyślany sposób i w odpowiednich warunkach – chociaż niekoniecznie muszą się odbywać w profesjonalnym laboratorium.

2.4. Badanie opinii

Badanie subiektywnych opinii użytkowników na temat oprogramowania pozwala określić, jak postrzegana jest jego jakość oraz własności użytkowe. Opinie użytkowników są najczęściej zbierane przy pomocy standaryzowanych kwestionariuszy, co ma zapewnić bardziej obiektywną ocenę oraz porównywalność wyników.

Badanie ankietowe poziomu satysfakcji użytkowników z pracy z oprogramowaniem pozwala na poznanie chęci użytkowników do pracy z danym oprogramowaniem, ewentualnie chęci zakupu danego produktu – często w odniesieniu do produktów konkurencyjnych, oferujących podobną funkcjonalność.

O ile oprogramowanie stanowi główne, udostępniane przez pracodawcę narzędzie pracy użytkownika, niski poziom satysfakcji prowadzi do narzekań personelu, ciągłej frustracji oraz rosnącej absencji – badanie ankietowe pozwala na wczesne wykrycie niepokojących zjawisk.

Wynik subiektywnej oceny jakości użytkowej oprogramowania jako pomiar poziomu satysfakcji stanowi uzupełnienie pomiarów skuteczności i efektywności, które można zebrać bezpośrednio podczas testów użyteczności. O słuszności pomiaru poziomu satysfakcji użytkownika dowodzą różne badania [BEGI04]. Przyjmuje się nawet, że poziom satysfakcji użytkownika jest pochodną jakości użytkowanego wyrobu programowego.

2.5.Czarnoksiężnik z Oz

Metoda o intrygującej nazwie pozwala na przetestowanie danego interfejsu jeszcze przed jego implementacją. W metodzie tej ukryty operator symuluje (udaje) odpowiedzi systemu, dlatego zwany jest czarnoksiężnikiem.

Metoda może zostać użyta we wczesnym okresie projektowania interfejsu. Pozwala zespołowi pominąć czasochłonny proces implementacji i skupić się na testowaniu użyteczności produktu. „Czarnoksiężnik" (operator) – członek zespołu projektowego – zajmuje miejsce w pokoju kontrolnym, obserwując reakcje użytkownika znajdującego się w pomieszczeniu eksperymentalnym. Zadanie operatora polega na możliwie szybkim symulowaniu odpowiedzi systemu na działania użytkownika. Osoba badana często pozostaje nieświadoma, iż reakcje systemu są zdalnie wywoływane przez operatora.

„Czarnoksiężnik" musi być w stanie szybko i precyzyjnie reagować na napływające dane o działaniach użytkownika. Założenie to najłatwiej zrealizować w przypadku prostych systemów opartych na głosie lub ruchach ręki. Oprogramowanie używane przez operatora powinno być na tyle proste i łatwe w obsłudze, by ten nie miał problemów z obsługą zdarzeń w czasie rzeczywistym. Metoda Czarnoksiężnika z Oz pozwala na:

- zebranie informacji o naturze interakcji,
- wykrycie, które techniki i urządzenia do wprowadzania danych są najbardziej reprezentatywne dla danej interakcji,

- testowanie interakcji przed zbudowaniem funkcjonalnego modelu,
- odkrycie jakie rodzaje problemów ujawniają się podczas pracy z interfejsem,
- analizowanie aspektów formy produktu które mogą mieć wpływ na interakcje.

2.6.Ocena heurystyczna

W tej metodzie grupa ekspertów ocenia zgodność poszczególnych elementów interfejsu ze standardowym zestawem specjalnych wytycznych zwanych heurystykami stosowanych w badaniach użyteczności a następnie porządkuje je w kolejności od najbardziej do najmniej uciążliwych dla użytkownika z podaniem odpowiednich rang uciążliwości. Po ocenie szczegółowy raport przekazywany jest zespołowi projektowemu. Raport powinien jasno definiować zasady, według których nadano rangi poszczególnym problemom.

Heurystyki opracowane przez Jakoba Nielsena i Rolfa Molicha stanowią najbardziej ogólne zasady budowy prawidłowej interakcji człowiek-maszyna i pochodzą z badań statystycznych nad czynnikami najsilniej kształtującymi opinię użytkowników w zakresie odczuwanej wygody i satysfakcji z oprogramowania. Są to [NIEL94]

Pokazuj status systemu. System powinien zawsze informować użytkownika, co się dzieje przez odpowiednie potwierdzenia i komunikaty.

Zachowaj zgodność pomiędzy systemem a rzeczywistością. System powinien mówić językiem użytkownika i posługiwać się zrozumiałymi analogiami zaczerpniętymi z rzeczywistości.

Daj użytkownikowi pełną kontrolę. Użytkownicy często wybierają błędne opcje i dlatego powinni mieć zapewnione "wyjście awaryjne," najlepiej za pomocą funkcji "cofnij" i "powtórz".

Trzymaj się standardów i zachowaj spójność. Te same słowa, symbole, sytuacje i działania powinny być stosowane w jednakowy sposób w całym produkcie, w zgodzie z zasadami przyjętymi dla danego środowiska, platformy czy systemu operacyjnego.

Zapobiegaj błędom. Zapobieganie błędom przez dopracowany dialog z użytkownikiem jest mniej pracochłonne niż projektowanie wyrafinowanego systemu obsługi błędów.

Pozwalaj wybierać zamiast zmuszać do pamiętania. Działania użytkownika powinny być wynikiem wyboru z listy, a nie przywoływania z pamięci; wszystkie potrzebne w danej sytuacji informacje i instrukcje powinny być cały czas widoczne na ekranie, aby nie obciążać pamięci użytkownika.

Zapewnij elastyczność i efektywność. Użytkownicy powinni mieć możliwość dopasowywania sposobu wykonywania typowych zadań oraz dostępu "na skróty" do potrzebnych funkcji.

Dbaj o estetykę i umiar. Oszczędny układ graficzny polepsza czytelność, zmniejsza obciążenie wzroku i skraca czas odszukania informacji.

Zapewnij skuteczną obsługę błędów. Komunikaty o błędach powinny być sformułowane prostym, życzliwym językiem oraz powinny wskazywać typ problemu i sposób jego rozwiązania.

Zadbaj o pomoc i dokumentację. Jeśli system jest zaopatrzony w dokumentację, powinna ona umożliwiać szybkie odnalezienie żądanej informacji, a instrukcje rozwiązywania problemów powinny być zwięzłe i dotyczyć zadań użytkownika.

Ocena heurystyczna pozwala na:

- szybkie i w miarę tanie wypracowanie informacji zwrotnej dla projektantów, oraz ocenę możliwości dalszego ulepszenia interfejsu,
- przewidzenie wielu problemów przed testowaniem z udziałem użytkowników, co daje dużą oszczędność czasu i środków finansowych,
- znalezienie błędów trudnych do wykrycia za pomocą metod opierających się na mierzeniu wydajności,
- uzyskanie zgodności z powszechnie spotykanymi standardami.

2.7. Persony

Persona jest to specjalnie opracowany profil typowego użytkownika, który pozwala trafniej określić oczekiwania poszczególnych grup docelowych użytkowników systemu i przez to polepszyć jego funkcjonalność i użyteczność. Stworzenie profilów użytkownika w postaci „person" pozwala na:

- wyraziste scharakteryzowanie głównych grup użytkowników oraz lepsze dopasowanie funkcjonalności systemu do ich rzeczywistych potrzeb,
- dokładniejszy opis docelowego środowiska użytkowania

Profil jest definiowany przede wszystkim przez cele i oczekiwania użytkownika, w tym przez korzyści, które dana „persona" zamierza osiągnąć z użytkowania planowanego systemu. Dane do profilu pochodzą z obserwacji i wywiadów z rzeczywistymi osobami, które w przyszłości będą korzystały z systemu w różnych rolach i na różnych stanowiskach. Każdy profil powinien mieć nazwisko i zdjęcie „persony" (z reguły zmyślone), a dodatkowo mogą występować inne dane jak: wiek, płeć, stanowisko, miejsce pracy, opis stanowiska pracy, rodzaj problemów użytkownika i wykonywanych zadań, z jakich informacji użytkownik korzysta w swojej pracy i skąd je pozyskuje i wiele innych.

Technika „person" jest wykorzystywana przede wszystkim w projektowaniu oprogramowania dla przedsiębiorstw, jak i portali korporacyjnych przeznaczonych dla pracowników, których charakterystyki łatwiej jest przygotować.

3. Znaczenie użyteczności

Użytkownicy coraz to większe znaczenie przykładają do własności użytkowych produktu, co sprawia, że zarządzanie jakością w przodujących firmach

informatycznych w znacznym stopniu jest dziś ukierunkowane na doskonalenie użyteczności i zapewnienie pełnej satysfakcji klienta-użytkownika. Wynika to z coraz częstszego postrzegania użyteczności jako kluczowej składowej jakości produktu, oraz z akceptowaniem nakładów związanych z doskonaleniem użyteczności jako celowej inwestycji, która szybko się zwróci (zwiększając szanse na sukces rynkowy produktu, budując jego przewagę konkurencyjną oraz kształtując wizerunek wytwórcy i wartość marki). Z tego powodu niektóre firmy decydują się prowadzić własne laboratoria badań użyteczności. Inne korzystają z usługodawców zewnętrznych, oferujących tego typu badania z udziałem użytkowników na podobnej zasadzie, jak zleca się typowe badania rynku.

We wszystkich przypadkach niedostatki użyteczności produktu pociągają nieuzasadnione koszty poprawy wad funkcjonalnych i użytkowych, pogarszają wizerunek firmy oraz w znacznym stopniu decydują o komercyjnym sukcesie lub porażce danego produktu.

Podsumowując możemy stwierdzić, iż użyteczność jest ważna z kilku punktów widzenia:

- dla projektanta: użyteczność w znacznym stopniu decyduje o sukcesie lub porażce produktu, o tym czy ludzie będą chętnie z niego korzystali i polecali innym,
- dla menedżera w przedsiębiorstwie: słaba użyteczność systemu którego używają pracownicy powoduje ich irytację, obniża produktywność, powoduje pomyłki, błędy i reklamacje, pogarszając w efekcie wizerunek firmy w oczach klientów,
- dla handlowca na rynku produktów interaktywnych odczuwana przez nabywcę użyteczność umożliwia wybór pomiędzy systemami, które wywołują zadowolenie ze sprawnego wykonania zadania a systemami, które sprawiając trudności w obsłudze wywołują frustrację i niechęć

Literatura

[SIKO97] Sikorski M.: Ocena jakości użytkowej oprogramowania wspomagającego zarządzanie przedsiębiorstwem. Materiały konferencyjne, Wydawnictwo Uniwersytet Gdański, Gdańsk 1997.

[SIKO05] Sikorski M.: Z punktu widzenia użytkownika: użytkowa warstwa IT i oprogramowania, Computerworld nr 36/2004, 5 października 2004.

[NIEL94] Nielsen J.: Ten Usability Heuristics, W http://www.useit.com/papers/heuristic/heuristic_list.html

[NIEL94] Nielsen J.: How to contact a Heuristic Evaluation, W http://www.useit.com/papers/heuristic/heuristic_evaluation.html

[BEGI04] Begier B.: Ocena jakości wyrobu programowego przez użytkowników, Politechnika Poznańska, Wydawnictwo Politechniki Gdańskiej, Gdańsk 2004,.
[SBWW06] Sikorki M., Bobe M, Wichar M, W www.uzytecznosc.pl

ROZDZIAŁ XII

WPŁYW KULTURY ORGANIZACYJNEJ NA WDROŻENIE SYSTEMU INFORMATYCZNEGO W ABANKU

Justyna Maria BUGAJ

Ekspansja handlu międzynarodowego, deregulacja, fuzje i akwizycje to tylko niektóre czynniki, wyznaczające kierunki rozwoju dla sektora bankowego. Znaczenie tych przemian podkreślono na obradach zgromadzenia Unii Europejskiej[1], którego jednym z najbardziej priorytetowych punktów stanowiło zmniejszenie ryzyka rozliczeniowego oraz podniesienie standardu usług finansowych, w tym w szczególności:

- zapewnienie realizacji transakcji w jak najkrótszym czasie,
- oferowanie nieprzerwanego świadczenia usług przez całą dobę,
- obsługiwanie systemów wielowalutowych,
- kreowanie jak największej wartości dodanej, tak aby klienci nie szukali innego, bezpośredniego dostępu do rynku[2].

Następstwem tego wydarzenia dla Polskiego sektora finansowego była kompleksowa potrzeba zmian, tak w strukturze organizacyjnej, jak i w systemach informatycznych, niespełniających wytycznych Unijnych. W niemal wszystkich bankach i towarzystwach ubezpieczeniowych w Polsce zaczęto powoływać dyrektorów, specjalistów lub nawet zespoły ds. CRM. Jednak efekty tych zmian nie spełniły oczekiwań interesariuszy[3]. Głównym powodem był dystans, zniechęcenie i brak motywacji pracowników do współpracy przy wdrożeniach, często podejmowanych w tym samym czasie przez kilka departamentów danej instytucji. Wdrożenia te również nie zawsze przygotowane były zgodnie z wizją rozwoju danej instytucji finansowej.

Poniżej przedstawiony jest przykład instytucji finansowej, której przygotowanie do wdrożenia (faza I) przebiegało według wytycznych typowego projektu informatycznego. Jednak kolejne etapy (faza II i faza III) przyniosły nie tylko opóźnienia czasowe i wzrost oporu wśród pracowników, ale szczególnie duże obciążenia finansowe samej instytucji. Wdrożenie założonego systemu było konieczne do dalszej egzystencji banku na rynku polskim, stąd ważniejsze było

---

[1] The Group of Thirty – *Consultative Group on International Economic and Monetary Affairs*

[2] Do znamion sytuacji w sektorze finansowym można było również zaliczyć: redukcję marż (wyższy stopień porównywalności produktów, większa konkurencja), zmianę postaw i zachowań klientów (lepiej poinformowani, mniej lojalni i bardziej wymagający) oraz obniżenie barier wejścia na rynek (np. niektóre produkty finansowe zaczynali oferować producenci samochodów lub operatorzy sieci komórkowych).

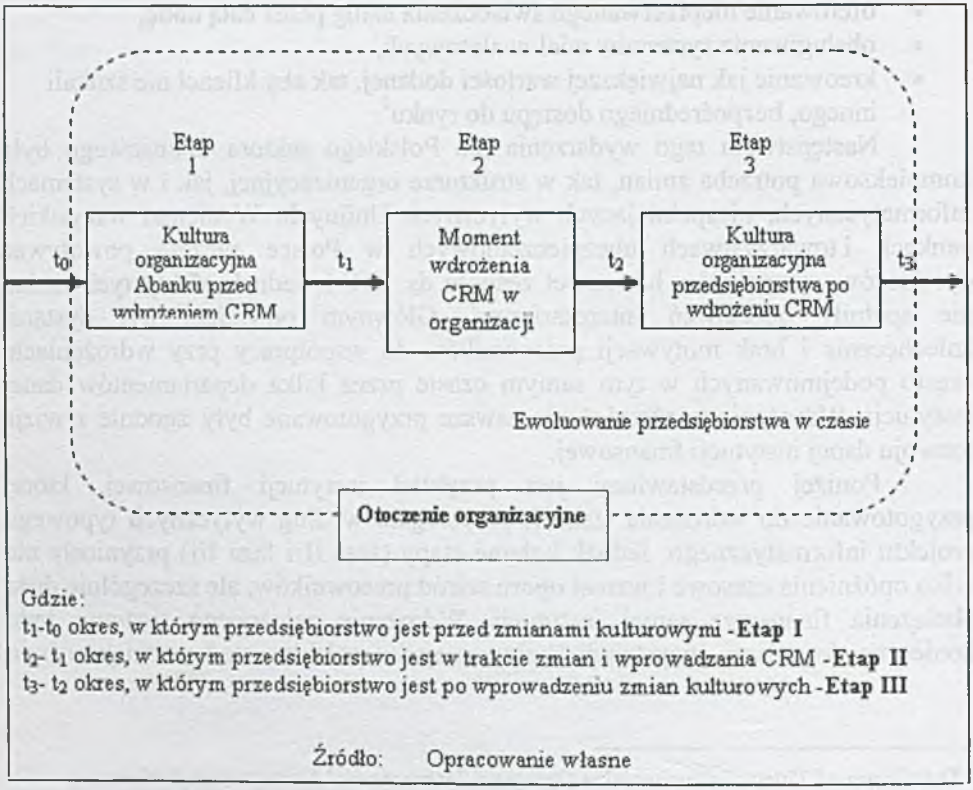
[3] Czyli akcjonariuszy, właścicieli oraz kluczowych klientów.

utworzenie i uruchomienie samego programu za wszelką cenę niż straty wynikłe z niedotrzymania terminów przez firmy wdrożeniowe.

Rysunek 1 przedstawia wzorcowy przebieg wdrożenia koncepcji zarządzania relacjami CRM w przedsiębiorstwie w kontekście zmian kultury organizacyjnej. Pierwszy etap pokazuje sytuację przedsiębiorstwa podczas przygotowania do zmian kulturowych i CRM. Może on trwać dość długo w czasie (od kilku do kilkunastu miesięcy), jest zależny od dostępu organizacji do środków finansowych oraz możliwości zaangażowania doświadczonych specjalistów.

Etap drugi prezentuje okres wdrożenia, który może trwać nawet kilka lat (zależy między innymi od wielkości organizacji, jej charakteru, złożoności, ilości danych). Jest to czas najbardziej burzliwy oraz najtrudniejszy dla firmy, związany z różnymi sytuacjami kryzysowymi, których źródło niekoniecznie jest związane z samym procesem wprowadzania zmian i z projektem.

Rys. 1. Model przebiegu zmian kultury organizacyjnej pod wpływem CRM



Etap trzeci jest zwieńczeniem działań dokonanych podczas poprzednich faz. Od tego, w jaki sposób zostanie przeprowadzone przygotowanie do wdrożenia, jakimi środkami finansowymi (czy nie wyczerpały możliwości organizacji), w jakim czasie (na ile został przekroczony ustalony wcześniej termin) oraz jakie dodatkowe zasoby musiały zostać wykorzystane zależy powodzenie całego

procesu, w tym ukończenie i zakorzenienie się zmian CRM w kulturze organizacyjnej.

Poszczególne etapy przebiegają różnie w różnych organizacjach. Zależy to w dużej mierze od charakteru firmy, jej podatności na zmiany, od wielkości budżetu przeznaczonego na projekt oraz dostępu do specjalistów. Ważny jest również typ zastanej kultury oraz siła jej wpływu na organizację.

Opis organizacji Abank

Abank powstał pod koniec lat osiemdziesiątych ubiegłego wieku jako spółka Skarbu Państwa na bazie istniejącego wcześniej banku państwowego. Od początku istnienia cieszył się dobrą opinią wśród klientów, szczególnie korporacyjnych. Jego notowania jeszcze wzrosły po udanej prywatyzacji i wdrożeniu pierwszego systemu zintegrowanego, który pozwalał na realizację transakcji w czasie rzeczywistym dzięki połączeniom *on-line*.

Pozycja lidera rynkowego wymagała prowadzenia ciągłych obserwacji innych członków sektora bankowego, szczególnie tych Unijnych (Abank stale czerpie wzorce szczególnie z niemieckich korporacji). A ponieważ działania te wiązały się z różnego rodzaju inwestycjami, spowodowały potrzebę wyboru inwestora strategicznego, którego silna pozycja potwierdziłaby czołowe miejsce wśród rankingu polskich banków. Inwestorem tym została duża instytucja finansowa z niemieckim kapitałem.

Od początku powstania Abank grał rolę silnego partnera rynkowego, który w miarę upływu lat powiększał kapitał akcyjny oraz brał udział w przejęciach i fuzjach innych podmiotów gospodarczych, uzupełniających jego ofertę produktową o dobra komplementarne, takie jak: leasing, biuro maklerskie, fundusze inwestycyjne, bank hipoteczny, towarzystwo emerytalne, bank internetowy. W ciągu kilkunastu lat zdobył pozycję w czołówce firm z branży stosując często niekonwencjonalny sposób inspirowania ludzi i wdrażania w życie pomysłów, które z początku wydawały się niemożliwymi do zrealizowania.

Dziś jest to bardzo duża Grupa finansowa zatrudniająca kilkanaście tysięcy pracowników. Należy do najważniejszych instytucji finansowych w Polsce pod względem posiadanego kapitału, rozmiarów działalności, tempa rozwoju, różnorodności oferowanych produktów oraz wysokiego poziomu technologii bankowej. Jest cenionym specjalistą w obsłudze transakcji związanych z eksportem i handlem zagranicznym.

Abank wielokrotnie był nagradzany w różnych branżowych konkursach, w tym zdobył tytuł *Bank Przyjazny dla Przedsiębiorców* i *Najlepszy Bank w Polsce*. Za wykorzystanie nowoczesnych technologii został laureatem konkursu *Lider Informatyki*.

Faza I – rok 2001

Wejście Polski do UE skutkowało liberalizacją prawa bankowego, nie

tylko jeśli chodzi o unifikowanie numerów kont, o zabezpieczenia dostępu do kont internetowych, ale szczególnie o zmianę sposobu raportowania do odpowiednich władz, w tym do *Narodowego Banku Polskiego*. W Abanku wywołało to potrzebę wdrożenia kompleksowego systemu informatycznego, który zastąpiłby funkcjonalnością systemy istniejące dotychczas (wymiana oprogramowania), spełniałby wszystkie wymagania związane z nowym prawem bankowym, jak również spowodowałby wzrost wartości banku wśród klientów i akcjonariuszy.

W roku 2002 został rozpisany przetarg na realizację wdrożenia systemu CRM oraz koncepcji zarządzania relacjami. W jego wyniku projekt podzielono funkcyjnie na trzy części i wybrano trzy firmy do realizacji wdrożenia:

- Firma informatyczna X dostarczała oprogramowanie i szkoleń związanych z CRMO (w tym budowę hurtowni danych)[4],
- Firma informatyczna Y dostarczała oprogramowania i szkoleń związanych z CRMA,
- Firma konsultingowa Z –spełniała rolę kontrolną nad całym projektem i czuwała nad wszystkimi koniecznymi do przeprowadzenia zmianami w organizacji.

Utworzono komitet sterujący, w którego skład wchodzili członkowie zarządu poszczególnych firm z przetargu oraz przedstawiciele kadry kierowniczej Abanku. Podjęto decyzję o powołaniu *zespołów projektowych* przypisanych do poszczególnych funkcji projektu oraz odpowiedzialnych za wykonanie zadań przed *Komitetem Sterującym*. Na tym etapie podjęto również decyzję o wyborze metodologii wdrożenia i sposobach przeprowadzenia koniecznych zmian organizacyjnych[5] oraz akcję informacyjną skierowaną do pracowników, w celu zidentyfikowania i zjednania tzw. *motorów projektowych*[6], wspomagających działania na kolejnych etapach wdrożenia.

Analiza informacji na temat działań podejmowanych w praktyce przez departamenty wskazała na istnienie dużej dbałości o zatrzymanie obecnych klientów Abanku. Wynikało to prawdopodobnie z oczywistości działań w tym zakresie – były one realizowane, na co dzień przez wszystkie departamenty, co więcej miały one wyższy priorytet od pozyskiwania nowych klientów. Wyraźne postulowanie celów dotyczących pozyskiwania nowych klientów świadczyło o silnym dążeniu departamentów do rozszerzenia działań w tym zakresie. W tym czasie były one mocno ograniczone, co wynikało głównie ze znacznego obciążenia zasobów Abanku, a także niewystarczającego wsparcia tego zadania (m.in. poprzez niewystarczające udostępnianie baz potencjalnych klientów, narzędzi ułatwiających oszacowanie potencjału klientów itp.).

W procesie zarządzania klientem tworzenie polityki oraz koordynacja

[4] CRMO- CRM Operacyjny, CRMA-CRM Analityczny

[5] Opis przypadku dotyczy zmian kulturowych zachodzących w organizacji, w związku z czym nie będzie traktował szerzej o innych aspektach projektu, szczególnie wchodzących w tematykę zarządzania projektami informatycznymi.

[6] Inaczej – agentów zmiany.

działań również nie zawsze były dostatecznie scentralizowane. Efektywne zarządzanie procesem pozyskiwania nowych klientów było ograniczone brakiem możliwości analitycznego wsparcia podejmowanych decyzji (np. oceny opłacalności pozyskania danego klienta). Na tym etapie Abank miał trudność z dokonaniem skutecznej oceny prawdopodobieństwa utracenia poszczególnych klientów, a także monitorowaniem liczby oraz wartości klientów traconych. Brak było również systematycznie prowadzonej analizy satysfakcji klientów.

Istniejące w tym czasie w Abanku zasoby technologiczne można było, scharakteryzować w następujący sposób:

- brak jednego, centralnego, spójnego źródła danych; dane były rozproszone, przechowywane w wielu różnych systemach, a także w lokalnych aplikacjach. Najczęściej nie można było ich porównać, nawet jeśli, w niektórych obszarach dotyczyły tego samego klienta. Właściwie nie można było uzyskać danych historycznych.
- brak metody dystrybucji wyników analiz; raporty przekazywało się osobom uprawnionym i zainteresowanym w postaci drukowanej lub poprzez plik elektroniczny (najczęściej **.xls* lub **.doc*), w następstwie występował brak dedykowanego miejsca przechowywania wyników analiz, stąd w razie potrzeby były one powtarzane[7].

W związku z tym realizacja projektu CRM w Abanku miała na celu zapewnienie wsparcia informatycznego pracowników współpracujących z klientami oraz realizację wyznaczonych celów biznesowych, takich jak osiągnięcie wzrostu wartości Abanku dla udziałowców poprzez stabilny wzrost zyskowności klientów. Drogą do osiągnięcia tego celu była realizacja celów pośrednich:

- Zwiększenie liczby klientów Abanku poprzez zatrzymywanie dotychczasowych oraz pozyskiwanie nowych klientów;
- Wzrost wartości klientów Abanku poprzez zwiększenie sprzedaży produktów *up-selling* i *cross-selling*;
- Zwiększenie łatwości zaspokajania potrzeb informacyjnych na wszystkich poziomach organizacji.
- Zwiększenie bezpieczeństwa związanego z dostępem do danych oraz ograniczenie roli intuicji w podejmowaniu decyzji biznesowych w Abanku.

Dla zarządzających Abankiem istotne było, aby kultura organizacyjna w miarę możliwości była zgodna z celami strategicznymi w odniesieniu do całej działalności banku. Dlatego też jednym z założeń projektowych było stworzenie nowej, rozbudowanej wewnętrznie sieci komputerowej, która powstałaby na bazie tej działającej wcześniej w ograniczonym zakresie. Również połączenie centrali bezpośrednio z jednostkami oddziałowymi i siłami pracującymi w terenie miało potwierdzić zgodność ustalonych celów, usprawnić komunikację oraz w efekcie

[7] W wyniku takich powtórnych realizacji tego samego raportu ujawniono znaczne różnice w wynikach finansowych jednego z oddziałów, co zostało wykryte przez przypadek.

końcowym zwiększyć wartość Abanku dla wszystkich interesariuszy (w tym pracowników).

Kultura organizacyjna przed wdrożeniem

Abank był przykładem organizacji o silnej kulturze organizacyjnej, ze sztywnymi regułami postępowania. Przygotowanie do zmian było długie i mozolne, często nie przynoszące oczekiwanych rezultatów. Struktura organizacyjna była tradycyjna, silnie zhierarchizowana. Pełniła rolę raczej pasywną, podtrzymując stare przyzwyczajenia, utrwalając wykształcone postawy u pracowników i utrudniając zmianę istniejącej kultury organizacyjnej. Co prawda, w wyniku rozwoju nowych usług bankowych zmieniły się kryteria rekrutacji i struktura kwalifikacji pracowników, a wykonywana przez nich praca była dostosowywana do nowych potrzeb, jednak zmiany te były płytkie i ograniczały się często tylko do dodatkowych zajęć poza tymi przypisanymi do danego stanowiska.

Różnice kulturowe pojawiały się nie tylko na bazie grup pracowniczych, wynikających z charakteru wykonywanej przez nich pracy w poszczególnych departamentach, ale również w oddziałach i innych jednostkach nabytych w wyniku fuzji i przejęć. Oczywiście były pewne normy ustalone dla wszystkich i wszystkich obowiązujące jak np. sposób ubierania się[8], witania, prowadzenia zebrań, umawiania spotkań, przepływu dokumentów, szablonu tych dokumentów, itp. Jednak każdy departament był tak na prawdę enklawą, rządzoną właściwie swoimi prawami, podobnie jak oddziały oddalone od centrali, szczególnie te z przejętych banków. Rozbieżności wynikały nie tylko z podzielanych przez personel norm, założeń i wartości, ale również przez różnice technologiczne (różne oprogramowanie, różne systemy i sprzęt komputerowy połączony *nibynóżkami*).

Przykładem wielorakiego układu sił był departament marketingu, w którego skład wchodzi również dział sprzedaży. Konflikty najczęściej pojawiały się tam na tle niezrozumienia charakteru pracy sił sprzedaży przez *marketingowców*, tym bardziej, że w centrali mieli oni tylko swoich przedstawicieli, a tak na prawdę to pracowali w terenie i w oddziałach. Dodatkowo czasem wymagane było podjęcie natychmiastowych decyzji, które były niemożliwe pod nieobecność uprawnionych osób. Odczuwało się to szczególnie podczas pracy z kluczowymi klientami lub dużymi korporacjami, których obsługa była indywidualna i nie szablonowa. Było tam bardzo trudno skontaktować się bezpośrednio z przełożonym, chyba, że było się wcześniej umówionym. Wszelkie działania podlegały procedurom i wymagały potwierdzenia na piśmie.

---

[8] W Abanku obowiązywał strój tradycyjny, elegancki, w dominujących kolorach ciemnych: bordo, czerń, brąz. Należało nosić koszulę z białym lub jasnym kołnierzykiem. Kobiety mogli chodzić w spódnicach za kolana, z „pełnymi" butami –zakryte palce i pięta. Wyjątek od ustalonego ubioru stanowił piątek, gdzie raz w tygodniu dominowały stroje sportowe i wtedy tylko można było przyjść w dżinsach. Nie stosowanie tych zasad groziło sankcjami, ustalonymi najczęściej w poszczególnych departamentach, było to np. mniejsza nagroda roczna lub mniejsza premia.

Na kulturę organizacyjną wpływała również polityka kadrowa i szkolenia, kształtujące nie tylko umiejętności, lecz także postawy i motywacje pracowników. Wymagano od nich wysokich kompetencji, poszanowania norm i reguł. Preferowano zatrudnianie młodych ludzi, którzy w miarę upływu lat będą mogli awansować tworząc kadrę kierującą postępującą dokładnie jak poprzednia.

Jednak w wyniku planowanych zmian i rozpoczęciem projektu w Abanku zaczęto mówić o dużych zwolnieniach, w związku, z czym wśród pracowników wzrosła obawa o stanowiska. W wyniku tego również nasilił się dystans do pracy, podobnie jak niechęć do wymiany informacji i uczestniczenia w szkoleniach. W większości myślano jedynie o zachowaniu stanu rzeczy, bez wprowadzania zmian. Przy tym aspekty motywacyjne nie były do końca jasne, mówiono np. *szybciej będzie się wykonywać niektóre czynności*, co pracownik rozumiał: *to co wykonywało dwóch ludzi teraz może jeden*. Oczywiście nie mówiono nic o nagrodach lub też preferencjach dotyczących aktywnych uczestników zmian.

Władza w Abanku należała do naczelnego kierownictwa, co sprzyjało centralizacji i kontrolowaniu podległych kierowników średniego szczebla, a ponieważ organizacja była znacznie rozbudowana, to problemy komunikacyjne i błędy w interpretacji poleceń były na porządku dziennym. Zarząd komunikował się bezpośrednio z kadrą zarządzającą, a ta przekazywała dalej zarządzenia pracownikom. Jedynie kierownicy posiadali więcej kompetencji i uprawnień, dzięki czemu mogli spotykać się z kierownikami innych działów w celu wymiany informacji. Ponad tymi podziałami była komórka techniczna, która się mogła kontaktować ze wszystkimi departamentami bezpośrednio.

Poszczególne departamenty współpracowały ze sobą na styku wspólnych zadań, ograniczając jednak wymianę informacji do absolutnie koniecznych[9]. Oczywiście istniały struktury nieformalne, które rządziły się własnymi prawami, jednak nie były na tyle silne by oddziaływać bezpośrednio na podejmowane decyzje. Centralizacja decyzji wpływała zatem również na postawy pracowników, zniechęcając ich np. do przejawiania jakichkolwiek aktywności. I tak osoby ambitne, aktywne, wyróżniające się podejmowały wcześniej lub później decyzję o zmianie pracy[10].

Faza II – rok 2003

W wyniku przeprowadzenia badania pilotażowego zaproponowano by projekt podzielić na mniejsze części oraz aby wdrożenie rozpocząć od centrali i tam sprawdzić najpierw jego efektywność. Rozpoczęto przygotowanie do stworzenia zorganizowanego systemu gromadzenia, organizowania i analizy

[9] Przykładem była *niechciana* współpraca między departamentem marketingu i finansowym oraz między opisywanym wcześniej departamentem marketingu i wchodzącym w jego skład działem sprzedaży.

[10] W trakcie trwania projektu i wkrótce po aż 4 osoby z kierowniczych stanowisk odeszły do konkurencyjnych firm, gdzie za podobne stanowisko i udział w przygotowaniach do podobnego projektu uzyskały wyższy ekwiwalent pieniężny.

danych dotyczących klientów powiązanego z tworzoną równolegle *hurtownią danych*. Równocześnie opracowano procedury umożliwiające bieżące monitorowanie wyników osiąganych we współpracy z klientem, szczególnie w aspekcie finansowym. Włączono wyniki monitorowania do systemu motywacyjnego dla służb sprzedaży oraz udostępniono im wyniki tworzonych na bieżąco analiz w celu informacyjnego wsparcia prowadzonych przez nie działań w stosunku do odbiorców. Szeroko określane pojęcie *klienta*[11] Abanku uległo teraz podziałowi na:

1. *Klientów wspólnych* – do tej kategorii zaliczeni zostali ci klienci Abank, którzy będąc aktualnie korporacyjnymi klientami lub klientami *Private* byli jednocześnie klientami którejś z firm wchodzących w skład grupy kapitałowej Abank.
2. *Klientów potencjalnych* – do tej kategorii zaliczone zostały wszystkie osoby fizyczne i instytucje nie będące dotychczas klientami Abank, a które wchodziły w zakres zainteresowania Abanku jako potencjalni klienci. W kategorii potencjalnych klientów wyróżnione zostały dwie grupy:
   a. *wewnętrzna* – obecni klienci którejś z firm wchodzących w skład grupy kapitałowej Abank,
   b. *zewnętrzna* – osoby fizyczne i instytucje niebędące obecnie klientami Abank ani firm wchodzących w skład grupy kapitałowej, do której należy Abank.
3. *Grup klientów* zdefiniowanych poprzez obowiązujące w Abank kryteria przynależności do grupy.

W trakcie realizacji projektu wyodrębniono także następujące grupy użytkowników, których przedstawiciele zostali wybrani w każdym departamencie:

1. *Analitycy* –odpowiedzialni za realizację poszczególnych projektów analitycznych, z podkreśleniem, że zespół taki powinien zostać umiejscowiony w departamencie odpowiedzialnym za bezpośrednią opiekę nad wdrażanym rozwiązaniem, a docelowo w każdym dziale.
2. *Interpretatorzy biznesowi* –również docelowo w każdym dziale.
3. *Odbiorcy informacji* –kadra zarządzająca i inni uprawnieni pracownicy.

Modelowy CRM zaprojektowany dla Abanku, uwzględniał pięć perspektyw:

1. celów biznesowych– definicja odpowiedzi na pytanie *po co pracujemy*? Określenie strategicznych kierunków działań Abanku, do których zaliczono:

   - zwiększenie liczby klientów Abanku, poprzez: zatrzymywanie dotychczasowych i pozyskanie nowych klientów,

[11] Klient korporacyjny – rozumianych był jako każda instytucja tj. osoba fizyczna wykonująca działalność gospodarczą, osoba prawna, jednostka organizacyjna nieposiadająca osobowości prawnej (o ile posiada zdolność prawną), korzystająca z usług Abanku oraz Klient indywidualny – zwany dalej *Private* korzystający z usług Abanku

- wzrost wartości klientów dla Abanku, którego filarami są: aktywizacja klientów (*up-selling* i *cross-selling*) i wzrost przeciętnej zyskowności transakcji poprzez odpowiednie kształtowanie oferty (z uwzględnieniem optymalizacji alokacji klientów i produktów do kanałów dystrybucji).

2. procesów – specjalizacja działań, jakie powinny być podejmowane w celu realizacji dwóch poziomów celów biznesowych: procesy biznesowe, wspierające cele biznesowe oraz proces CRM, opisujący czynności ukierunkowane na efektywne wykorzystanie rozwiązania CRM na potrzeby Abank.
3. struktury organizacyjnej – perspektywa wskazująca, jakie jednostki organizacyjne (z punktu widzenia obszarów odpowiedzialności) oraz w jakiej wzajemnej relacji powinny być zaangażowane przy realizacji procesów. Ważnym elementem było tu zapewnienie odpowiednich kompetencji i uprawnień wymaganych do skutecznej realizacji procesów. Perspektywa ta była niezbędna w celu właściwego osadzenia procesów w strukturze organizacyjnej Abank.
4. platformy technologicznej – perspektywa opisująca szeroko rozumiane środki techniczne potrzebne do realizacji procesów.
5. kultury organizacyjnej – perspektywa opisująca *jak i z kim się pracuje*, określała warunki współpracy między poszczególnymi działami w Abanku, sposoby komunikacji, a także procedury postępowania między jednostkami organizacyjnymi w całej grupie finansowej Abanku.

W trakcie Fazy II wykonano również takie czynności jak:

- zreorganizowano procesy obsługi klienta w firmie. Kluczowym zadaniem w tym zakresie było uaktualnienie procesów i procedur operacyjnych poprzez uzupełnienie ich o działania wykorzystywane w modelu CRM.
- określono zgodny z wdrożoną koncepcją zakres współpracy i kompetencji działów funkcjonalnych firmy, przygotowano rozwiązania ograniczające potencjalne konflikty związane z wdrożeniem projektu.
- podjęto pracę nad systemem ewidencjonującym koszty klienta (metodą *ABC*) oraz prace związane z opracowywaniem strategicznej karty dokonań (tzw. *karta Kaplana*).

W Abanku na tym etapie występowała ograniczona możliwość oceny efektów podejmowanych działań. Na sytuację tę wpływały następujące czynniki:

- brak w większości przypadków wypracowanych, mierzalnych kryteriów oceny (w trakcie tworzenia);
- brak sformalizowanego i systematycznego procesu zbierania informacji zwrotnych z (innych niż służby sprzedaży) punktów kontaktu z klientem (np. lada, administracja systemami informatycznymi), w celu ich wykorzystania w procesie

monitorowania, brak nawyku tego działania;
- brak prowadzonych w sposób systematyczny badań postrzegania i oceny relacji projektu przez klientów (np. badania satysfakcji).

Na etapie Fazy II pojawiły się również problemy kompetencyjne. Wyznaczone grupy użytkowników podlegały rotacji, zmieniały się ich uprawnienia a w związku z tym dalej przepływ informacji nie był taki jak powinien. Problemem był również brak odpowiednich kwalifikacji. Administratorzy techniczni powinni umieć korzystać z programu, podobnie jak analitycy, którzy mieli umieć tworzyć raporty. Również przeprowadzone szkolenia potwierdziły, że nie wszyscy pracownicy mogą nabyć nowe wymagalne umiejętności, zatem przypisanie do konkretnych grup powinno odbyć się ponownie, aby nie powtórzyła się sytuacja, że nie zidentyfikowane osoby odpowiedzialne były za zorganizowany, systematyczny proces analizy i przetwarzania danych dostępnych w systemach Abanku.

Choć przepływ informacji nadal był sformalizowany (część funkcji kontrolnych przejął dział marketingu z działu controlingu, ze względu na szybki dostęp do raportów i możliwość interpretacji wyników) wiele pozytywnych aspektów aplikacji CRM nie zostało wcale wyjaśnionych lub też zrobiono to niedokładnie[12]. Służby handlowe zrozumiały wdrożenie jako odebranie im kompetencji i ograniczenie obowiązków, co w konsekwencji powodowało obawę przed obniżką pensji lub zwolnieniem z pracy. Nie mówiąc już o zagrożeniu *przejęciem klienta* przez konkurencję (niezadowolony z obsługi klient rozglądał się za inną, zbliżoną ofertą). Ustalono nowy termin ukończenia projektu (przesunięto datę ukończenia fazy realizacji) i rozpoczęto *ponowną sprzedaż* programu, tym razem stawiając nacisk na komunikację między departamentami oraz wśród pracowników.

Pod koniec prac na tym etapie ujawniły się również konflikty między poszczególnymi zespołami projektowymi, wynikające z problemów komunikacyjnych. Na przykład, po kilku miesiącach prac okazało się, że dwa zespoły (z firmy informatycznej X i Y) równolegle pracowały na tym samym zbiorze unifikując dane. Zdarzenie to przesunęło prace nad projektem o kolejny miesiąc, podczas którego głównie trwały rozmowy na szczeblu komitetu sterującego, a wszyscy wykonawcy mieli urlop od prac projektowych.

[12] Pracownicy wypowiadając się o CRM mieli na myśli wyłącznie aplikację informatyczną, choć byli uczeni pracować na niej, nie potrafili wymienić jej podstawowych funkcjonalności, sprowadzając je do kilku najprostszych, zapamiętanych ze szkoleń. Kwestie dotyczące usprawnienia działań i komunikacji, szczególnie dotyczące planowania wyników sprzedaży nie były przez nich poruszane.

Kultura organizacyjna podczas Fazy II

W trakcie fazy realizacji grupa pracowników przekonanych o konieczności wdrożenia początkowo się zwiększyła (2/3 pracowników deklarowało chęć współpracy przy projekcie). Liczne usprawnienia i możliwości zaoszczędzenia czasu pracy pozwoliły na łatwiejsze uporanie się z obowiązkami. Jednak w miarę trwania projektu obowiązków zaczęło przybywać, zwłaszcza nowych, które nie koniecznie były dobrze opanowane w działaniu. Czynności te nie były poparte również żadnymi narzędziami motywacyjnymi (np. w celu zwiększenia efektywności pracy). Przeprowadzono co prawda szereg obowiązkowych szkoleń dla wyznaczonych pracowników (zatem z góry traktowali te zajęcia niechętnie) ale dotyczyły one wyłącznie nauki korzystania z programu.

Co prawda podjęto dyskusję o zmianie struktury organizacyjnej, która mogłaby dodatkowo wpłynąć na umacnianie się rozpoczętych w poprzednim etapie zmian kulturowych, jednak skończyło się to na wytyczeniu bliżej nieokreślonych czasowo działaniach[13]. Nie zastosowano sugestii zmian struktury organizacyjnej, określającej zakres i podział kompetencji oraz odpowiedzialności poszczególnych jednostek organizacyjnych podczas realizacji zadanych procesów, zatem wypracowane wcześniej nowe zachowania okazały się krótkotrwałe.

Realizacja procesów CRM potrzebowała wsparcia ze strony technicznej, której podstawowymi wymaganiami było jedno, centralne, spójne źródło danych, a co za tym idzie scentralizowany, wszechstronny system analityczny oraz łatwa dystrybucja wyników analiz do wszystkich zainteresowanych i uprawnionych stron[14]. Nowe grupy pracowników, tworzone dla odpowiednich funkcji były połączone między innymi stopniem dostępu do informacji. Część z nich miała uprawnienia administratorów, część analityków tworzących raporty, a w końcu część tylko mogła odczytywać dane, bez możliwości wprowadzania zmian. Owe podziały zaczęły owocować kontaktami nieformalnymi.

Przejrzyste przypisanie odpowiedzialności do jednostek organizacyjnych, skutkowało skuteczniejszą realizacją zadań dzięki centralizacji czynności stanowiących elementy danego procesu. To sprawiło wzmocnienie zmian zachodzących w kulturze organizacyjnej. Tworzone zespoły, nowe funkcje i nowe grupy spowodowały powstawanie nowych zależności i więzi, które odpowiednio ukierunkowane (i podtrzymane) mogły poprawić wymianę informacji (poprawić

[13] Kierownictwo deklarowało chęć wdrażania kolejnych modułów pod warunkiem sprawdzenia w działaniu już zainstalowanych, nie sprecyzowało jednak sposobu sprawdzania oprogramowania (czy będzie inny, czy taki samo jak wyznaczony przez dostawcę) ani kiedy będzie zrobiony.

[14] Dane pochodzące z systemów Abanku oraz badań rynku były teraz zbierane, przechowywane, organizowane i udostępniane przez jedno, główne repozytorium danych. Dane o charakterze przejściowym, np. pośrednie wyniki analiz, wykorzystuje się do tworzenia raportów, sprawozdań, planów, a następnie po kolejnych etapach obliczeń, przechowywuje się w lokalnych repozytoriach danych.

komunikację między departamentami oraz oddziałami) oraz mogły pomagać przełamywać bariery tworzone wśród niechętnych projektowi pracowników.

Faza III – rok 2004

Na tym etapie zakończono prace związane z wymianą sprzętu, zakupem kompletnego oprogramowania i instalacją. Wyznaczone osoby ukończyły również szkolenia. Rozpoczął działanie jeden główny system oparty na wspólnej bazie, czyli na *hurtowni danych*. Zakończono również prace nad siecią Intranet i zewnętrznym portalem internetowym.

Zaczęła wychodzić wewnętrzna gazetka (rozsyłana na *adresy mailowe*) informująca nie tylko o bieżących działaniach kadry zarządzającej Abanku, o tym co się dzieje w oddziałach, ale również o postępach prac w projekcie. Uruchomiono system zabezpieczenia dostępu do danych i nowy sposób autoryzacji. Przy opracowywaniu interfejsów użytkowników systemu zwrócono szczególną uwagę na udzielanie kompetencji i dostępu do danych (aby nie tworzyły się konflikty typu: *dotychczas miałem dostęp, a teraz ma go tylko mój kolega, który się wcześniej tym nie zajmował*). Każdy pracownik miał odtąd dostęp tylko do zasobów, na których pracuje. Jego profil mógł być zainstalowany właściwie na każdym komputerze w Abanku (bez dostępu do sieci zewnętrznej). Po zsynchronizowaniu danych można było kontynuować pracę bez potrzeby odszukiwania zapisanych gdzieś w systemie plików.

Centralizacja wykonywania analiz i gromadzenia wiedzy, zapewniła dogodny dostęp do informacji, bez dodatkowego obciążania departamentów operacyjnych. Pojawiła się możliwość uzyskania różnego rodzaju danych z jednego źródła oraz łatwość wyodrębnienia interesujących użytkownika danych (na różnym poziomie agregacji). Poprawiło się także tempo przetwarzania danych w informacje oraz rozpowszechnienie wykorzystania rozwiązania CRM w Abanku. Ujednolicono pojęcia biznesowe oraz nowe sformułowania krążące w obiegu poprzez ogólnodostępny wirtualny *słownik projektowy*.

Ograniczono rolę intuicji w podejmowanych decyzjach. Zostały one wsparte raportami analitycznymi, o znacznie poszerzonej funkcjonalności. Umożliwiły one między innymi obliczanie dynamiki określonych zjawisk, a co za tym idzie tworzenie określonych rekomendacji.

Aplikacja CRM wprowadziła także usprawnienia w komunikacji, połączyła wiele istniejących baz danych umożliwiając prowadzenie monitoringu klientów trafiających do różnych oddziałów tej samej organizacji. Efekty były widoczne nie tylko przy obliczaniu wartości klientów i prognozowaniu ich zachowań, ale również przy generowaniu różnego rodzaju raportów finansowych (w tym do sprawozdawczości obowiązkowej, których czas wykonania zmniejszył się z kilku tygodni nawet do kilku dni).

Podsumowanie

Abank był przykładem dużej polskiej organizacji, gdzie wielość stopni zarządzania oraz pozostałości po starym systemie gospodarczym utrudniały przeprowadzanie jakichkolwiek zmian. Na dzień dzisiejszy trwają przygotowania do zakupu kolejnych modułów CRM i do utrwalenia zmian kulturowych[15] w związku z wprowadzoną niedawno koncepcją zarządzania relacjami.

W Abanku CRM był kojarzony najczęściej jako narzędzie i dlatego cały projekt przebiegał głównie w myśl założeń typowego projektu informatycznego. Sugerowane przez grupę doradczą kulturowe zmiany głębsze, dotyczące zarówno stanowisk, czynności, norm, jak i procedur (jak choćby tych związanych ściśle z obsługą klienta) zostały wprowadzone częściowo (opisano stanowiska na nowo, określono – dodano nowe obowiązki), a istotne zmiany przełożono w czasie, tłumacząc się wyczerpaniem środków budżetowych na ten cel.

W wyniku przekroczenia terminów czasowych (projekt początkowo był szacowany na jeden rok) oraz dynamicznymi zmianami na rynku krajowym (opisanych w Fazie I) w tym ogólnej stagnacji rynku oraz strat finansowych odnotowanych w Abanku, sprawdziły się przepowiednie o likwidacji części stanowisk, szczególnie tych w centrali[16]. Silna kultura organizacyjna Abanku, który cały czas był pod wpływem zmian ustrojowych (jeszcze nie została wymieniona kadra zarządzająca z poprzedniego ustroju gospodarczego dyktująca warunki pracy) była przykładem, gdzie niezmiernie trudno jest wprowadzić jakiekolwiek zmiany do ustalonego trybu działania. Dlatego też CRM był tu wdrażany stopniowo, kolejnymi modułami. Co prawda, pracownicy (tylko pierwszoliniowi) pracowali w myśl koncepcji CRM, lecz teraz nauczyli się posługiwać nowym sprzętem oraz rozpoczęli działania według nowych procedur (dzielenie się danymi, wiedzą).

Jeśli zmiany przeprowadzone w poszczególnych fazach projektu mają być trwałe to stare stanowiska i funkcje muszą być stopniowo wypierane przez nowe, wypracowane podczas projektu. Szczególnie wtedy, kiedy większość pracowników się do nich przekonała i kiedy system informatyczny spowodował sprawniejsze i szybsze wykonywanie dotychczasowych zadań[17]. Aplikacja CRM usprawniła bowiem pracę, skracając czas generowania i tworzenia raportów, odszukiwania istotnych informacji z baz danych oraz umożliwiła lepszą komunikację między poszczególnymi działami[18] oraz klientom Abanku.

---

[15] W związku z charakterem przeprowadzonego badania można było opisać tylko część widocznych poziomów kultury w analizowanej organizacji. Do dalszej interpretacji potrzebne są dodatkowe badania, uwzględniające psychologiczny charakter zachowania się grupy w trudnych i konfliktowych sytuacjach.

[16] Mówiło się, że właściwie co piąta osoba straciła pracę, w rzeczywistości co siódma.

[17] Np. opracowanie tygodniowych zestawień sprzedaży zostało zmniejszone z 2 dni na 4 godziny

[18] Jednak i tutaj sukces projektu był połowiczny, ponieważ pracownicy bardziej pojętni, zaczęli wykorzystywać swoje umiejętności, zarówno do uzyskania przewagi nad innymi w

Spis literatury

1. Babbie E., *Badania społeczne w praktyce*, Wydawnictwo Naukowe PWN, Warszawa 2004
2. Bugaj J., *Zarządzanie relacjami z klientem (Customer Relationship Management)*, „Organizacja i Kierowanie", nr 3/2003
3. Cameron K.S. Quinn R.S., *Kultura organizacyjna—diagnoza i zmiana*, Oficyna Ekonomiczna, Kraków 2003
4. Forsyth R., *Re-evaluating CRM cliches No 2: A 360 degree view of customer*, www.crm-forum.com, (2003-03-24)
5. Konecki K., *Kultura organizacyjna. Główne perspektywy analityczno-badawcze.*, www.centrumwiedzy.edu.pl (2004-11-09)
6. Materiały projektowe Abank (2004)
7. Peppers D., Rogers M., *Letter to Wall Street: start thinking about Long-term Value*, www.peppersandrogersgroup.com, (2004-03-13)
8. Sułkowski Ł., *Czy jest możliwe kształtowanie kultury organizacyjnej*, „Organizacja i Kierowanie", Nr 4 (106), rok 2
9. Szymura-Tyc M., *Ewolucja modelu przewagi konkurencyjnej a wartość firmy*, www.zti.com.pl/instytut/pp/referaty, (2005-04-27)
10. Trompenaars F., Hampden-Turner Ch., *Siedem wymiarów kultury*, Oficyna Ekonomiczna, Kraków 2002

---

kontaktach z przełożonymi (lepszego iszybszego wykonywania pracy) jak i do wykorzystania tej przewagi w stosunku do innych pracowników.

ROZDZIAŁ XIII

WSPOMAGANIE DZIAŁALNOŚCI PRZEDSIĘBIORSTWA PRZY POMOCY TECHNOLOGII INFORMACYJNEJ

Janusz K. GRABARA

Wykorzystanie wiedzy i umiejętności zarówno kadry zarządzającej, jak i programistów pozwala na transformację IT w przedsiębiorstwie, dlatego też kadra zarządzająca i managerowie IT w wielu znanych przedsiębiorstwach pracują razem, aby usprawnić wspomaganie działalności przedsiębiorstwa przy pomocy technologii informacyjnej. Efektem współpracy jest to, że możliwe są: obniżenie kosztów IT, ominięcie ograniczeń nieelastycznych systemów wspomagania oraz zmiany w biznesie służące podniesieniu jego udziału w zarządzaniu IT[1].

Przedsiębiorstwa, które zastosowały nowe systemy IT w późnych latach 90 wiedzą, że nieefektywne zaangażowanie biznesu w programy IT może zwiększyć zwroty. Aby uniknąć powtarzania tego błędu, biznes i managerowie IT zaczęli gromadzić systemy IT w „domeny" – ustalać aplikacje i bazy danych, które dają się zarządzać jako jednostki dla biznesu – zamiast zarządzać nimi na poziomie aplikacyjnym lub organizować je ze względu na rodzaj komputerów i systemy operacyjne na jakich działają aplikacje. Przedsiębiorstwo może na przykład połączyć systemy, które zawierają informacje o klientach i ponownie przeanalizować połączenia między nimi w celu uniknięcia zawiłych systemów połączeń wewnętrznych, z możliwymi duplikatami i ponownymi próbami niekończących się aplikacji, dzięki czemu łączone są koszty utrzymywania i tworzenia[2]. Pracownikom IT łatwiej jest zmienić lub zeskalować takie systemy, dzięki czemu mogą wspomagać zmiany biznesowe szybciej i mniej kosztownie. Co więcej, nowe podejście eliminuje źródło napięcia między biznesem a IT.

Transformowanie architektury IT w sposób ewolucyjny jest skuteczniejsze niż wprowadzanie zmian w poszczególnych systemach lub zastępowanie ich w całości. Przedsiębiorstwa mogą również stosować oszczędności wynikające z wykorzystania finansowania jednego projektu do następnego.

Zespół wykonawców biznesu i IT, w oparciu o decyzje przedsiębiorstwa w sprawach obecnych wymagań biznesowych i przyszłej strategii, zdefiniował domeny, określił jak systemy mają być grupowane oraz przedstawił jak powinny się komunikować między sobą. Eksperci IT użyli wtedy centralnej integracji opartej na technologiach middleware aby utworzyć konieczne połączenia. Celem integracji było przekształcenie czegoś co było zawiłe i skomplikowane – z wieloma aplikacjami i bazami danych w architekturze – w system modularnych

---

[1] Skowron-Grabowska B., *Uwarunkowania i kierunki rozwoju systemów informatycznych a logistyka,* [w:] „Informacja i komunikacja w logistyce" Wyd. AE Katowice 2005, s.187

[2] Cholewa M, Selinger B., *Wspomaganie procesów biznesowych w rozproszonym środowisku,* [w:] „Komputerowo zintegrowane zarządzanie" WNT Warszawa 2004, s.190

i logicznych bloków z minimalnymi przewodami łączącymi je ze sobą. To podejście przypomina umowę „kup jedno – drugie weź za darmo": naprawiasz swoją architekturę IT aby obniżyć koszty i uzyskać elastyczność, a jeśli zrobisz to dobrze, osiągniesz biznesową własność IT w nagrodę. W dużych przedsiębiorstwach taki proces może trwać ponad dwa lata a koszty mogą przekraczać 10 milionów dolarów. Każda firma wspomagana przez IT, które przyczynia się do wzrostu kosztów dodawania nowych aplikacji powinna przemapować i zmienić jego architekturę, co pozwala na zmniejszenie kosztów integracji wszystkich aplikacji o 60 do 70 procent w średnim okresie. Przedsiębiorstwo powinno przynajmniej przeanalizować zmianę architektury IT, jeśli napotyka na jakiekolwiek wyzwanie biznesowe, takie jak: zmiana strategii, krzyżowa integracja jednostek biznesowych, fuzja, nabycie wymagające wspomagania ze strony IT[3]. Nawet przedsiębiorstwa, które nie postrzegają technologii informacji jako problemu powinny rozważyć celowość zmian w zakresie IT, aby przygotować się na przyszłe wyzwania.

Architektura IT jest ważna dla kadry zarządzającej, ponieważ może wspomagać lub utrudniać włączanie nowych procesów, produktów i kanałów. Problemy spowodowane słabą architekturą IT są szczególnie widoczne w sektorach takich jak bankowość, logistyka i telekomunikacja, gdzie IT jest podstawą działalności[4]. Wiele banków na przykład uważa, że ich jednostki IT nie mogą efektywnie wspomagać ofert wielokanałowych, ponieważ produkty i kanały są obsługiwane przez oddzielne systemy IT które komunikują się tylko w ograniczonym zakresie. W innych przemysłach intensywnie korzystających z IT, takich jak produkcja czy handel detaliczny, menadżerowie uważają, że dostosowywanie IT do zmieniających się wymagań zabiera za dużo czasu, ich koszty są za wysokie lub zawodzą, nie przynosząc oczekiwanych efektów. Wiele takich problemów pojawia się w architekturze IT. Pojawia się rozdźwięk między tym czego biznes wymaga od niej, a tym co departamenty IT mogą dostarczyć bez angażowania dużych kosztów rozwojowych. Wykonawcy biznesu, którzy często wierzą że menadżerowie IT nie rozumieją ich wymagań i nie dostarczają rzeczywistej wartości, nie lubią powierzać pracy nad projektami IT swoim dobrym pracownikom. Z ograniczonym wkładem z biznesu, tworzący IT zmagają się z dostosowaniem najlepszych możliwych do osiągnięcia systemów, ale bardziej z perspektywy technologicznej niż biznesowej.

Dla pracowników IT architektura technologii informacji jest najważniejsza. W większości przedsiębiorstw, jednostki IT dodają nowe systemy poprzez uzupełnianie ich bezpośrednio z nimi połączonymi systemami. Jest to skuteczne, kiedy systemy nie muszą być połączone z kluczowymi aplikacjami lub bazami danych. Ale kiedy takie połączenia są konieczne, uzupełnianie jest

---

[3] Laartz J., Monnoyer E., Scherdin A., *Designing IT for business,* [w:] "The Mc Kinsey Quarterly", 2003 No.3, s.79

[4] Gołuchowski J., *Charakterystyka technologii informatycznych wspomagających współpracę elektroniczną,* [w:] „Informacja i komunikacja w logistyce" Wyd. AE Katowice 2005, s.29

niewystarczające, o czym często zapominają pracownicy IT, którzy przykładają zbyt mało uwagi do tworzenia nowego systemu wraz z jego wewnętrznymi elementami pasującego do całego obrazu IT, ponieważ wymaga to angażowania dużych środków pieniężnych i jest czasochłonne. Metoda uzupełniania już istniejących systemów tylko pozornie jest tańsza i szybsza, w dłuższym okresie czasu dochodzi do kumulacji i powstawania złożonych systemów trudnych i kosztownych w zarządzaniu, utrzymywaniu, zmienianiu i skalowaniu.

Niektóre przedsiębiorstwa próbowały rozwiązać problemy z architekturą poprzez zastępowanie ważnych części przez nagłe nowe wdrożenia. To podejście można przedstawić jako zastosowanie rozbudowanych systemów planowania źródeł w przedsiębiorstwie, czyli monolitycznych, zintegrowanych aplikacji, które zastępują wiele innych, zazwyczaj w obszarach zarządzania kadrami, operacjami i rachunkowości, albo w łańcuchu dostaw. Odchodzenie od starych systemów i roszady w ich połączeniach jest kosztowną i ryzykowną strategią, która przynosi różne efekty i zazwyczaj nie pomaga minimalizować dystansu między biznesem a IT. Kilkakrotnie zaobserwowano przypadki, w których biznes wprowadzał bezpośrednio od ogromne złożone projekty programów tylko po to, aby zużyć nadwyżki czasu i budżetu, później konieczna jednak była kontrola takiego IT. Coraz częściej przedsiębiorstwa dokonujące naprawy i ulepszania swoich systemów IT przyczyniają się do zmniejszenia dystansu między kulturą biznesu a IT.

Przedsiębiorstwa które dokonały przeglądu swoich architektur IT zgodnie z ewolucją zarządzania osiągnęły obydwa cele. Przez zmodularyzowanie, dezorientacja w systemach IT zamieniona została w logiczne i przejrzyste domeny prowadzone przez biznes, co umożliwiło managerom biznesu kontrolowanie kierunków technologii informacyjnej. Domeny dostarczają fundamentu dla własności biznesu i możliwości rachunkowych dla IT.

Tak szeroka metodologia ewolucyjna nie polega na przypadkowych zmianach lub na ryzykownych gwałtownych rewolucjach, tylko pozwala na wybrane połączenia i skupienie się na zastępowaniu przestarzałych systemów. Wydaje się że jest to nie tylko mniej ryzykowne i bardziej efektywne kosztowo niż wcześniejsze podejścia, ponieważ zaczyna się od zużywania istniejących aktywów i przekształcenia ich krok po kroku. Sztuką jest aby zdobyć krótkoterminowe korzyści podczas tworzenia stałego obrazu IT, który jest elastyczny nawet dla wspomagania potrzeb biznesowych, bez kosztownych ulepszeń.

Przedsiębiorstwa w wielu przemysłach – włączając bankowość, rynek energii, ubezpieczenia, logistykę, produkcję, detal i telekomunikację – podążają tą drogą grupując aplikacje i bazy danych w domeny biznesowe. Wymaga to wiedzy o zewnętrznych procesach biznesowych przedsiębiorstw i potencjalnych strategiach, dlatego też zadania te powinna realizować kadra zarządzająca.

Zespół zajmujący się tymi zagadnieniami powinien być ustalony, wspomagany i kontrolowany przez członków zarządu – na przykład głównego kierownika operacyjnego (COO) – oraz zawierać starszych wykonawców, managerów jednostek biznesowych i specjalistów od procesów i IT. Na przykład bank detaliczny utworzył zespół zawierający managerów jednostek biznesowych,

specjalistów od procesów z jednostek biznesowych, managerów IT i starszych architektów IT. W ważnej firmie logistycznej remont architektury IT został zainicjowany przez COO, który prowadził projekt razem z głównym kierownikiem informacyjnym (CIO). Zespół przedsiębiorstwa (który zawierał managerów operacyjnych z różnych krajów, operatorów sieci i hubów oraz architektów IT) najpierw określił domeny i wymienił połączenia za pomocą jakich miała się odbywać komunikacja, a następnie określił w kolejności jakie nowe połączenia mają być utworzone.

Przyjmując strategię technologii informacyjnej jako punkt początkowy, przedsiębiorstwo musi zadać sobie pytanie czego oczekuje od IT, powinno ocenić aktualne możliwości dla IT i zdecydować jaki kierunek chce obrać. Zespół w banku zaczął od analizowania zachęt dla biznesu, obecnych projektów IT, planów procesów biznesowych oraz aplikacji i baz danych IT, zachowując jednocześnie pewne szczegółowe i palące problemy. Bank miał zamiar, na przykład, zaoferować swoje produkty przez kilka kanałów – w każdym ze swoich oddziałów, centra telefoniczne i Internet. Ale jego systemy IT nie mogły utworzyć i używać tych samych informacji i procesów biznesowych jednocześnie, ponieważ były utworzone tylko dla jednej podstawy z ograniczonymi możliwościami komunikacji. Zespół szukał wspólnych cech w tych elementach IT – na przykład klient potrzebował informacji o więcej niż jednym produkcie – i wkrótce zdecydowano o reorganizacji systemów banku w 6 domen. Zawierają one domenę kanałową (z subdomenami takimi jak: odgałęzienia, centrum telefoniczne i Internet), domenę produktów (konta, kredyty, opłaty) i domenę klienta (wspomaganie sprzedaży, informacje dla klientów). Wszystkie są związane z wielokanałową ofertą banku.

Następnie zespół ustalił instrukcje wymiany informacji wśród domen. Specjaliści IT powołali się na takie interfejsy jak „usługowe", które kontrolują sposób obsługi różnych części biznesu jednocześnie. Oddziały banku, centrum telefoniczne i Internet na przykład potrzebują standardowego serwisu umożliwiającego im dostęp do każdego potrzebnego profilu obecnego w nowej subdomenie klienta. Jest to łatwo dostępne poprzez wywoływanie usługi „uaktualnienie informacji o kliencie". Inne usługi bankowe mogą zawierać „śledzenie używalności kanałów przez klienta" czy „szacowanie dochodowości produktu".

Kilka setek standardowych usług biznesowych może zastąpić dziesiątki tysięcy dostosowanych interfejsów pomiędzy aplikacjami i bazami danych. Te usługi przynoszą ważne korzyści dla biznesu i IT ponieważ muszą być zazwyczaj wdrażane tylko raz, są dostępne dla wszystkich domen, pomagają unikać duplikacji, tak więc umożliwia to redukowanie kosztów i złożoności oraz czyni biznes bardziej elastycznym. Zmiany usług w biznesie różnią się od zmian w technologii, co więcej systemy IT w domenach muszą być modyfikowalne lub zastępowalne (jako efekt na przykład zmiany technologicznej lub fuzji), usługi odzwierciedlają stabilne relacje biznesowe między domenami i zazwyczaj pozostają takie same. Managerowie IT w domenach mogą dlatego też zmieniać system IT bez konieczności ponownego wgrywania setek interfejsów do innych

systemów.

Wpływ modernizacji architektury IT na zarządzanie przedsiębiorstwem jest istotny dla przedsiębiorstw w wielu sektorach. Na przykład operator europejskiej telefonii kablowej przebudował swoje IT przez utworzenie usług fakturowania i informacji dla klientów, które mogą być ponownie wykorzystane do wspomagania nowych produktów. Dostawca usług internetowych, który wydawał 70% budżetu na rozwój aplikacji do modyfikowania cech i tylko 30% na rozwój aktualnych produktów zaadaptował podejście zarządzania ewolucyjnego i teraz te proporcje zostały odwrócone. Firma wydobywająca ropę naftową, która zmieniła swoją architekturę IT następnie stosunkowo łatwo połączyła to IT z drugim systemem przedsiębiorstwa, z którym dokonała fuzji.

Prowadzący biznes IT mogą łatwo przekształcać się we właścicieli biznesu nawet kiedy pierwsze domeny zostały zmodyfikowane. W kilku przedsiębiorstwach, powstałe wzajemne relacje podczas pracy na wczesnych etapach projektów zmieniających architekturę IT zostały sformalizowane w nowym modelu IT na zasadach nadzoru i kontroli. W tym rozdzieleniu, managerowie biznesu mają więcej bezpośredniej odpowiedzialności za zarządzanie domenami i połączeniami, podczas gdy managerowie technologii informacyjnej są nadal odpowiedzialni za dostarczanie wspomagania przez IT do domen i usług.

Jednym z najbardziej ciekawych przykładów tego podejścia może być globalne przedsiębiorstwo logistyczne, które zmodyfikowało architekturę swojego IT po fuzji, przemapowało systemy w układ globalnych domen takich jak produkcja i integracja klientów.[5] Manager biznesowy bliżej powiązany z daną domeną jest odpowiedzialny za jego architekturę; główny kierownik operacyjny na przykład odpowiada za ogólnie obowiązującą architekturę grupy sterującej domeną produkcji. Ten komitet sterujący, który zawiera managerów IT i biznesowych wraz z reprezentantami innych funkcji, których interfejsy domen, określają wspólnie i rozwijają domeny IT i interfejsy.

Na poziomie centralnym, COO (lub czasami CFO) dzielą całkowitą odpowiedzialność za architekturę IT przedsiębiorstwa z grupą ją tworzącą czyli zazwyczaj CIO i innymi ludźmi z IT oraz kierownikami biznesowymi. Kierownicy biznesowi zajmują stałe stanowisko w sprawie obciążania IT podejmowaniem decyzji – system zapewniający że decyzje IT będą kontynuowane nawet po tym jak projekty zostaną zakończone. CIO zazwyczaj ciągle ma odpowiedzialność operacyjną za tworzenie aplikacji i infrastruktury IT oraz za zarządzanie całym budżetem IT. Ale managerowie biznesowi i liderzy mają dużo większą kontrolę nad wkładem IT który bezpośrednio oddziałuje na biznes oraz nad tym czym trzeba zarządzać, zmieniać i inwestować w technologię aby uzyskać rezultaty dla biznesu.

Era własności biznesowej IT w Polsce jest obecnie w stadium początkowym.[6] Niektóre przedsiębiorstwa w różnych sektorach podejmują ważne

---

[5] Laartz J., Sonderegger E., Vinckier J.,*The Paris guide to IT architecture,* [w:] "The Mc Kinsey Quarterly", 2000 No.3,s. 118

[6] Praca pod red. M. Niezgódki, *Proponowane kierunki rozwoju społeczeństwa informacyjnego w Polsce do 2020 roku,* Min. Nauki i Informatyzacji, Warszawa 2004, s. 5

kroki aby ewolucyjnie zarządzać swoimi architekturami IT, a efekty zachęcają innych do podążania ich śladem. Następne wyzwanie to przemiana i ochrona przed powrotem do złych nawyków rozdzielania biznesu od IT.

Literatura

1. Cholewa M, Selinger B., Wspomaganie procesów biznesowych w rozproszonym środowisku, [w:] „Komputerowo zintegrowane zarządzanie" WNT Warszawa 2004
2. Gołuchowski J., Charakterystyka technologii informatycznych wspomagających współpracę elektroniczną, [w:] „Informacja i komunikacja w logistyce" Wyd. AE Katowice 2005
3. Laartz J., Monnoyer E., Scherdin A., Designing IT for business, [w:] "The Mc Kinsey Quarterly", 2003 No.3
4. Laartz J., Sonderegger E., Vinckier J.,The Paris guide to IT architecture, [w:] "The Mc Kinsey Quarterly", 2000 No.3
5. Praca pod red. M. Niezgódki, Proponowane kierunki rozwoju społeczeństwa informacyjnego w Polsce do 2020 roku, Min. Nauki i Informatyzacji, Warszawa 2004
6. Skowron-Grabowska B., Uwarunkowania i kierunki rozwoju systemów informatycznych a logistyka, [w:] „Informacja i komunikacja w logistyce" Wyd. AE Katowice 2005

CZĘŚĆ 3

ROZDZIAŁ XIV

TECHNOLOGIA INFORMACYJNA W PROCESACH NIELINIOWEGO MONTAŻU FILMOWEGO I TELEWIZYJNEGO

Marcin DUDZIŃSKI

„Montaż jest wszystkim" – stwierdzenie autorstwa Lwa Kuleszowa, rosyjskiego reżysera i teoretyka filmu, jest już niemal tak stare jak kino. Stwierdzenie to najlepiej jednak obrazuje rolę, jaką we współczesnym filmie, dokumencie, materiale telewizyjnym czy jakiejkolwiek innej formie filmowej czy telewizyjnej odgrywa montaż.

Scenarzysta pisze historię, reżyser oczami swej wyobraźni dokonuje inscenizacji wybranych ze scenariusza sytuacji i tworzy z nich rzeczywistość, operator rejestruje za pomocą kamer zaistniałe fakty, czyniąc kamerę głównym świadkiem wydarzeń, jednak to właśnie montażysta nadaje końcowemu dziełu klimat, charakter i atmosferę, łącząc poszczególne, pozornie nie pasujące do siebie fragmenty, w jedną, zwartą historię, spójną całość.

Rola montażu w produkcji filmowej i telewizyjnej jest obecnie nie do przecenienia. To właśnie montaż wzmacnia intencje twórców, pokazuje widzowi świat od strony dotychczas mu nieznanej, montaż sugeruje zrozumienie historii, zaciekawia, nadaje nowe znaczenia, buduje napięcie, nakazuje widzowi uwierzyć w to, co obserwuje. Dobrze wykorzystany, jest niezwykłym narzędziem pozwalającym poskładać w całość różnorodne, nieraz jakże odmienne fragmenty, stwarzając jednocześnie wrażenie ich ciągłości.

Biorąc pod uwagę niezliczone możliwości zabawy obrazem i dźwiękiem, ogromną wyobraźnię twórców, wielość znaczeń jaką nadać można poszczególnym zdarzeniom, nie sposób przecenić wartości jaką, wraz z dokonującym się postępem i dynamicznym rozwojem rynku komputerowego, dają nowe techniki montażu. Wpływ postępu technologicznego, skomputeryzowania życia codziennego, nie pozostał bez echa również, a może przede wszystkim jeśli chodzi o nowoczesne techniki montażu.

Do niedawna jedynym stosowanym rodzajem montażu wideo był tzw. montaż liniowy. Przegrywanie z jednego magnetowidu na drugi wybranych wcześniej ujęć, na czym w największym skrócie polegał ten sposób montowania, do dziś stosowany jest nawet m.in. w Telewizji Polskiej. Sposób ten, mimo całego wachlarza możliwości uatrakcyjnienia montowanego materiału wideo, mocno ogranicza montażystę, wydłuża czas pracy i obliguje do wcześniejszego „montowania w myślach", gdyż wyklucza powrót i dodanie pewnej sekwencji w już zmontowanym fragmencie końcowego materiału. Oczywiście montaż liniowy pozwala na wprowadzanie efektownych przejść m.in. poprzez zastosowanie tzw. miksera wideo, korekcji obrazu, czy wprowadzenia napisów, dzięki użyciu edytora napisów.

Wady i niedogodności z jakimi wiąże się jednak montowanie w sposób

liniowy są jednak w obecnych warunkach nie do przeskoczenia. Niemożność dokonywania większych zmian w zmontowanym materiale, bez ponownego wykonania wielu czynności, mniejsza precyzja i mniejsze możliwości oprawy oraz wysokie koszty, sprawiają, że montaż liniowy powoli odchodzi do lamusa.

Rys1. Zestaw do montażu liniowego. Stanowisko montażowe TVP S.A. Zbiory własne autora.

Pojawienie się pierwszych komputerów, bardzo szybko zaowocowało nowymi możliwościami obróbki i przetwarzania obrazu. Komputerowy montaż wideo, zaczęto określać, dla odróżnienia od poprzednika, mianem montażu nieliniowego. Możliwe stało się dzielenie obrazu na ujęcia, dodawanie efektów, napisów, udźwiękowienie, dokonanie korekt obrazu, eliminacja niedoskonałości i zapisanie zmontowanego materiału na kasecie wideo, już wcześniej mając pewność, że tworzony materiał, zarówno w postaci surowej, jak i po montażu jest zapisany na twardym dysku komputera. Podstawową zaletą tego montażu jest konieczność dysponowania tylko jednym magnetowidem podająco-nagrywającym.

Montaż nieliniowy pozwala więc na taką edycję plików audio lub wideo, która umożliwia dostęp do dowolnego miejsca materiału źródłowego, bez względu na jego położenie w tzw. surówce, czyli pierwotnym pliku.

O ile na starcie nowy system montażu nie dość, że był skomplikowany i w dużej mierze niezrozumiały, drogi i tym samym słabo dostępny, to wraz ze stałym rozwojem i informatycznym postępem, dziś nawet przeciętnej, średniej klasy komputer, wyposażony w odpowiednią kartę montażową i specjalistyczne oprogramowanie umożliwia dokonywanie nawet skomplikowanego montażu. Z całą stanowczością można więc powiedzieć, że montaż wszedł pod strzechy i dziś każdy może zmontować swój własny film z wakacji, urodzin czy wesela. Edycja nieliniowa jest obecnie najpopularniejszą metodą montażu w produkcji filmowej i

telewizyjnej.

Zagłębiając się w szczegóły nowoczesnej techniki montażu, należy dokładniej omówić wszelkie związane z nim aspekty, poczynając od wymagań sprzętowych, poprzez oprogramowanie, na funkcjach i dostępnych efektach kończąc.

Wymagania sprzętowe

By móc montować, należy posiadać kartę przeznaczoną do montażu nieliniowego. Karty te można podzielić na trzy podstawowe kategorie:

1. Karty analogowe
2. Karty cyfrowe
3. Karty cyfrowo-analogowe

Ad 1. Karty analogowe wykorzystują złącza analogowe do przechwytywania i zapisu obrazu i dźwięku ze źródła. Często wymagają dodatkowej karty dźwiękowej, umożliwiają jednak dowolny stopień kompresji, a w przypadku kart bardziej profesjonalnych, nawet pracę bez kompresji. Wykorzystują złącza composite, S-Video i component.

Ad 2. Karty cyfrowe wykorzystują złącza cyfrowe do przechwytywania i zapisu cyfrowo zapisanego obrazu i dźwięku. Stopień kompresji jest tu stały i wynosi 5:1. Do większości kart cyfrowych można podłączyć nie tylko cyfrowy sprzęt wideo (kamerę, magnetowid DV, Digital8), ale również niektóre modele twardych dysków, nagrywarek CD, DVD i innych urządzeń komputerowych. Wyjścia ze złączami analogowymi composite i S-Video pozwalają na analogowy podgląd i zgrywanie zmontowanego materiału.

Ad 3. Karty cyfrowo-analogowe wykorzystują złącza cyfrowe, jak i analogowe do przechwytywania i zapisu obrazu i dźwięku. Karty te wykorzystują rozwiązania zaczerpnięte z dwóch poprzednio opisanych kart, zarówno jeśli chodzi o kompresję (stały i regulowany stopień) jak i złącza (wyjścia cyfrowe i analogowe).

Rys2. Zestaw do montażu nieliniowego. Zbiory własne autora.

Programy

O ile różnice pomierzy poszczególnymi typami kart często są minimalne, o tyle różnice w oprogramowaniu są niekiedy na tyle duże, że można bez wątpliwości stwierdzić, iż to właśnie ono decyduje o klasie danej karty montażowej. Podobnie jak w przypadku kart, oprogramowanie dzielimy na trzy podstawowe kategorie:

1. Oprogramowanie amatorskie
2. Oprogramowanie półprofesjonalne
3. Oprogramowanie profesjonalne

Ad 1. Już nazwa wskazuje, że nie należy się po oprogramowaniu amatorskim spodziewać wielu możliwości związanych z edycją i zapisem końcowym zmontowanego materiału. Najlepszym przykładem tego typu programów jest Pinnacle Studio, program szybki i prosty w obsłudze, pozwalający na oryginalne zmontowanie filmu z np. uroczystości rodzinnych, czy nagranych dla własnego użytku widowisk sportowych.

Ad 2. Oprogramowanie to jest zupełnie zadowalające wielu użytkowników, także tych bardziej zaawansowanych, z uwagi na duże możliwości edycji i końcowego zapisu zmontowanego materiału, korzystną cenę i sprzedaż łącznie z kartami montażowymi. Flagowym przykładem tego oprogramowania jest Adobe Premiere.

Ad 3. To oprogramowanie umożliwia realizację najbardziej zaawansowanej i skomplikowanej edycji stosowanej w systemach profesjonalnych. Użytkownikami tego typu programów są zwykle najbardziej wymagający odbiorcy na czele ze studiami filmowymi i telewizyjnymi.

Najbardziej znanymi programami tego typu są m.in. Avid czy Final Cut. Firma Avid powstała w 1987 roku a dziesięć lat później, w 1997 roku zaprezentowała prototyp systemu do nieliniowej edycji wideo, będący początkiem produktów o nazwie Avid. Najbardziej znanymi produktami firmy są Avid Media Composer oraz Avid Xpres DV. Avid Media Composer z racji swej prostoty i filozofii działania stał się standardem na światowym rynku montażu telewizyjnego i filmowego. Do niedawna Avid Media Composer dostępny był jedynie jako hardware i software tworzący w sumie kompletną stację edycyjną, jednak obecnie

można program ten nabyć jako sam software.

Funkcje

Od montażu nieliniowego nie ma już odwrotu. Największe możliwości i stale obniżające się koszty inwestycji, pozwalające na stworzenie własnego, domowego studia multimedialnego sprawiają, że przejście od edycji liniowej do nieliniowej następuje w bardzo szybkim tempie.
Posiadanie nowoczesnej karty tworzącej system montażu nieliniowego pozwala na realizację wielu rozmaitych funkcji.

Podstawową funkcją jest edycja z szybkim, możliwym w każdym czasie, dostępem do dowolnego miejsca montowanego materiału wideo i jego podział na poszczególne ujęcia, dodanie efektów przejść (miksów), napisów, profesjonalnej grafiki i animacji komputerowej, stosowaniem efektów specjalnych, korekcją obrazu i dźwięku za pomocą specjalnych filtrów, edycją ścieżek dźwiękowych, regulacją tempa i kierunku odtwarzania materiału wideo. Gotowy materiał coraz częściej można już zapisywać nie tylko na kasecie wideo, ale i na własnej płycie CD lub DVD.

Karty montażowe mogą być również wykorzystywane przy tworzeniu pokazów lub prezentacji multimedialnych. Prezentacje te mogą być zapisane na płytach CD lub dzięki podłączeniu karty montażowej do telewizora odtwarzane bezpośrednio poprzez tę kartę.

Posiadacz dobrej klasy drukarki może wykorzystać kartę montażową do pozyskania i wydruku wybranych ujęć z zarejestrowanego materiału wideo.
Funkcją oryginalną, coraz powszechniej wykorzystywaną nie tylko przez profesjonalistów, jest możliwość eksportu materiału wideo na potrzeby Internetu. Materiał taki zapisywany jest w typowych formatach strumieniowych jak WindowsMedia/ASF, QuickTime, RealVideo.

Skomplikowany proces montażu nieliniowego, zamieniający surowy materiał wideo na gotowy film, reportaż, teledysk rozpoczyna się od przeniesienia materiału wideo z kasety na twardy dysk komputera. Czynność tę określa się jako:

1. Przechwytywanie

Przechwytywanie materiału wideo przy wykorzystaniu kart analogowych nazywany jest digitalizacją i polega na przetworzeniu materiału wideo z postaci analogowej na cyfrową. Im wyższy stopień kompresji, tym gorsza jakość obrazu.

W przypadku kart cyfrowych przechwytywanie polega jedynie na cyfrowym kopiowaniu skompresowanego materiału wideo z kasety na dysk. Kamera cyfrowa już w trakcie filmowania kompresuje obraz w stosunku 5:1, nie można więc dowolnie regulować parametrów obrazu i dźwięku. Niezmiennie wysoka jakość pozostaje jednak zachowana w całym toku przechwytywania, obróbki i zapisu materiału, w przeciwieństwie do kart analogowych.

2. Edycja

Edycja jest bez wątpienia najważniejszym procesem każdego montażu. Uporządkowanie chaotycznie zarejestrowanych ujęć, ich uatrakcyjnienie i ożywienie za pomocą precyzyjnego docinania, kopiowania, układania w pożądanej kolejności, tworzenia stopklatek, regulacji tempa i kierunku odtwarzania klipu, czyni z pozornie nieciekawego nagrania zwartą całość.

3. Efekty

Wszelkie efekty specjalne podzielić można na efekty przejść (miksy), filtry obrazu, kluczowanie obrazu, nakładanie obrazu oraz wprowadzanie napisów.

Miksy wykorzystujemy, gdy chcemy w atrakcyjny sposób połączyć ze sobą dwa różne ujęcia. Efekty przejść dzielimy na dwuwymiarowe (przenikanie obrazu, tzw. kurtynki) oraz trójwymiarowe. Na ilość i różnorodność efektów przejść wpływ ma klasa, jakość i typ programu edycyjnego.

Dzięki filtrom obrazu możemy skorygować parametry obrazu poprzez regulację balansu kolorów, jasności, kontrastu, nasycenia, możemy wprowadzić proste efekty, takie jak odwrócenie obrazu w pionie i poziomie, czy efekty specjalne (rozmycia, wyostrzenia, kalejdoskopu, fali, deformacji, efektów świetlnych itp.)

Efekt kluczowania polega w skrócie na usunięciu z obrazu wideo pewnych obszarów mających cechy wspólne (np. jasność, kolor) i uzyskanie możliwości nakładania na siebie poszczególnych warstw z kolejnymi obrazami wideo lub grafiką (np. color key, chroma key, luma key, blue screen, alpha channel).

Przy nakładaniu obrazu mamy do czynienia z efektami powstałymi w wyniku nałożenia na obraz wideo jednego lub więcej obrazów pochodzących z plików wideo lub plików graficznych. Obrazy te często są zdeformowane np. poprzez zmianę wielkości, kształtu nakładanego obrazu.

Wprowadzenie napisów jest efektem polegającym na nałożeniu na obraz wideo przede wszystkim napisów, ale również grafik lub małych animacji. Napisy mogą być statyczne i dynamiczne, przesuwane w pionie (roll) i poziomie (crawl).

4. Dźwięk

Podobnie jak obraz wideo, także i dźwięk poddawany jest różnorodnym zabiegom edycyjnym w procesie montażu. Podstawową operacją jest ustawianie żądanego poziomu dźwięku, czyli regulacja głośności. Należy również wspomnieć o regulacji panoramy dźwięku, umożliwiającej precyzyjne balansowanie i ustawianie dźwięku w kanałach, filtrach dźwięku, pozwalających na korekcję pasma i barwy dźwięku lub na tworzenie efektów specjalnych (pogłosu, echa) oraz o korektorach do dokładnego ustawiania żądanej barwy.

5. Zapisywanie i eksportowanie

Zapis możliwy jest w wielu formatach i standardach. Rodzaj zapisu uzależniony jest głównie od rodzaju karty (analogowa, cyfrowa) oraz sprzętu wideo. Dynamiczny rozwój multimediów i Internetu sprawił, że konieczny jest

eksport materiału wideo do rozmaitych formatów, takich jak Video-CD czy DVD Video. W przypadku Internetu eksport związany jest ze strumieniowym przesyłaniem danych audio-wideo i kompresją obrazu i dźwięku do takiego stopnia, by zmniejszyć strumienie danych do rozmiarów nadających się do swobodnego przesyłania w sieci. Strumieniowość oznacza, iż pobierany materiał jest odtwarzany na bieżąco. Typowe zastosowania eksportowanych w ten sposób materiałów to spoty reklamowe czy zapowiedzi premier filmów kinowych.

Parafrazując przytoczony w początkowych wersach cytat rosyjskiego reżysera, mając jednocześnie na uwadze późniejsze rozważania, można z niemałym przekonaniem stwierdzić, że wszystko jest montażem. Film, teledysk, reportaż jest bowiem sumą kilku wizji artystycznych, także tej proponowanej przez montażystę. Jednak to właśnie w procesie montażu następuje ostateczne zespolenie nagranego wcześniej materiału, nierzadko połączone z usuwaniem niedoróbek, ukrywaniem błędów, maskowaniem niedociągnięć, upiększaniem. Postępujące w galopującym tempie zaawansowanie i progres technologiczny i związane z tym przejście od montażu liniowego do edycji nieliniowej otwiera przed montażystami nowy, jakże bogaty wachlarz możliwości, czyniąc ten rodzaj twórczości dostępnym dla każdego.

Literatura

1. „Montaż nieliniowy-fakty i mity" – Adam Bakalarz, PC World Komputer;
2. „Wszystko jest montażem" – Film & TV Kamera 4/2004
3. „Avid Media Composer i Mojo SDI" - Jarek Barzan i Jarek Somborski, Film & TV Kamera 3/2006.

Polskie Towarzystwo Informatyczne

Oddział Górnośląski

ul. Raciborska 3, 40-074 Katowice
tel./fax: +48 32 2519811
www.pti.katowice.pl
Katowice@pti.org.pl

ISBN 83-60810-06-0

9 788360 810064

ISBN 978-83-60810-06-4