Jedenasta Górska Szkoła PTI

Jakość systemów informatycznych
Problemy i rozwiązania

Szczyrk, 21–25 czerwca 1999
Dom wczasowy „Zagroń"

Polskie Towarzystwo Informatyczne
Oddział Górnośląski

Jedenasta Górska Szkoła PTI

Szczyrk '99 — 21-25 czerwca 1999

„Jakość systemów informatycznych – problemy i rozwiązania"

Szanowni Państwo

Nasze jedenaste świętojańskie spotkanie w Beskidzie nazwaliśmy z pewną dezynwolturą „Jakość systemów informatycznych – problemy i ***rozwiązania****". Dziesięć lat, które upłynęły od naszego pierwszego spotkania w tym miejscu przyniosło zdecydowanie więcej problemów niż rozwiązań. Chyba najważniejsze, czego się nauczyliśmy w tym czasie to fakt, że mamy jeszcze bardzo wiele do nauczenia się przed sobą.*

Tegoroczna Szkoła zgromadziła, jak co roku, przedstawicieli wszystkich poważnych polskich producentów oprogramowania, użytkowników systemów informatycznych, naukowców. Uważamy, że takie spotkania są najcenniejszą (w niczym nie ujmując świetnym Wykładowcom) częścią Szkoły. W czasie kuluarowych dyskusji okazuje się niekiedy, że gdy dwóch robi to samo, to wcale efekty nie muszą być identyczne. Skutkiem takiego spotkania jest zwykle poprawa pracy tych, którzy działają mniej sprawnie. Nie mieliśmy żadnych informacji o obniżeniu poziomu prac w jakiejkolwiek formie, w wyniku udziału w szkole. Proszę zwrócić uwagę na tę sprzeczną z zasadami entropii tendencję.

Tegoroczna Szkoła jest pierwszą konferencją PTI po odbytym niedawno Zjeździe Towarzystwa. Jakkolwiek zjazd nie przyniósł rewolucyjnych zmian w działaniu PTI, to jednak istotnie odnowił i odmłodził skład Zarządu Głównego. W uchwale zjazdowej znalazło się zalecenie zwrócenia w działaniach PTI uwagi na problemy szeroko rozumianej inżynierii oprogramowania.

W tym roku w Kazimierzu nad Wisłą odbędzie się w październiku organizowana przez PTI Pierwsza Krajowa Konferencja Inżynierii Oprogramowania. Jej pomysł powstał nie gdzie indziej, jak w Szczyrku w czasie zeszłorocznej szkoły. W efekcie nastąpił rozdział dwóch nurtów obecnych w Szczyrku od początku. Nurt teoretyczny znalazł swoje ujście w recenzowanej konferencji, pozostawiając Szczyrkowi problemy bardziej praktyczne. Sądzimy, że taki rozdział dobrze będzie służył obu obszarom.

Zapraszając do aktywnego udziału w zajęciach szkoły zachęcamy również do niezamykania oczy na piękno Beskidów na przełomie wiosny i lata.

Piotr W. Fuglewicz

Szanowni Państwo

Nasze jedenaste świętojańskie spotkanie w Beskidzie nazwaliśmy z pewną dezynwolturą „Jakość systemów informatycznych – problemy i ***rozwiązania****". Dziesięć lat, które upłynęły od naszego pierwszego spotkania w tym miejscu przyniosło zdecydowanie więcej problemów niż rozwiązań. Chyba najważniejsze, czego się nauczyliśmy w tym czasie to fakt, że mamy jeszcze bardzo wiele do nauczenia się przed sobą.*

Tegoroczna Szkoła zgromadziła, jak co roku, przedstawicieli wszystkich poważnych polskich producentów oprogramowania, użytkowników systemów informatycznych, naukowców. Uważamy, że takie spotkania są najcenniejszą (w niczym nie ujmując świetnym Wykładowcom) częścią Szkoły. W czasie kuluarowych dyskusji okazuje się niekiedy, że gdy dwóch robi to samo, to wcale efekty nie muszą być identyczne. Skutkiem takiego spotkania jest zwykle poprawa pracy tych, którzy działają mniej sprawnie. Nie mieliśmy żadnych informacji o obniżeniu poziomu prac w jakiejkolwiek formie, w wyniku udziału w szkole. Proszę zwrócić uwagę na tę sprzeczną z zasadami entropii tendencję.

Tegoroczna Szkoła jest pierwszą konferencją PTI po odbytym niedawno Zjeździe Towarzystwa. Jakkolwiek zjazd nie przyniósł rewolucyjnych zmian w działaniu PTI, to jednak istotnie odnowił i odmłodził skład Zarządu Głównego. W uchwale zjazdowej znalazło się zalecenie zwrócenia w działaniach PTI uwagi na problemy szeroko rozumianej inżynierii oprogramowania.

W tym roku w Kazimierzu nad Wisłą odbędzie się w październiku organizowana przez PTI Pierwsza Krajowa Konferencja Inżynierii Oprogramowania. Jej pomysł powstał nie gdzie indziej, jak w Szczyrku w czasie zeszłorocznej szkoły. W efekcie nastąpił rozdział dwóch nurtów obecnych w Szczyrku od początku. Nurt teoretyczny znalazł swoje ujście w recenzowanej konferencji, pozostawiając Szczyrkowi problemy bardziej praktyczne. Sądzimy, że taki rozdział dobrze będzie służył obu obszarom.

Zapraszając do aktywnego udziału w zajęciach szkoły zachęcamy również do niezamykania oczy na piękno Beskidów na przełomie wiosny i lata.

Piotr W. Fuglewicz

Barbara Begier

Politechnika Poznańska
e-mail: **begier@ai-kari.put.poznan.pl**

Ogólne zasady inżynierii a poziom dojrzałości procesów tworzenia oprogramowania

XI Górska Szkoła PTI - Szczyrk 1999

Ogólne zasady inżynierii a poziom dojrzałości procesów tworzenia oprogramowania

Barbara Begier
Politechnika Poznańska
e-mail: begier@ai-kari.put.poznan.pl

XI Górska Szkoła PTI - Szczyrk 1999

Ogólne zasady inżynierii a poziom dojrzałości procesów tworzenia oprogramowania

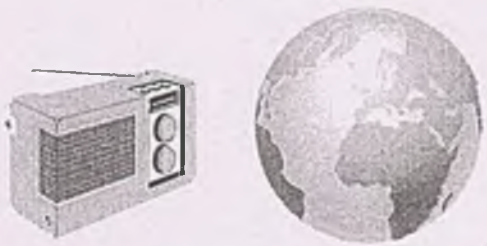
Barbara Begier
Politechnika Poznańska
e-mail: begier@ai-kari.put.poznan.pl

Inżynieria - *projektowanie i konstruowanie obiektów i urządzeń technicznych*

Inżynieria oprogramowania (1968) *ustanawianie i stosowanie solidnych zasad inżynierii w celu uzyskania w sposób ekonomiczny oprogramowania, które jest niezawodne i działa wydajnie na rzeczywistej maszynie*

Inżynieria oprogramowania (*Yingxu Wang, SEES'98*) - dyscyplina, która do tworzenia dużego oprogramowania adoptuje inżynierskie podejście, to jest ustanowione metodologie, procesy, narzędzia, standardy, metody organizacji i zarządzania, systemy zapewnienia jakości i inne, aby w rezultacie uzyskać wysoką produktywność, niskie koszty, sterowalną jakość i wymierny harmonogram opracowywania.

Inżynieria systemów (*Bell Labs, lata 40.*) - utożsamiana z technologią pracy zespołowej



System (*Arthur D. Hall, 1962*) jest to zbiór obiektów wraz z relacjami pomiędzy tymi obiektami oraz pomiędzy ich własnościami.

Otoczenie danego systemu stanowi zbiór wszystkich obiektów nie należących do systemu, których własności oddziałują na system i zarazem ulegają zmianie pod wpływem działania systemu.

Zgodnie z inżynierią systemów tworzenie systemu obejmuje 5 faz

- Badania wstępne - badanie potrzeb i otoczenia, planowanie przedsięwzięcia
- Pierwsza faza projektowania - projekt wstępny, warianty rozwiązań, wybór wariantu
- Druga faza projektowania - planowanie realizacji systemu
- Pierwsza faza realizacji - prototypy i ich badania
- Druga faza realizacji

Zadania inżynierii systemów

- opracowanie zasad
- informowanie
- formułowanie planów
- zapewnienie metod
- harmonizowanie działań
- optymalizacja zasobów
- przeprowadzanie każdej czynności z użyciem określonej techniki

Pożądane cechy osobowe inżyniera

- Systemowy punkt widzenia (całość koncepcji)
- Rozsądek i obiektywizm, podporządkowanie wymaganiom
- Wyobraźnia
- Zręczność i dyplomacja w kontaktach z ludźmi
- Dar przekonywania ludzi
- Umiejętność korzystania z doświadczeń

Projektowanie tradycyjne stosowane do XIX wieku

- Integracja procesu projektowania z procesem realizacji w jednym zespole (firmie)
- Stosowanie metody prób i błędów wobec braku metod badania poprawności projektu
- Zdobywanie praktycznych kwalifikacji podczas pracy

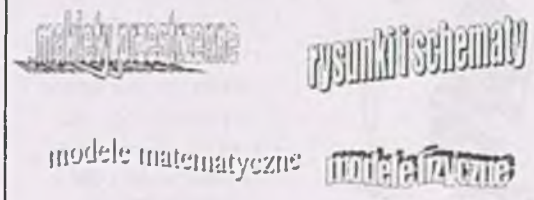
Zasady rzeczowe projektowania wg teorii projektowania (*Cz. Szymczak*)

- fizyczna wykonalność obiektu
- strukturalizacja procesu projektowania
- dekompozycja problemu, a w następstwie projektu
- zwiększanie pewności wykonania projektu, zaniechanie projektów nierealnych lub nieefektywnych
- dominująca rola koncepcji całości systemu
- komunikatywność: pełny opis projektu, zgodność stosowanych definicji i oznaczeń

Zasady normatywne projektowania wg teorii projektowania (*Cz. Szymczak*)

- Zasada zaspokojenia potrzeby (konieczność jej opisu)
- Zasada wartości użytkowej - wartość użytkowa obiektu powinna być co najmniej równa kosztom wytworzenia
- Zasada wykonalności finansowej
- Zasada minimalnego zaangażowania finansowego
- Zasada optymalności - dobór kryteriów oceny i ich miar
- Zasada jedności funkcji, formy i konstrukcji - konieczność harmonii, ścisła współpraca autorów, kompromisy
- Zasada równomiernego zużycia
- Zasada bezpieczeństwa i niezawodności

Modelowanie ma na celu badanie zachowania systemu przy działaniu rozmaitych czynników



Symulacja - eksperymentowanie na modelu, naśladowanie zachowania się systemu w rozmaitych warunkach i okolicznościach

Awaria - przekroczenie ustalonej (zadanej *a priori*) granicy przez zmienne określające zachowanie się systemu

Ocenie podlegają nie tylko własności techniczne i użytkowe projektowanego systemu, ale również właściwości ekonomiczne oraz psychologiczno-estetyczne

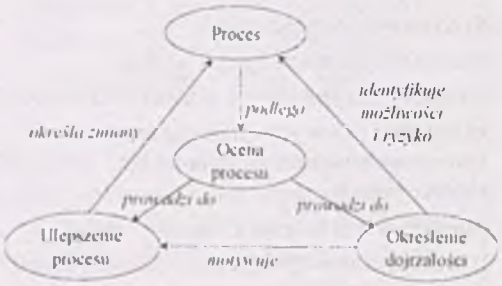
ISO/IEC TR 15504 (15.08.1998)

1: Concepts and introductory guide
2: A reference model for processes and process capability
3: Performing and assessment
4: Guide to performing assessments
5: An assessment model and indicator guidance
6: Guide to competency of assessors
7: Guide for use in process improvement
8: Guide for use in determining supplier process capability
9: Vocabulary

Standard ISO/IEC 15504 pozwala:

- na użytek własny firmy - ocenić stan procesów w celu określenia sposobu ich ulepszenia
- określić odpowiedniość wykonywanych w organizacji procesów do klasy wymagań
- określić odpowiedniość procesów dostawcy do potrzeb konkretnego kontraktu lub klasy kontraktów
- redukować niepewność nabywcy w wyborze dostawcy
- zmniejszyć ryzyko przez umożliwienie kontroli
- zapewnić podstawę określenia równowagi między potrzebami, wymaganiami i kosztami a możliwościami wytwórczymi konkurujących dostawców

Kontekst oceny procesów zgodnie z ISO/IEC 15504

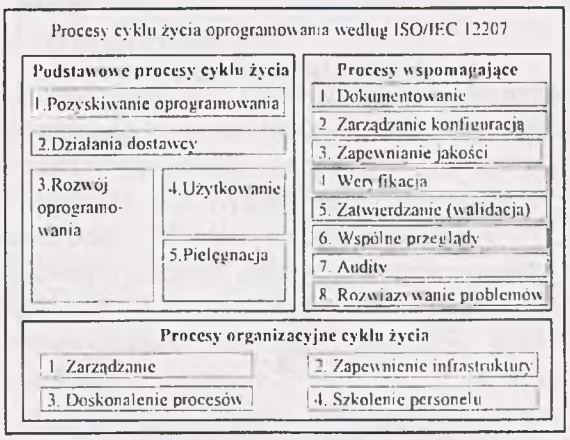


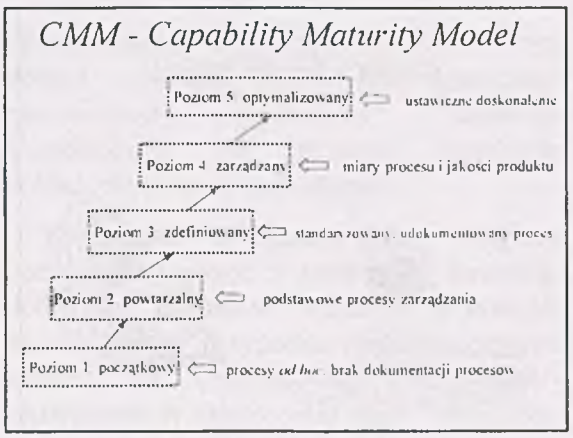
Model oceny procesów wg ISO/IEC 15504

- Określono kategorie procesów realizowanych przez dostawcę, przeniesione niemal dokładnie z ISO 12207: dla kategorii zidentyfikowano procesy składowe
- Dla całych kategorii oraz procesów składowych zdefiniowano oczekiwane rezultaty, które stanowią podstawę oceny danego procesu
- Przyjęto 6-stopniową skalę oceny dojrzałości procesów - 5 stopni obecnych w modelu CMM oraz poziom zerowy dla nieobecnego lub niekompletnego procesu
- Na danym poziomie dojrzałości proces opisuje zestaw atrybutów, które stanowią kryteria przeprowadzanej oceny każdego procesu dostawcy

Oceniane kategorie procesów

- CUS: działania na styku 'klient-dostawca' od formułowania kontraktu po przekazanie oprogramowania z zapewnieniem jego prawidłowego użytkowania i serwisu
- ENG: specyfikowanie, implementacja, dokumentowanie, pielęgnacja
- SUP: procesy wspomagające
- MAN: zarządzanie przedsięwzięciem
- ORG: określenie celów organizacji dostawcy i metod ich osiągania; zarządzanie zasobami





Rezultaty prawidłowej realizacji np. procesu CUS.4 (*requirements elicitation process*)

- Ustanowienie ciągłej komunikacji z klientem
- Zdefiniowane wymagania i zgoda klienta
- Ustanowiony mechanizm włączania nowych wymagań
- Ustanowiony mechanizm ciągłego monitorowania potrzeb klienta
- Ustanowiony mechanizm zapewniający klientowi możliwość określenia statusu wymagań
- Zmiana technologii oraz potrzeb klienta będzie identyfikowana, a ich wpływ na procesy zarządzany

Atrybut procesu na poziomie 1:
- PA 1.1 atrybut wykonania procesu

Atrybuty procesu na poziomie 2:
- PA 1.1 atrybut wykonania procesu
- PA 2.1 atrybut zarządzania wykonaniem
- PA 2.2 atrybut zarządzania produktem procesu

Atrybuty procesu na poziomie 3:
- PA 1.1 atrybut wykonania procesu
- PA 2.1 atrybut zarządzania wykonaniem
- PA 2.2 atrybut zarządzania produktem procesu
- PA 3.1 atrybut definicji procesu
- PA 3.2 atrybut zasobów procesu

itd.

N (*not achieved*), 0-15%,
P (*partially*), 16-50%,
L (*largely*), 51-85%,
F (*fully*), 86-100%

Forma zapisu wyniku oceny dojrzałości procesów

Atrybuty procesów charakterystyczne dla poziomów

Procesy w ramach kategorii	1.1	2.1	2.2	3.1	3.2	4.1	4.2	5.1	5.2
CUS.1	F								
CUS.2	L								
CUS.3	L								
CUS.4	L								
ENG.1	L								
ENG.2	F								
ENG.3	P								

Przeprowadzenie oceny procesów

- Założenie powtarzalności procesu oceny
- Zdefiniowanie zakresu oceny oraz danych wejściowych przed zbieraniem danych do oceny
- Dobór zespołu oceniającego i jego lidera
- Zaplanowanie oceny
- Zidentyfikowanie technik analizy i oceny danych oraz ocenianych procesów przed oceną
- Zatwierdzenie danych podlegających ocenie
- Podjęte drogą głosowania decyzje są raportowane.

Walka z biurokracją?

Inspektor nadzoru

Pożądane uregulowania prawne

Monitorowanie przedsięwzięć publicznych w zakresie informatyzacji przez niezależnych obserwatorów

Henryk Krawczyk

Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska

Narzędzia i eksperymenty pomiarowe oprogramowania

Streszczenie: Przedstawiono problemy metrologii oprogramowania i jej wykorzystanie w procesie oceny jakości aplikacji informatycznych. Zaprezentowano metodologię pomiarów, narzędzia wspomagające wykonywanie eksperymentów pomiarowych oraz sposoby opracowania i prezentowania wyników pomiarowych.

1. Wprowadzenie

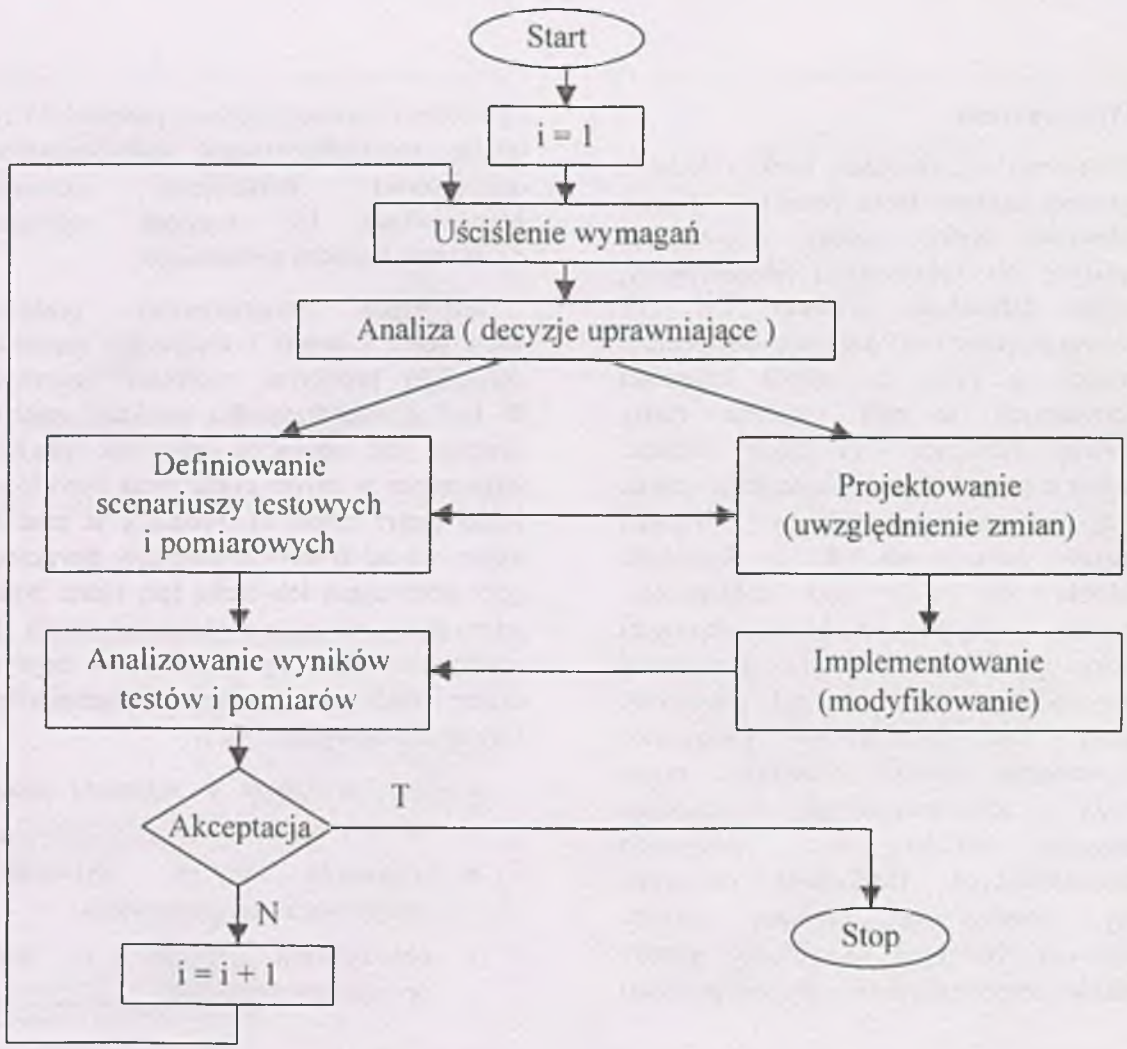
Metrologia jest dziedziną nauki i techniki obejmującą zarówno teorię pomiarów (pojęcia podstawowe, modele, metody, wzorce) jak i praktykę ich wykonywania (eksperymenty, narzędzia, laboratoria, jednostki miar) [7]. Głównym pojęciem metrologii jest więc pomiar. Rozumie się przez to zespół czynności wykonywanych w celu ustalenia miary określonej wielkości. Na ogół wielkość mierzoną α porównujemy z elementami wzorca $\alpha_1, \alpha_2, \ldots, \alpha_n$ i stwierdzamy w wyniku pomiarów, że $\alpha \in \langle \alpha_i, \alpha_{i+1} \rangle$. Szerokość przedziału $\langle \alpha_i, \alpha_{i+1} \rangle$ jest dokładnością pomiarów i zależy od klasy przyrządu pomiarowego. W celu zapewnienia wysokiej wiarygodności pomiarów przyrządy pomiarowe powinny być legalizowane i okresowo uwierzytelniane przez odpowiedni organ rządowy w celu stwierdzenia, że spełniają wymagania ustalone w przepisach międzynarodowych. Dokładność pomiarów zależy również od przyjętej metody pomiarowej. Na ogół wyróżniamy metody pomiarów bezpośrednich (np. długość pudełka) i pośrednich (pomiar objętości pudełka). W tym drugim przypadku istotne jest zapewnienie odpowiedniej dokładności pomiarów bezpośrednich by otrzymać wymaganą dokładność pomiaru pośredniego.

Metrologia oprogramowania przejmuje wiele pojęć i metod z tradycyjnej metrologii dotyczącej pomiarów wielkości fizycznych. W tym jednak przypadku obiektem zainteresowania jest aplikacja oraz inne produkty wytworzone w całym cyklu życia oprogramowania (patrz tabela 1). Produkty te znacznie różnią się od typowych obiektów fizycznych, gdyż dominującą ich cechą jest różna postać informacji a nie pewne własności materii. Co więcej metrologia oprogramowania odgrywa istotną rolę w inżynierii oprogramowania z uwagi na następujące fakty:

- jest narzędziem w inżynierii jakości oprogramowania,
- wspomaga proces wytwarzania i utrzymania oprogramowania,
- obiektywizuje ocenę i audyt oprogramowania.

Tabela 1. Podstawowe produkty wytwarzane w cyklu życia oprogramowania

Faza życia Oprogramowania	Podstawowe produkty aplikacji informatycznej
Specyfikacja	Dokumentacja specyfikacji
Analiza	Dokumentacja analizy
Projekt	Architektura (komponenty)
Implementacja	Kod źródłowy, kod binarny
Testowanie	Konfiguracja systemu, dokumentacja testowa
Eksploatacja	Raporty o zachowaniu się aplikacji
Wycofanie	Protokół kasacji



Rys 1. Rola pomiarów w cyklu wytwarzania oprogramowania

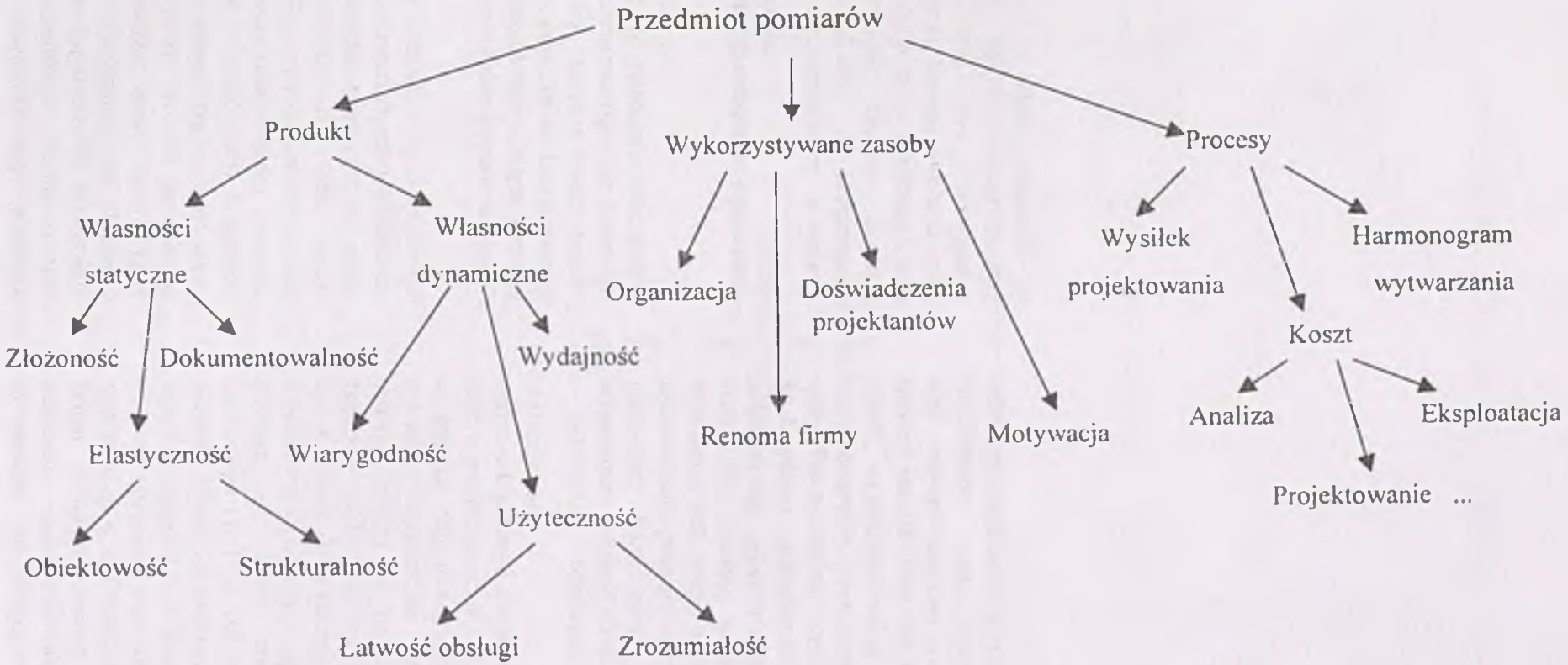
Rysunek 1 przedstawia rolę pomiarów w procesie wytwarzania oprogramowania. Złożone aplikacje informatyczne realizowane czy to w modelu kaskadowym czy spiralnym wymagają wykonania wielu iteracji [3,4], w których uściśla się zarówno stawiane im wymagania, jak również podejmuje się różnego typu decyzje o modyfikacji produktów w celu zapewnienia odpowiednich charakterystyk jakościowych końcowemu produktowi. Metrologia oprogramowania jest istotna zarówno przy podejmowaniu decyzji lokalnych dotyczących poszczególnych komponentów aplikacji, jak i globalnych dotyczącej całej aplikacji.

W pracy przedstawiono wybrane problemy metrologii oprogramowania z punktu widzenia inżynierii jakości. Dotyczą one przygotowania eksperymentów pomiarowych (wielu pomiarów zadanych charakterystyk), jak i narzędzi wspomagających ich wykonanie oraz analizę otrzymanych wyników.

2. Metrologia oprogramowania jako narzędzie oceny jakości

Z rozważań przedstawionych w poprzednim rozdziale wynika, że przedmiotem pomiarów mogą być zarówno produkty wytwarzane w cyklu życia oprogramowania jako towarzyszące im procesy oraz oczywiście wykorzystywane zasoby. Rys. 2 przedstawia zakres metrologii oprogramowania uwzględniający wszystkie te aspekty odnoszące się w zasadzie do produktu końcowego, tj. programu wytworzonej aplikacji. W przypadku produktu oceniać możemy jego charakterystyki statyczne, tzn. odnoszące się do kodu źródłowego charakterystyki dynamiczne uwzględniające zachowanie się programu podczas jego wykonywania się w zadanym środowisku [1,2]. W tym drugim przypadku pojawia się nowa trudność dotycząca powtarzalności pomiarów. Wymaga to zapewnienia jednakowych warunków wykonania się aplikacji. Dotyczy to nie tylko podania tych samych danych na wejście programu, ale również odtworzenie stanu środowiska w którym aplikacja będzie wykonana. To drugie zdanie może być trudne do zrealizowania w przypadku systemów sieciowych czy rozproszonych.

Wykonanie pomiarów dla procesów wytwarzania aplikacji jest jeszcze bardziej trudniejsze, gdyż należy uwzględnić takie niewymierne charakterystyki jak wysiłek projektowania, jego koszt, harmonogram wytwarzania czy częstość zmian dokonywanych w produktach. Wymaga to interdyscyplinarnego podejścia integrującego punkty widzenia dotyczące zarządzania, ekonomii, psychologii czy metodologii wytwarzania. W przypadku oceny wykorzystywanych zasobów, które dotyczą zarówno ludzi jak i używanych przez nich narzędzi, techniki pomiarowe powinny uwzględniać różne aspekty: techniczne, organizacyjne a nawet psychologiczne.



Rys 2. Zakres metrologii oprogramowania

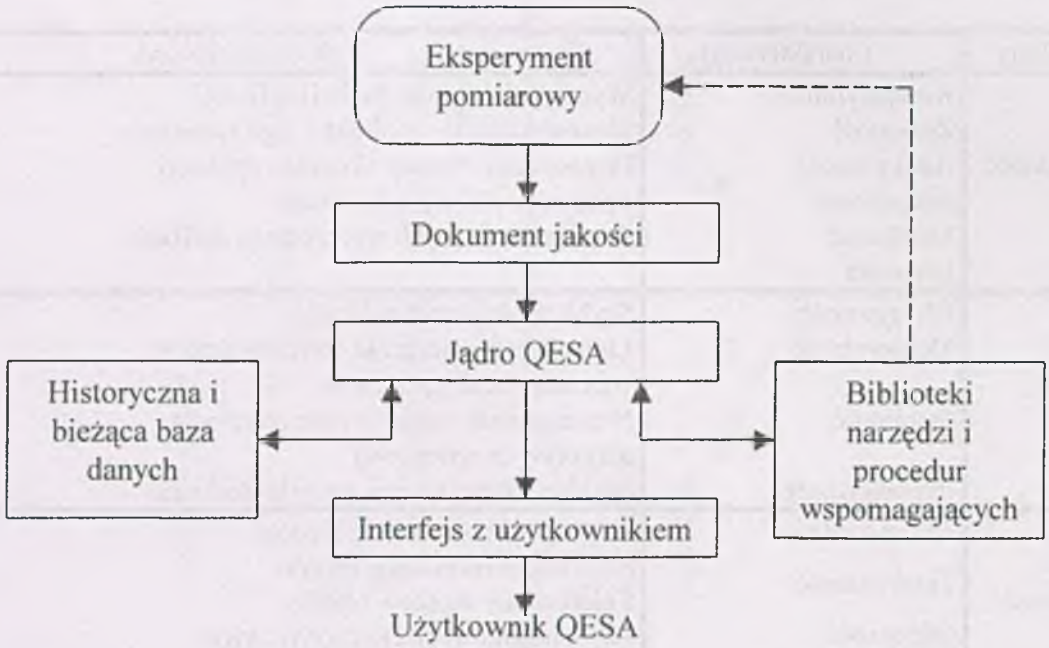
Tabela 2 Opis charakterystyk jakości oprogramowania

Atrybuty	Charakterystyki	Opis charakterystyk
Funkcjonalność	Kompatybilność	Wyczerpujący zbiór funkcji aplikacji
	Złożoność	Skomplikowanie produktu i jego elementów
	Adekwatność	Dopasowanie funkcji do zadań aplikacji
	Integralność	Spójność aplikacji jako całości
	Możliwość śledzenia	Powiązanie różnych reprezentacji aplikacji
Wydajność	Efektywność	Szybkość działania aplikacji
	Skalowalność	Liniowe przyspieszenie obliczeń wraz ze wzrostem liczby przerwań
	Stabilność	Niezmienność osiągów wraz ze zmianą platformy programowej
	Interakcyjność	Szybkość komunikacji z użytkownikiem
Wiarygodność	Niezawodność	Częstość pojawiania się błędów
	Testowalność	Zdolność identyfikacji błędów
	Odporność	Tolerowanie skutków błędów
	Bezpieczeństwo	Zapobieganie sytuacjom krytycznym
	Zabezpieczenie	Kontrola dostępu do aplikacji
Satysfakcja	Łatwość użycia	Prostota obsługi
	Zrozumiałość	Zrozumienie aplikacji na podstawie jej opisu
	Łatwość nauki	Wysiłek dla umiejętnego obsługiwania aplikacji
	Produktywność	Stopień wspomagania aplikacji przez system
	Akceptacja	Akceptacja niefunkcjonalnych wymagań użytkownika
Elastyczność	Przenośność	Przystosowanie do nowych platform
	Modyfikowalność	Łatwość dokonania zmian wymaganych przez środowisko
	Konfigurowalność	Możliwość dopasowania do nowych potrzeb użytkownika
	Łatwość testowania	Prostota projektowania i wykorzystania testów

Na dzisiejszym etapie rozwoju metrologii oprogramowania wykonanie wielu z wyżej wymienionych eksperymentów pomiarowych nie jest możliwe. Najważniejszym ogniwem kompleksowego procesu oceny jest nadal człowiek z jego dużym doświadczeniem i niezrozumiałą jak dotąd intuicją. Dla dobrze zdefiniowanych modeli jakości [5] można opracować precyzyjne eksperymenty pomiarowe. Wiele z nich może być wspomagane odpowiednimi narzędziami pomiarowymi dotyczącymi wybranej charakterystyki czy metryki. W tabeli 2 przedstawiono główne atrybuty jakości oraz odpowiadające im charakterystyki jakościowe. Wiele z nich może być bezpośrednio zmierzone bądź estymowane na podstawie pomiarów historycznych albo szacowane przez ekspertów. Pojawia się problem, jak zagospodarować te wszystkie wyniki pomiarowe i w jaki sposób dokonywać całościowej lub wybiórczej analizy jakościowej oprogramowania.

3. Narzędzie QESA

Różne mogą być metody oceny jakości wspomagane odpowiednim narzędziem pomiarowym [8]. Poniżej ograniczę się tylko do prezentacji oryginalnej metodologii, która integruje różne podejścia. Metoda ta bazuje na opracowanym narzędziu QESA [6], którego ogólna architekturę przedstawia rysunek 3.



Rys 3. Architektura systemu pomiarowego

W proponowanej metodologii oceny jakości wyróżnia się następujące kroki postępowania:

1. Definicja modelu jakości (opis trójpoziomowego drzewa: atrybuty – charakterystyki – metryki; patrz tabela 2).
2. Zaplanowanie odpowiednich do modelu jakości eksperymentów pomiarowych bezpośrednich i pośrednich umożliwiających utworzenie zależności badanej charakterystyki lub metryki od podanych parametrów aplikacji i/lub określonych parametrów środowiska w którym ta aplikacja będzie wykonywana.
3. Wykonanie pomiarów i utworzenie dokumentu jakości, który przechowuje otrzymane dane pomiarowe, jak również inne zadane parametry.
4. Analiza dokumentu jakości z ewentualnością uzupełnienia pomiarów (powrót do punktu 2) oraz podjęcie decyzji o zapamiętaniu dokumentu jakości w historycznej bazie danych.
5. Analiza uogólnionych zależności na podstawie historycznych dokumentów jakości dotyczącej danej klasy aplikacji bądź wybranej fazy życia konkretnej aplikacji, z możliwością odtworzenia i uzupełnienia niektórych historycznych eksperymentów pomiarowych.

Prezentowana metodologia oceny jakości posiada następujące cechy:

— uniwersalność określona przez możliwość analizy różnych aplikacji, dzięki standardowemu interfejsowi pomiędzy konkretnym narzędziem pomiarowym a jądrem pakietu QESA,

— elastyczność wyrażona poprzez dobór odpowiedniego modelu jakości w zależności od potrzeb użytkownika oraz sugestii eksperta co do ważności poszczególnych atrybutów czy charakterystyk oraz relacji między nimi,

— rozbudowywalność dzięki wydzieleniu biblioteki narzędzi pomiarowych i procedur wspomagających analizę jakości oraz możliwości dołączania nowych przydatnych do realizacji innego typu eksperymentów pomiarowych bądź umożliwiających dodatkowe wnioskowanie ma podstawie dostępnych dokumentów jakości,

— bieżąca prezentacja wyników oceny pozwalająca na łatwe zorientowanie się, które z eksperymentów zostały zakończone oraz które wymagają dalszego wykonania bądź powinny być powtórzone.

Tak wiec pakiet QESA nie jest narzędziem oceniającym wszystkie atrybuty charakterystyki, czy metryki jakości oprogramowania, raczej wspomaga organizację takiego procesu oraz zapewnia przechowywanie reprezentatywnych wyników pomiarów. Może więc być łatwo włączony do procesu iteracyjnego wytwarzania oprogramowania przedstawionego na rys. 1.

4. Wnioskowanie o jakości aplikacji

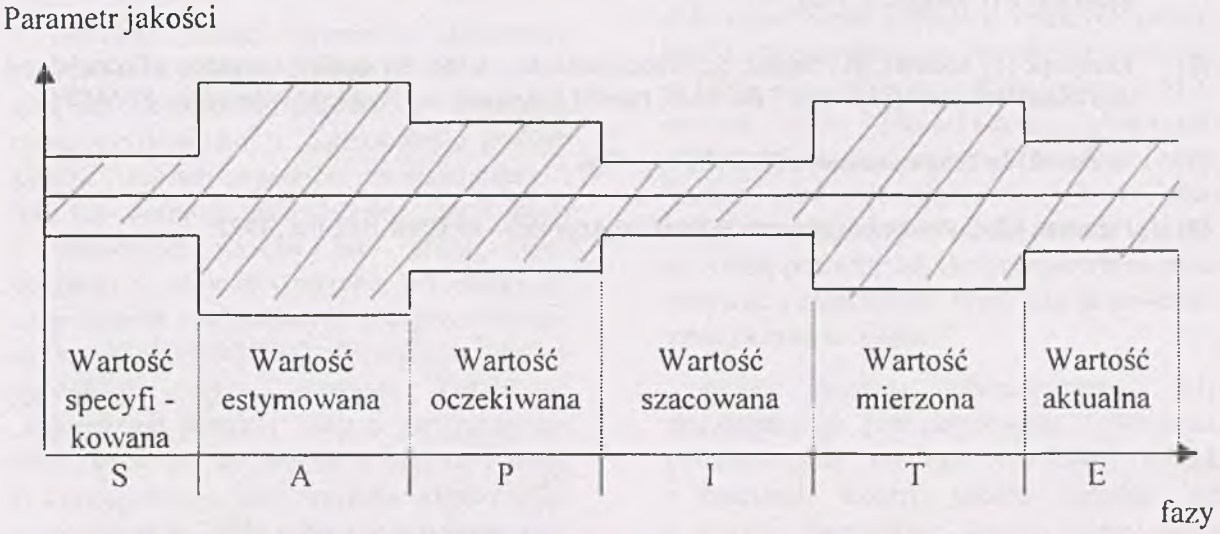
Ocenę jakości stanowić może jedna konkretna wartość ogólnego wskaźnika jakości, zbiór takich wartości, bądź zestaw reprezentatywnych charakterystyk zdjętych podczas wykonania różnych eksperymentów pomiarowych. Każda z tych form prezentacji ma swoje wady i zalety, dlatego w praktyce istotną rolę odgrywa właściwa forma wizualizacji.

Pakiet QESA posiada trzy różne formy wizualizacji wyników pomiaru i oceny jakości:

— w formie odpowiednich tabel zawierających mierzone lub ocenione wartości,

— w formie uogólnionego diagramu kołowego, gdzie poszczególne wyniki reprezentuje wartość poszczególnych parametrów (atrybutów, charakterystyk lub miar) z możliwością przedstawienia wzajemnych powiązań,

— w formie wykresów podających estymowane i rzeczywiste wartości danego parametru jakości dla poszczególnych faz wytwarzania aplikacji (patrz rys. 4).



Rys 4. Wizualizacja dowolnego parametru jakości w cyklu życia oprogramowania

Ostatnia forma prezentacji możliwa jest dzięki zapamiętywaniu dokumentów jakości dotyczących poszczególnych baz wytwarzania danej aplikacji. Jesteśmy w stanie określić nie tylko poszczególne wartości parametrów, ale dokładność ich estymacji lub pomiarów w poszczególnych fazach życia oprogramowania. Taki wykres dostarcza nie tylko informacji o procesie wytwarzania, ale może stanowić podstawę do doskonalenia samych procedur pomiarowych.

Prezentowana metodologia oceny posiada jeszcze inną zaletę. Dzięki rozpoznaniu różnych tendencji i zależności istniejących w procesie wytwarzania może proponować wartości wybranych parametrów jakości w danej fazie cyklu życia na podstawie znajomości innych lub tych samych parametrów w poprzednich fazach cyklu życia. Doskonalenie metod prognozowania może prowadzić do minimalizacji liczby wykonywania czasochłonnych i kosztownych procedur pomiarowych.

5. Uwagi końcowe

W pracy zaprezentowano nową metodologię oceny jakości aplikacji wykorzystującą różne techniki pomiarowe oprogramowania. Metodologia ta została zaimplementowana i jest dostępna w postaci pakietu QESA, który może być konfigurowalny i rozbudowywalny w zależności od potrzeb konkretnego użytkownika. W aktualnej wersji bierze się pod uwagę jedynie ocenę produktów wytworzonych w różnych fazach życia oprogramowania.

Bibliografia

[1] Drake T.: Measuring Software Quality: A Case Study. IEEE Computernov., 1996.

[2] Fenton N.E. et al.: Software Metrics. A Rigorous & Practical Approach. Int. Thomson Computer Press, 1995.

[3] Jóźwiak L., Ong S.A.: Quality –Driven Decision Making Methodology gfor System-Level Design. Proc. of EUROMICRO – 22, 1996.

[4] Krawczyk H., Wiszniewski B.: Analysis and testing of Distributed Software Applications. Research Studies Press Ltd., Baldock, Hertfordshire, England, 1998.

[5] Krawczyk H.: Modele i ocena jakości systemów informatycznych. Materiały 14-ego Jesiennego Spotkania PTI, Mrągowo, 1998.

[6] Krawczyk H., Sikorski, M., Szejko, S., Wiszniewski, B. "A tool for quality evaluation of parallel and distributed software applications". Int. Conf. Parallel Processing and Applied Mathematics, PPAM'99.

[7] Piotrowski J.: Teoria pomiarów. PWN, W-wa, 1986.

[8] Pressman RS: Software Engineering: A Practical Approach. McGraw-Hill, Inc., 1992

Marek Miłosz

Politechnika Lubelska
Katedra Informatyki
e-mail: marekm@pluton.pol.lublin.pl

Jakość systemów informatycznych: między teorią a praktyką

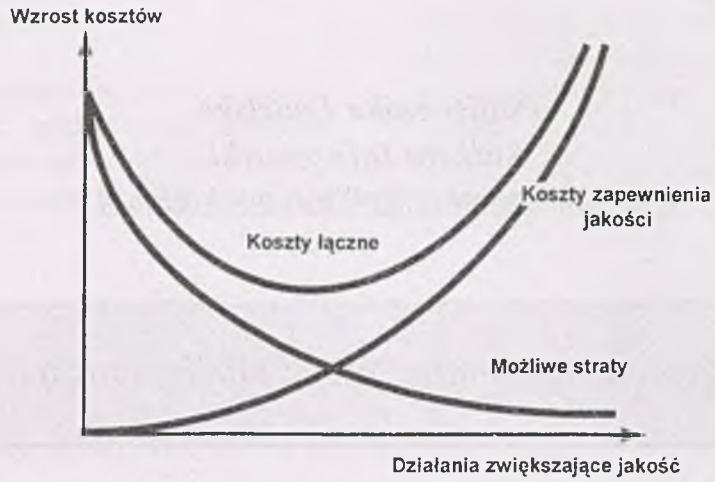
Referat przedstawia problemy związane z jakością systemów informatycznych i dwa podejścia do procesu zapewnienia jakości: weryfikacja produktu i procesu jego wytworzenia. Szczegółowo został omówiony model EFQM i jego zastosowanie do oceny i porównania jakości procesów w firmach informatycznych.

1. Jakość i jej wymiar ekonomiczny

Definicji jakości systemów informatycznych (i nie tylko) jest wiele. Zwykle posługują się one pojęciem „zgodności z wymaganiami użytkownika" i „zaspokojenie potrzeb klienta", czasem „zgodność ze standardami", lub też definiują jakość jako „ogół cech i właściwości wyrobu lub usługi, które decydują o zdolności wyrobu lub usługi do zaspokojenia stwierdzonych lub przewidywanych potrzeb" (ISO 8402). Definicje te bardzo ogólnikowo traktują problem. Zaspokajać „stwierdzone potrzeby" można oprogramowaniem, które raz na godzinę zawiesza system. Z drugiej strony, jeśli potrzeby użytkownika są związane ze 100% gotowością systemu (np. oprogramowanie bankomatu) to żaden system informatyczny ich nie spełni.

Problemem jest zatem definiowalność (stwierdzenie a, zgodnie z definicją, nawet przewidywanie) potrzeb użytkownika. Kto ma je definiować? Użytkownik. Zwykle ma on dość ograniczone pojęcie o własnych potrzebach i ich zaspokajaniu. Ponadto użytkownicy są różni – co jednemu jest potrzebne w SI to innemu może przeszkadzać. Wymagania użytkowników zmieniają się także w czasie. Trudno więc jest zdefiniować je w jeden spójny i niezmienny w czasie sposób. Ale tak czy owak potrzeby użytkowników trzeba przewidywać i zaspokajać. Tylko kto to powinien robić i kto za to zapłaci?

Jakość systemu informatycznego (SI), niezależnie jak jest pojmowana i mierzona, związana jest bowiem w ścisły sposób z kosztami. Koszty jakości składają się z dwóch elementów: kosztu zapewnienia jakości i kosztu strat, związanych z jej brakiem [10]. Oba koszty mają przeciwstawne tendencje, tzn. pierwszy rośnie wraz z wysiłkami projakościowymi a drugi maleje (rys. 1).



Rys. 1. Tendencje zmiany kosztów systemów informatycznych, związanych z działaniami projakościowymi

Dodatkowy koszt (straty) związany z jakością przy braku jakichkolwiek działań zapewniających tą jakość jest duży, ale możliwy do policzenia. Składają się na niego wszelkiego typu dodatkowe koszty utrzymania systemu, straty biznesowe (np. obniżenie sprzedaży) czy też marketingowe (np. utrata wiarygodności). Są one ponoszone przez klienta – użytkownika SI.

Minimalizacja do zera możliwych strat użytkowników SI jest związana ze wzrostem (praktycznie do nieskończoności) kosztów zapewnienia jakości. Na koszty te składają się koszty:

- opracowania systemu spełniającego wszystkie zdefiniowane (a także przewidywalne) wymagania klienta;
- identyfikacji, które produkty nie spełniają ww. kryteriów;
- poprawy (i wymiany) zidentyfikowanych produktów.

Koszty te są oczywiście ponoszone przez producenta – dostawcę SI.

„Cena jakości" (tj. koszty łączne) posiada zatem pewien punkt optymalny, leżący pomiędzy obydwoma skrajnymi przypadkami. Można nazwać go punktem ***optymalnej jakości*** systemu informatycznego, satysfakcjonującej obie strony: dostawcę i klienta. Problem polega na jego znalezieniu, a także na osiągnięciu swoistego konsensusu pomiędzy dostawcą i klientem. Polega on na tym, że każda ze stron musi ponosić koszty, związane z jakością systemu informatycznego. U dostawcy są to koszty zapewnienia jakości, a u klienta – zwiększone koszty eksploatacji czy też utracone korzyści.

Zwykle to klient dyktuje warunki, ale też i za wszystko płaci. Decydując się zatem na „wysoką jakość" wie, że będzie musiał za nią odpowiednio dużo zapłacić. Zrobi to tylko w sytuacji, kiedy dodatkowe koszty zwrócą się brakiem strat. To dlatego jedni wykorzystują systemy informatyczne w cenie 100. czy 200. złotych (ale jeżące się od błędów lub ograniczone funkcjonalnie) a inni płacą znacznie więcej za podobne produkty innych producentów i o innej jakości. Przeniesienie na klienta kosztów jakości systemu informatycznego znajduje zwykle odwzorowanie w niskiej cenie produktu. I to jest normalne (zresztą nie tylko w informatyce). Patologią są natomiast sytuacje, kiedy z tego przeniesienia dostawca czerpie nadmierne korzyści i nadmiernie obciąża klienta. Jest to szczególnie groźne zjawisko w przypadku oprogramowania powielarnego, którego koszty kopiowania i dystrybucji są niewielkie a klientów dużo. Przy niskich nawet kosztach jednostkowych łączne straty klientów są wówczas ogromne. Duże są za to oszczędności dostawcy SI.

2. Problemy jakości systemów informatycznych

Systemy informatyczne są specyficznymi produktami w wielu aspektach, w tym i jakościowym. Posiadają cały szereg cech wyróżniających je od innych produktów lub usług. Są to:

- ***abstrakcyjność i złożoność SI*** – cecha ta powoduje, że trudno jest zdefiniować parametry charakteryzujące atrybuty jakości systemu i jeszcze trudniej je zmierzyć; atrybuty te powinny odwzorowywać wiele różnych aspektów SI;
- ***niewidzialność systemów informatycznych*** – cecha ta wynika z abstrakcyjności SI ale też jest związana z szerszym problemem, w tym i nieujawnianiem przed dostawców szczegółów architektury systemów czy też kodu źródłowego programów; dostawcy postępują tak z różnych powodów – przeważnie jest to element ochrony praw autorskich lub też wynika on ze złożoności danego elementu SI; dość często klient nie ma żadnej możliwości skontrolowania zakupionego produktu, nawet w sytuacji gdy otrzyma pełną jego dokumentację;
- ***jednostkowość systemów informatycznych*** – większość SI jest tworzona (nie należy mylić ze sprzedażą, która jest rezultatem prostego i łatwego powielania raz wytworzonego SI) jako „produkcja" jednostkowa, w sensie niepowtarzalnych warunków (w tym i wymagań) jej realizacji; jednostkowość produkcji wymusza inne podejście do problemów jakości niż w przypadku produkcji masowej (np. praktycznie bezużyteczne są statystyczne miary jakości produktów) przy zachowaniu pewnych elementów masowej produkcji (np. wielu klientów kupujących dany produkt – możliwość rozłożenia kosztów – ale też i problemów – na wielu klientów);
- ***wielowarstwowość SI*** – cecha ta prowadzi do „rozmycia odpowiedzialności" za produkt; SI jest bowiem systemem składającym się z różnych elementów, pochodzących od różnych dostawców; użytkownik nie wie kogo obarczyć odpowiedzialnością za błędne działanie SI; w sytuacji wielu różnych dostawców elementów składowych SI stwarza to możliwości różnych nadużyć dostawców w stosunku do klienta: trudno jest bowiem zdefiniować obszar odpowiedzialności poszczególnego dostawcy; pewnym wyjściem z tej sytuacji jest korzystanie ze specjalnego typu dostawców SI – integratorów, którzy przyjmują odpowiedzialność za połączenie w jedną całość wielu warstw SI;
- ***brak jednoznacznych miar jakości SI*** – nie jest możliwe stworzenie jednolitego systemu pomiaru jakości SI opartego o wielkości kwantyfikowalne i jednoznacznie mierzalne; problemem jest różnorodność (i słaba definiowalność) wymagań użytkowników (również tych samych ale w różnym czasie), różnorodność cech jakości i ich wag (subiektywizm); większość systemów bazuje na ocenach ekspertów i/lub ankietach, a zatem na subiektywnych ocenach ludzi;
- ***brak możliwości praktycznej weryfikacji jakości SI a priori*** – cecha ta jest związana z brakiem miar, a w konsekwencji z brakiem możliwości jednoznacznego wyspecyfikowania „stwierdzonych lub przewidywanych potrzeb"; nie można także zweryfikować jakości SI z przyczyn „obiektywnych" – brak jest fizycznych możliwości zupełnego (tj. wyczerpującego wszystkie możliwości) przetestowania nietrywialnego SI [6]; problem leży po obu stronach, ale przede wszystkim wynika z braku orientacji (wyartykułowania potrzeb) klienta i jego różnorodności (w sensie wykorzystania tego samego jednostkowego produktu przez wielu klientów o różnych wymaganiach i różnych warunkach).

Specyfika SI w aspekcie jakości doprowadziła do wytworzenia szczególnego podejścia do SI, w którym klient płaci nie wiadomo za co (np. „za użytkowanie produktu") a dostawca ogranicza zakres gwarancji do... wartości nośników (tj. do ok. 1 USD lub mniej za CD ROM przy wartości produktu rzędu tysięcy USD).

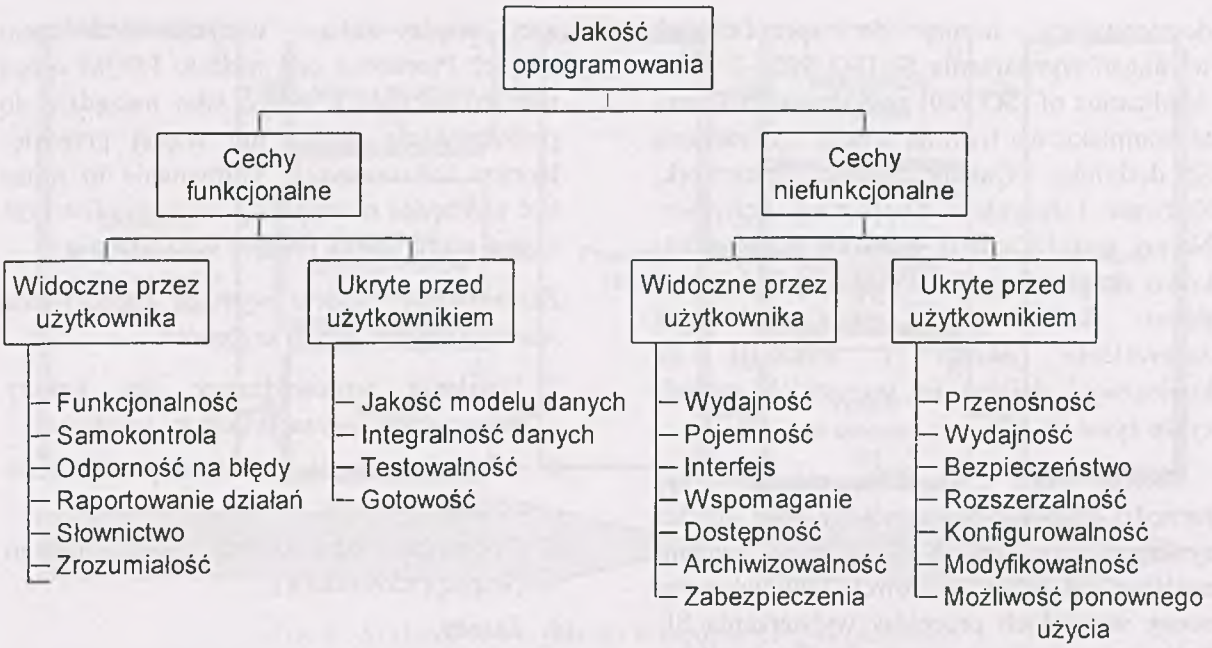
3. Cechy jakości systemów informatycznych

Cechy opisujące jakość SI, jej parametry czy też kryteria oceny nie są i nie mogą być jednorodne. Na SI można bowiem spojrzeć z wielu różnych stron – przekrojów. Inżynieria oprogramowania częściowo porządkuje te przekroje - podobnie jak w każdym projekcie inżynierskim. Przykładowo, w budownictwie projekt obiektu nie ogranicza się jedynie do przedstawienia jego architektury. W jego skład wchodzą również projekt konstrukcyjny, projekt technologiczny, projekt instalacji elektrycznej, grzewczej, hydraulicznej itp. Każdy z tych elementów oceniany jest przy pomocy innych kryteriów. Łączna ocena nie jest prostą (czy nawet ważoną) sumą poszczególnych ocen. Nawet dla tak dobrze zdefiniowanego obszaru jak budownictwo kompleksowa ocena jakości nie jest trywialnym problemem. Każdy SI można przedstawić z różnych stron i wyróżnić w nim (zgodnie z kanonami inżynierii oprogramowania) struktury: funkcjonalną, informacyjną, oprogramowania, techniczną i przestrzenną. W zasadzie w każdej z tych struktur można zdefiniować specyficzne kryteria jakości.

Kryteria jakości są definiowane w wielu metodykach, normach i publikacjach. Definicje te bardzo często są dość zróżnicowane. Przykładowo model McCalla (wykorzystany notabene jako baza w normie ISO 9126) wychodzi z trzech przekrojów jakościowych SI definiowanych z punktu widzenia użytkownika: funkcjonowanie systemu, przystosowanie do modyfikacji i mobilność oprogramowania (rys. 2).

Rys. 2. Drzewo jakości wg McCalla

Ciekawą klasyfikację cech jakości SI zaproponowano w [1] – rys. 3. Charakterystyki jakości zostały podzielone na dwie grupy: funkcjonalne i niefunkcjonalne. W każdej z nich wydzielono dwie podgrupy: widoczne (a więc w jawny sposób definiowane) i niewidoczne (tj. takie, które nie są definiowane, dostrzegane lub określane) przez użytkownika. Taka klasyfikacja pokazuje na istotność całego szeregu cech jakości oprogramowania nieznanych użytkownikowi lub niemożliwych do poznania bez ścisłego dostępu do dostawcy SI.



Rys. 3. Drzewo jakości wg Brinkwortha

Podobne charakterystyki zdefiniowano w raporcie Narodowego Instytutu Standardów i Technologii Stanów Zjednoczonych – NIST [14] dla systemów wielokrotnego wykorzystania:

- adaptacyjność,
- efektywność,
- kompletność,
- modularność,
- niezawodność,
- pielęgnowalność,
- poprawność,
- przenośność,
- zrozumiałość,
- zupełność.

Tamże zdefiniowano cały szereg różnych wskaźników (typu: gęstość błędów, złożoność cyklomatyczna, spójność więzów międzymodułowych czy pokrycie aplikacji testami), które umożliwiają ilościową ocenę poszczególnych cech jakościowych SI.

Bazując na kryteriach jakości (tj. na ich systemie i oszacowanych wartościach) można tworzyć różnego typu modele oceny jakości kompleksowej SI, np. metodą QFD (ang. *Quality Function Deployment*) [8] czy też stosując metody taksonomiczne [7].

Niezależnie jednak od zastosowanych kryteriów jakości SI ich wartości można ocenić jedynie post factum, tj. po wdrożeniu systemu. Dla klienta, który już zainwestował w system jest to za późno. Metody te nie mają więc racji bytu w sytuacji produkcji oprogramowania na zamówienie – indywidualnego. Ponadto sam proces wyznaczania całego szeregu wskaźników jakości SI jest długotrwały i dość kosztowny. Analizy niektórych systemów mogą trwać dłużej niż wymiana wersji oprogramowania.

4. Systemy zapewnienia jakości

Jedną z cech charakterystycznych SI jest ograniczona (z punktu widzenia praktyki) możliwość użycia do oceny ich jakości opisanych wcześniej metod (wykorzystujących głównie dane statystyczne). Metody te są szeroko stosowane w procesach produkcji masowej. Pozwalają dość dokładnie oszacować poziom jakościowy partii identycznych wyrobów. SI są produktami jednostkowymi i zamiast szacować liczbę błędnych produktów poprawniej jest nadzorować ich wytwarzanie. Taka jest idea większości znanych na świecie systemów jakości: amerykański, japoński i europejski. Ten ostatni znany jest jako seria norm ISO 900x. Istnieje specjalny dokument

dostosowujący normę do specyficznych wymagań wytwarzania SI ISO 9000-3: „The Application of ISO 9001 to Software". Normy te kompleksowo traktują proces wytwarzania SI definiując: Quality System Framework, Software Lifecycle i Supporting Activities. Normy serii ISO 900x wskazują dość ogólnikowo drogi osiągania wysokiej jakości produktu końcowego, pewną ideologię zapewnienia jakości i wskazują na konieczność działań na wszystkich etapach cyklu życia SI.

Normy serii ISO 900x nie zawierają narzędzi do porównania między sobą jakości systemów wytwarzania SI. Takie porównanie możliwe jest drogą ilościowej i kompleksowej oceny wszystkich procesów wytwarzania SI. Taka ocena wymaga ustanowienia mierników ilościowych, które pozwolą dość szczegółowo ocenić jakość poszczególnych procesów wytwarzania i opracować na tej podstawie kompleksową miarę. Próbą stworzenia takiego systemu jest znany model CMM (ang. *Capability Maturity Model*), wprowadzający 5 poziomów dojrzałości procesów wytwórczych i umożliwiający klasyfikację organizacji wg tego kryterium. Dla organizacji wytwarzających SI bliższa jest mutacja modelu CMM pod nazwą SE-CMM (ang. *System Engineering Capability Maturity Model*). Analogiczny system stworzyła European Foundation for Quality Management.

5. Model EFQM

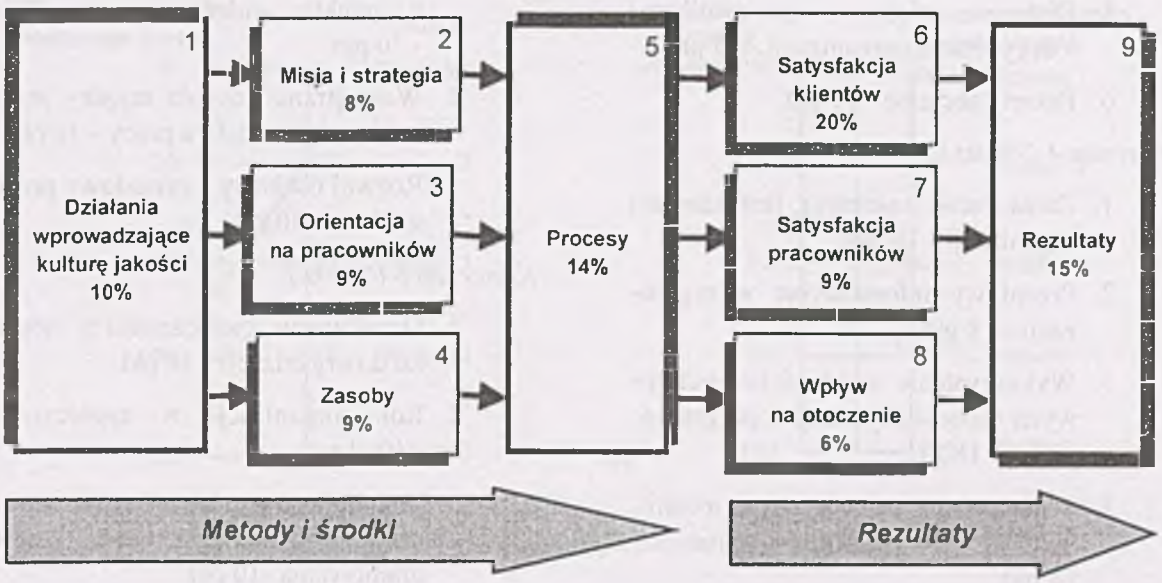
EFQM – European Foundation for Quality Management (EFQM) opracowała w 1991 roku model dla oceny przedsiębiorstw w zakresie wdrożenie Total Quality Management [15]. Pozwala on ilościowo ocenić jakość organizacji rozumianą jako satysfakcję klienta, pracowników i w szerszym rozumieniu – pozytywny wpływ na całe społeczeństwo. Model EFQM może służyć do samooceny firmy, wyszukiwaniu w niej elementów określających jakość a także służyć do porównania organizacji między sobą – wszystko pod kątem jakości. Pierwotne cele modelu EFQM mogą być rozszerzone i służyć jako narzędzie do porównywania dwóch lub więcej przedsiębiorstw softwarowych. Porównanie to może być niezbędne na etapie np. wyboru ofert firm czy na etapie auditu procesu wytwarzania SI.

Zastosowanie modelu wymaga oszacowania wartości następujących kryteriów:

1. Działania wprowadzające do kultury jakości (popularyzacja kultury jakości).
2. Misja i strategia działania producentów oprogramowania.
3. Orientacja na klienta wewnętrznego (współpracowników).
4. Zasoby.
5. Procesy.
6. Satysfakcja klientów zewnętrznych.
7. Satysfakcja klienta wewnętrznego.
8. Wpływ organizacji na otoczenie.
9. Rezultaty.

Kryteria te dotyczą dwóch obszarów: metod i środków (a więc oceny wysiłków projakościowych przedsiębiorstwa) i rezultatów (a więc oceny efektywności działań). Szczegółowe powiązania pomiędzy kryteriami, ich wagi oraz kolejność kwantyfikacji przedstawiono na rys. 4.

Model pozwala odwzorować zachowania i postawy decydentów organizacji oraz ich wpływ na organizację, klientów i pozostałe elementy otoczenia. Zachowania decydentów oceniane są z punktu widzenia działań prowadzonych na rzecz upowszechniania kultury jakości, dostosowania strategii i polityki organizacji do wymogów otoczenia i sposobu zarządzania dostępnymi zasobami mediów. Wszystkie te elementy poprzez procesy decyzyjne i wytwarzania oddziałują bezpośrednio na klientów i otoczenie, a miarą tego oddziaływania są wskaźniki ilościowe, umożliwiające porównanie kilku organizacji.



Rys. 4. Kryteria oceny jakości w modelu EFQM i ich wagi

Proponowane kryteria mają ogólny charakter i nie są bezpośrednio mierzalne. W związku z tym konieczne jest ich uzupełnienie dodatkowymi subkryteriami, które będą określać wielkości obserwowalne i mierzalne. Szczegółowa definicja kryteriów zależy od obszaru zastosowań modelu i specyfiki działalności prowadzonej przez organizację.

6. Ocena jakości procesów wytwarzania oprogramowania

W ocenie jakości proponuje się, by maksymalna liczba punktów możliwych do przyznania wynosiła 1000, a poszczególnym kryteriom przyporządkować taką liczbę punktów, która odpowiada przyjętej wadze danego kryterium (rys.4).

Proponuje się przyjęcie następujących obszarów analizy w ramach poszczególnych kryteriów oraz maksymalną dla nich liczbę punktów:

Kryterium 1 (100 pkt.):

1. Procedury umożliwiające wprowadzenie kultury jakości - 25 pkt.
2. Wymiana informacji między organizacją a klientem w aspekcie jakości - 25 pkt.
3. Rola decydentów w aktywnym propagowaniu i rozwoju kultury jakości wśród pracowników i otoczeniu przedsiębiorstwa - 25 pkt.
4. Kontrola działań związanych z wprowadzaniem i rozwojem jakości - 25 pkt.

Kryterium 2 (80 pkt.):

1. Budowa polityki i strategii działania organizacji z uwzględnieniem założeń kultury jakości - 20 pkt.
2. Rozwój polityki i strategii działania organizacji - 20 pkt.
3. Informowanie pracowników i otoczenia o strategii organizacji i sposobach jej realizacji - 20 pkt.
4. Aktualizacja założeń polityki organizacji i ulepszanie metod realizacji strategii działania z uwzględnieniem zmian w otoczeniu - 20 pkt.

Kryterium 3 (90 pkt.):

1. Zarządzanie zasobami ludzkich w aspekcie jakości działania - 15 pkt.
2. Podnoszenie kwalifikacji zawodowych pracowników - 15 pkt.
3. Zgodność celów pracowników i organizacji oraz kontrola ich realizacji - 15 pkt.
4. Ocena działań pracowników - 15 pkt.

5. Dialog między pracownikami a decydentami organizacji - 15 pkt.
6. Pakiety socjalne - 15 pkt.

Kryterium 4 (90 pkt.):

1. Zarządzanie zasobami finansowymi organizacji - 18 pkt.
2. Przepływy informacyjne w organizacji - 18 pkt.
3. Wykorzystanie urządzeń komputerowych i specjalistycznego oprogramowania - 18 pkt.
4. Zastosowanie nowoczesnych technik wytwarzania oprogramowania – 18 pkt.
5. Współpraca z dostawcami urządzeń komputerowych i oprogramowania - 18 pkt.

Kryterium 5 (140 pkt.):

1. Identyfikacja procesów (nie tylko wytwórczych) wpływających na jakość wytwarzania oprogramowania - 28 pkt.
2. Myślenie procesowe w odniesieniu do wytwarzania oprogramowania - 28 pkt.
3. Wykorzystanie kreatywności pracowników do ulepszania procesów - 28 pkt.
4. Metody wprowadzania zmian w przebiegu procesów - 28 pkt.
5. Ocena przebiegu procesów - 28 pkt.

Kryterium 6 (200 pkt.):

1. Ocena jakości usług przeprowadzona przez odbiorców oprogramowania - 125 pkt.
2. Ocena jakości usług przeprowadzona przez innych klientów zewnętrznych - 75 pkt.

Kryterium 7 (90 pkt.):

1. Jakość zabezpieczenia przez organizację właściwego przebiegu procesów wytwarzania oprogramowania z punktu widzenia pracowników - 30 pkt.
2. Wewnętrzne stosunki między pracownikami i atmosfera pracy - 30 pkt.
3. Rozwój osobowy i zawodowy pracowników - 30 pkt.

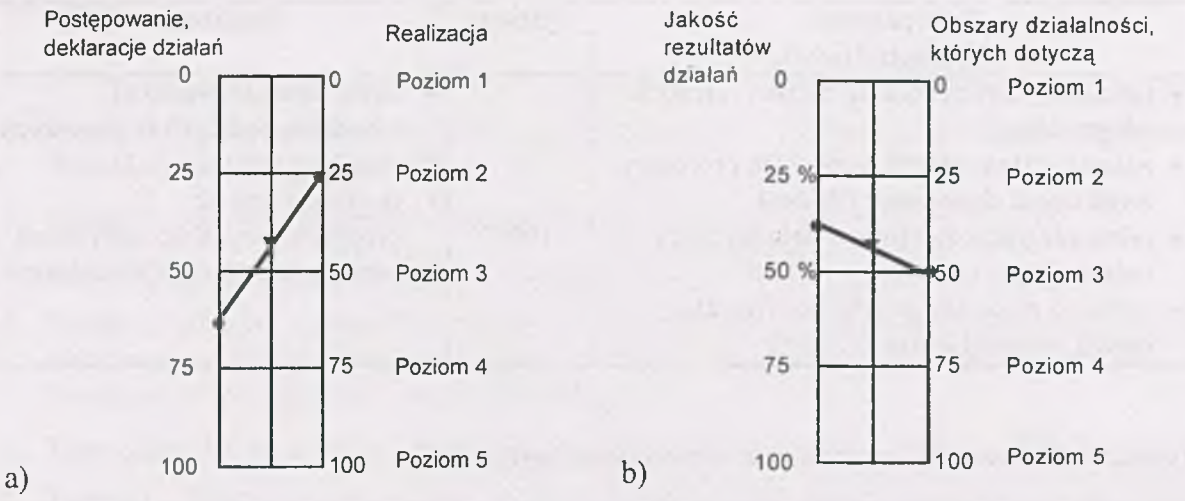
Kryterium 8 (60 pkt.):

1. Oczekiwania społeczności w stosunku do organizacji - 10 pkt.
2. Rola organizacji w społeczności - 10 pkt.
3. Współpraca organizacji z dostawcami sprzętu komputerowego i oprogramowania- 10 pkt.
4. Image organizacji - 10 pkt.
5. Popularyzacja organizacji w otoczeniu - 10 pkt.
6. Wpływ działalności organizacji na otoczenie - 10 pkt.

Kryterium 9 (150 pkt.):

1. Wskaźniki oceny finansowej działalności organizacji (np. realizacja budżetu, wartości odnowy sprzętu technicznego) - 75 pkt.
2. Wskaźniki oceny poziomu zadowolenia klientów zewnętrznych (np. ogólna ilość wykonanych usług, wartość reklamacji do ogólnej wartości usług, czas reakcji na zgłaszane problemy z eksploatacją oprogramowania) - 75 pkt.

Ocena wartości kryteriów wymaga zastosowania nomogramów, różnych dla każdego obszaru działań. W ocenie metod i środków analizowane są z jednej strony postępowanie i postawa decydentów organizacji, ukierunkowane na jakość, z drugiej - praktyczna realizacja deklarowanych postulatów zapewnienia jakości. Nomogram stosowany w tej ocenie jest przedstawiony na rys. 5a. W ocenie stopnia realizacji celów pod uwagę bierze się jakość rezultatów działań oraz zakres działań, których te rezultaty dotyczą. Nomogram stosowany w tego rodzaju ocenie jest przedstawiony na rys. 5b.



Rys.5. Nomogramy oceny kryteriów w modelu EFQM

Poszczególne oceny wyznaczane są niezależnie dla lewej i prawej strony nomogramu (tab. 1 i 2), a końcowa ocena według danego kryterium ustalana jest w punkcie przecięcia się prostej łączącej wartości pojedynczych ocen z osią pionową. W przypadku ustalenia ocen na poziomach pośrednich konieczne jest dokonanie liniowej interpolacji.

Tablica 1. System oceny metod i środków działań projakościowych [15]

Postępowanie, deklaracja działań	Poziom	Realizacja
• działania pozorowane lub nie nadające wartości, • słowne deklaracje nie mające pokrycia w możliwościach ich realizacji	1 0%	• niska skuteczność działania, • brak lub znacznie ograniczone możliwości realizacji przyjętych deklaracji projakościowych
• jednostkowe symptomy postępowania projakościowego, • okazjonalna kontrola działań, • częściowa integracja metod działania z codzienną pracą,	2 25%	• zaangażowanie około 1/4 potencjału przedsiębiorstwa. • realizacja działań projakościowych we wszystkich znaczących obszarach działania
• regularne ulepszanie metod działania, • definiowanie założeń działań projakościowych zrozumiałych dla wszystkich osób, • dobra integracja metod działania z codzienną pracą, • regularna kontrola skuteczności działań poprzez ocenę rezultatów	3 50%	• zaangażowanie około 1/2 potencjału przedsiębiorstwa, • realizacja działań projakościowych we wszystkich znaczących obszarach działania
• znajomość i rozumienie założeń postępowania projakościowego przez wszystkie osoby, • procedury stałych kontroli działań w trakcie ich przebiegu, • dobra integracja metod działania z codzienną pracą i planowanym rozwojem przedsiębiorstwa	4 75%	• zaangażowanie około 3/4 potencjału przedsiębiorstwa, • realizacja działań projakościowych we wszystkich znaczących obszarach działania

Postępowanie, deklaracja działań	Poziom	Realizacja
• procedury stałych kontroli działań w trakcie ich przebiegu, • znane i zrozumiałe dla wszystkich procedury zwiększania skuteczności działań, • pełna integracja działań z codzienną pracą i planowanym rozwojem, • działania mogą służyć jako wzorzec dla innych przedsiębiorstw	5 100%	• pełne zaangażowanie sił i środków będących w dyspozycji przedsiębiorstwa, • realizacja działań projakościowych we wszystkich znaczących obszarach działania

Tablica 2. System oceny rezultatów działań projakościowych [15]

Jakość rezultatów działań	Poziom	Obszary działalności, których dotyczą rezultaty
• niska, odbiegająca od norm akceptowanych przez klientów	1 0%	• wyniki oceny dotyczą pojedynczych obszarów działalności firmy
• pozytywny kierunek zmian kilku rezultatów oceny lub utrzymanie zadawalającego ich poziomu w okresie min. 3-ech lat, • w niektórych obszarach działania zgodność rezultatów oceny jakości z pojedynczymi celami organizacji	2 25%	• wyniki oceny dotyczą kilku znaczących obszarów działalności przedsiębiorstwa
• wiele składników oceny wykazuje pozytywny kierunek zmian lub utrzymanie zadawalającego ich poziomu w okresie min. 3-ech lat, • w wielu obszarach działania zgodność rezultatów oceny jakości z pojedynczymi celami organizacji, • możliwe porównania jakości działania z innymi organizacjami w kilku obszarach, • pojedyncze oceny zbliżone do wymagań TQM	3 50%	• wyniki oceny dotyczą wielu znaczących obszarów działalności przedsiębiorstwa
• większość składników oceny wykazuje wyraźny pozytywny kierunek zmian lub utrzymanie ich wysokiego poziomu, • zgodność oceny jakości z wieloma celami organizacji, • możliwe porównania jakości działania z innymi organizacjami w wielu obszarach, • wiele oceny zbliżonych do wymagań TQM	4 75%	• wyniki oceny dotyczą większości znaczących obszarów działalności przedsiębiorstwa
• wyraźne pozytywne trendy zmian składników oceny w okresie min. 5 lat we wszystkich obszarach działalności, • całkowita zbieżność oceny z celami organizacji, • wzorzec w wielu obszarach działania, • oceny zgodne z wymaganiami TQM,	5 100%	• wyniki oceny dotyczą wszystkich znaczących obszarów działalności przedsiębiorstwa

Literatura

1. Brinkworth J.: What is a High Quality Software System? SQM, No 27, 1995
2. Begier B.: System jakości w tworzeniu oprogramowania. IV Konferencja: Komputerowe Systemy Wielodostępne, Bydgoszcz, 1998, str. 101-111.
3. Begier B.: Zapewnienie jakości oprogramowania. BIS'97, Poznań, 1997, str. 177-191.
4. Byzia T.: Seria: Zarządzanie jakością oprogramowania, CW, 1999.
5. Dudak E., Korbel L., Miłosz M., Smołka J.: Problemy projektowania informatycznego systemu zarządzania przedsiębiorstwem z wykorzystaniem mikrokomputerów. Materiały V Krajowej Konferencji Informatyków. Poznań, 1987, str. 487-490.
6. Kaner Cam: The Impossibly of Complete Testing. Software Quality Assurance, 1997.
7. Korbel L., Miłosz E., Miłosz M.: Systemy zarządzania relacyjnymi bazami danych oraz ich przydatność do budowy informatycznych systemów zarządzania. Informatyka, nr 4, 1993, str. 8-12.
8. Krawczyk H.: Modele i ocena jakości systemów informatycznych. Czternaste Jesienne Spotkania PTI, Mrągowo, 16-20.11.1998, str. 31-45.
9. Lowell J.A.: Improving Software Quality. An Insider's Guide to TQM. John Wiley & Sons, Inc, 1993.
10. Maier Charles: Estimating the Cost of Quality. Software Quality Management, issue 4/5, 1990.
11. Miłosz M., Muryjas P.: Porównanie jakości procesów wytwarzania oprogramowania przy pomocy modelu EFQM. I Konferencja Informatyk Zakładowy. Kazimierz Dolny, 28-29.05.1998. Informatyka Stosowana, nr S3/98, str. 117-125
12. Miłosz M.: Elementy sterowania jakością oprogramowania użytkowego mikrokomputerów. Biuletyn Informacyjny Instytutu Transportu Samochodowego. Warszawa, nr 2, 1994, str. 19-29.
13. Qualität und Qualitätsstandards in der ausserschulischen Jugend- und Erwachsenenbildung. Materialen des Bundesministeriums für Familie, Senioren, Frauen und Jugend. Januar, 1997.
14. Quality Characteristics and Metrics for Reusable Software (Preliminary Report). U.S. Department of Commerce, National Institute of Standards and Technology, May 1994.
15. Total Quality Management in der Handelsakademie und Handelsschule. Qualitats-hak. Modell zur Seldsbewertund nach der European Foundation for Quality Management. Juni, 1996.
16. Zalewski R.: What can quality management benefit from information system? Int. Conference Business Information Systems'97, Poznań, 1997, pp. 147-163.

Marian Kuraś
Agnieszka Zając

Marian Kuraś[1], *Agnieszka Zając*[2]
Katedra Informatyki Akademii Ekonomicznej w Krakowie
e-mail: [1] eikuras@cyf-kr.edu.pl, [2] eizajac@cyf-kr.edu.pl

TI -- konieczny wydatek czy korzystna inwestycja?

Streszczenie: Wiele organizacji wprowadza technikę informacyjną pod wpływem nacisku otoczenia – intuicyjnie, nie w pełni świadomie, bez wizji i bez świadomości trudności. Zarządy firm decydują się na wydawanie wielkich środków często bez wyobrażenia o efektach ich wykorzystania. Przy braku zaufania i oszczędzaniu środków na ekspertyzy użytkownicy staja się łatwymi ofiarami eksperymentatorów. Liczne przypadki nieudanych projektów są rezultatem uznawania nakładów na TI za konieczne (?) wydatki a nie nadzwyczaj korzystne inwestycje. Celem wykładu będzie wskazanie przyczyn takich praktyk i ich skutków oraz wskazanie kierunków reorientacji.

... Cześć. Jak się Masz? Co robisz?
Programuję.
A co programujesz?
Coś, coś, coś.
A co, co co?
Właśnie jeszcze nie wiem.
A jak sądzisz?
Jeszcze trochę poprogramuję i wtedy zobaczę. Może użytkownik będzie coś wiedział?
Parafraza z Kubusia Puchatka

Introdukcja / zagajenie

Są społeczeństwa, których członkowie za co się wezmą, zarabiają pieniądze. Są też takie plemiona, które cały czas rozwijają naukę i technologię i cały czas dopłacają do tego. Dla pieniędzy estymy nie mają, chociaż nie można powiedzieć, że ich nie chcą. Pożądają ich chyba jeszcze bardziej niż społeczeństwa, które je na wszystkim zarabiają. Potrzebują tych pieniędzy na stworzenie sobie warunków do przyspieszania badań i rozwoju technologicznego, co jest bardzo kosztowne. Trzeba do tego nowoczesnych komputerów na przykład. Konstruowanie nowoczesnej techniki jest niemożliwe bez zakupienia ultranowoczesnych komputerów, które produkują społeczeństwa pierwszego rodzaju. Jak już powiedziano, produkują one komputery i jednocześnie zarabiają pieniądze. Plemiona drugiego rodzaju nie potrafią produkować komputerów (ani wielu innych rzeczy) i na dodatek nie umieją zarabiać pieniędzy. A potrzeby tych plemion są bardzo wysokie – muszą przecie prowadzić badania oraz rozwijać wiedzę i technikę.

Każdy obserwator, życzliwy plemionom II rodzaju, zastanawia się, dlaczego też mają one tak mało pieniędzy a tak dużo wydają na badania i rozwój techniki oraz jej zastosowań. Kiedy dowiaduje się, że na te cele wydają bardzo mało, traci zdolność pojmowania i sporo życzliwości. Ostrzej wówczas patrzy i krytycznie traktuje informacje. Radzi tedy, iżby zmienić sposób postępowania. Sugeruje byśmy z rozwijania istniejącej techniki zrezygnowali skoro ona już istnieje i nauczyli się wreszcie na niej zarabiać. Jeśli zastanowić się to wcale nie jest głupie. Przecież ją kupujemy i chyba możemy (powinniśmy) mieć korzyści większe niż poniesione koszty.

Niektórzy z doradców posuwają się dalej. Mówią: jeśli nie potraficie stosować tej nowej techniki, powinniście zaniechać badań, gdyż nie mogą one doprowadzić do powstania niczego nowego, niczego co się sprzeda. Tłumaczą: technika powstaje i rozwija się tam gdzie są potrzeby i to wiążące się z dobrym zarobkiem. Np. technika informacyjna powstała, bo była potrzebna do wyprodukowania broni atomowej. Zastosowano ją potem w biznesie, gdzie nie tylko zarobiła na siebie, ale jeszcze przyniosła różnorodne dodatkowe korzyści. Użytkownicy zażyczyli sobie wtedy potężniejszych komputerów, a że mieli czym zapłacić dostali, czego chcieli. Chytrzy konstruktorzy i producenci szybko zwęszyli wielki interes i zaoferowali biznesowi nowe potężne konstrukcje i pomoc w ich wykorzystaniu. Biznes zapłacił i znów zarobił z nawiązką, a że jest nienasycony pomógł producentom stworzyć nowe, potężne komputery, które wykorzystał itd.[1]

Wykład jest przeznaczony tylko dla osób, które nie rozumieją skąd biorą się nasze niedostatki i nie wstydzą się zadawać prostych pytań. Wielu menedżerów, użytkowników i informatyków poszukuje odpowiedzi na takie pytania i znajduje je głównie w książkach par excellence nienaukowych. Nauka i technika odpowiada tylko na pytania bardziej wysublimowane. Nam tymczasem trzeba refleksji i rozwiązania podstawowych problemów, którymi nauka nie zajmuje się już. Stąd podstawową literaturą wykorzystaną przez autorów jest *Kubuś Puchatek*, *Chatka Puchatka* i *Słownik Kubusia Puchatka*. Jako literaturę uzupełniającą wykorzystano szereg pozycji poświęconych doświadczeniom i radom dla tych, którzy na TI chcą zarobić (np. Cortada: 1998; Bysinger & Knight: 1996; Hoffman: 1994; Hogbin & Thomas: 1994; Karake: 1992; Keyes:1995; Luftman: 1996; Lincoln: 1990. Gdy już zgodnie z normami spisano wykorzystane źródła, trzeba wyjaśnić cel, jakiemu wykład ma służyć. Jest nim lepsze zrozumienie co jest celem stosowania TI w biznesie. Pokrótce można to sformułować jako robienie biznesu, czyli jeszcze mniej elegancko zarabianie pieniędzy.

1. Czy TI może służyć zarabianiu pieniędzy?

Wspaniała Myśl ...

Ogólnie można przyjąć, iż o ile przekonanie, że
każde zwierzę powinno mieć dom,
to Dobra Myśl, o tyle Wspaniała Myśl wiąże się
z podjęciem decyzji o zbudowaniu chatki
dla innego zwierzęcia, gdyż jego obecne warunki
bytowe są nieodpowiednie.
Może to prowadzić do Okropnych Pomyłek, istnieje
jednak spore prawdopodobieństwo
otrzymania nieprzeliczonych nagród w dziedzinie
architektury.
Milne: 1995; Melrose: 1997

Bardzo długo nie wypadało zarabiać pieniędzy. Przynajmniej oficjalnie. Można twierdzić, że wiele jednostek zabrało się za informatykę nie dlatego, że jest ona piękna (wiadomo, że jest) a dlatego, że daje wymierne korzyści. Korzyści te wynikają z chęci klientów oszczędzenia na wydatkach na TI. Otóż i mamy dowód na przydatność TI jako czynnika obniżki kosztów. Szkoda, że większość tych praktyk jest fikcją, gdyż taka obniżka odbywa się kosztem jakości i budżetu. Często działania te podejmowało się z powodzeniem w firmie, w której oficjalni wykonawcy nie potrafili spełnić oczekiwań klientów.

Zwykle użytkownicy zarówno ci zadowoleni jak i ci, których zastosowanie TI

[1] Biznes opiera się na pazerności w przeciwieństwie do maksymalnego zaspokajania potrzeb społecznych, co jest celem systemu, w którym nie ma biznesu a więc nie ma produkcji ani usług, przeto potrzeb społecznych ani tym bardziej indywidualnych zaspokajać nie można. Sytuację taką nazywa się sprawiedliwością społeczną.

w zupełności nie satysfakcjonowało nie są w stanie uzasadnić swych opinii. Nie potrafią powiedzieć, jakie są koszty TI i nie precyzują czego nie osiągnęli i jak wiele stracili na wprowadzaniu TI. Nie mówią, gdyż celów nie formułowali przed zamierzeniem i nie oszacowali wcześniej efektów a kosztów do dziś nie liczą. Oznacza to ni mniej ni więcej, że w ogóle TI nie sterują, gdyż nie można sterować czymś, czego się nie mierzy (DeMarco: 1982). Jeśli użytkownicy nie określali realnych celów ani kryteriów oceny i nie liczą wydatków na TI, nie można się dziwić, że zastosowania im nie służą.

Wiele lat doświadczeń w stosowaniu TI przyniosło mierne efekty. Zastosowania niekiedy tylko wykraczają poza elementarne przetwarzanie polegające na rejestrowaniu zdarzeń i tworzenie zestawień zbiorczych niezbędnych do opracowania sprawozdań statystycznych. Menedżerowie postrzegają TI co najwyżej jako narzędzie umożliwiające rozwiązywanie dzisiejszych i wczorajszych problemów. Do tych celów wykorzystują nieraz bardzo zaawansowaną TI. Czy to się opłaca? Nikt nie wie, gdyż menedżerowie nie potrafią określać celów i oceniać efektywności. Informatycy indagowani o efekty zwykle odpowiadają, że trudno jest oszacować efekty (jak gdyby ktoś mówił, że jest to łatwe). Kiedy mają zdeklarować się, twierdzą, że otrzymane efekty są niewymierne. Niewymierne znaczy żadne. Koszty zawsze są wymierne (jeśli się je liczy) i korzyści muszą być mierzalne – inaczej nie można o nich mówić.

Efekty mogą i powinny pojawić się, trzeba tylko zrozumieć jedno. Informatyka jako taka nie daje efektów a jedynie wiąże się z kosztami. Efekty powstają w wyniku dostarczania i wykorzystania **INFORMACJI.** Jeśli informatyki nie wykorzystuje się do tworzenia, dystrybucji informacji, jest ona nieprzydatna, nikomu oprócz informatyków (producentów, sprzedawców oraz twórców oprogramowania).

2. Jak TI może dodawać wartości do produktów i procesów?

Różnie Bywa na Tym Świecie (dogmat filozoficzny)

Twierdzenie, zgodnie z którym wszystko, co powinno ciebie dotyczyć, ciebie nie dotyczy

Milne: 1995; Melrose: 1997

Misją firmy jest tworzyć wartość dla klienta (Hammer, 1996)[2]. Jedynie pod tym warunkiem klient jest skłonny kupić od klienta produkt i/lub usługę. Firma może oczywiście sprzedawać swoje wytwory klientom, którzy nie mają sprecyzowanych potrzeb czyli nie mają wymagań dotyczących wartości związanej z produktem. Ale jak długo? Ci wytwórcy, którzy potrafią więcej dodać wartości do produktu czy procesu lepiej trafiają do klienta a więc więcej sprzedają i to po korzystnej cenie.

Wartość dla klienta przedstawiają specjalne (wyjątkowe) cechy wyrobu (usługi), dla których będzie skłonny ponieść większe koszty. Producent powinien poszukiwać możliwości dodania wartości do swego produktu (usługi), gdyż w ten sposób może zyskać przewagę nad konkurentami zdobywając nowych klientów. W przypadku banku wartość dla klienta mają produkty bankowe i serwis, który zapewni mu np. wysokie oprocentowanie wkładów, dostępność informacji oraz oszczędność czasu w kontaktach z bankiem i niskie opłaty za usługi.

Bank zatem może zyskać przewagę konkurencyjną poprzez takie działania jak obniżenie kosztów, zapewnienie klientowi sprawnej i szybkiej obsługi oraz informacji. Warto zauważyć, że żadna organizacja nie zdobędzie przewagi konkurencyjnej w wyniku wprowadzenia techniki informacyjnej (zakupu, stworzenia systemu informatycznego). Trudno sobie zarazem wyobrazić żeby bez jej kompetentnego zastosowania można było osiągnąć -- wymagane do zyskania przewagi konkurencyjnej i będące celem -- cechy produktów i/lub usług.

[2] Trzeba pamiętać, że klient kupuje nie wytwór, ale jego zdolność do wykonywania pewnych funkcji, jakie są mu potrzebne. Poszukuje nie 'systemu informatycznego', a czegoś, co pozwoli pracownikom sprawnie i szybko zarejestrować dane, gromadzić je, wyszukiwać, przetwarzać i udostępniać.

TI jest narzędziem, jakim należy się posłużyć do wprowadzenia nowych (przeprojektowanych) produktów i procesów. Organizacja daje nową wartość nie poprzez zaopatrzenie się w sprzęt i oprogramowanie a dzięki temu, że oferuje nowe (przeprojektowane) produkty i/lub procesy, które mają specjalne walory dla klienta. Te nowe produkty i usługi są lepsze i tańsze, gdyż celowo zastosowano technikę informacyjną. Skomputerowanie starych produktów, procesów i procedur nie zmienia ich przydatności dla klientów a nierzadko nawet pogarsza ich cechy. Nie sami więc informatycy i nie sama technika informacyjna tworzą wartość dla klienta a użytkownicy i informatycy. Współpraca użytkowników przyniesie pożądane efekty, pod warunkiem że utworzy się zespół z jasno określonym celem, kryteriami oceny, przydzielonymi zasobami. Zespół mający za zadanie zaprojektowanie nowego produktu/procesu powinno się budować zgodnie z regułami podawanymi w literaturze (np. DeMarco & Lister:1987).

3. Dla kogo w końcu są systemy informacyjne[3] - dla użytkowników czy dla klientów?

Wychlupanie – przyjście komuś z żywiołową pomocą bez jego świadomej zgody.

Milne: 1995; Melrose: 1997

W zasadzie dopiero przed kilkoma laty pogodzilimy się z tym, że wszystkie te komputery, sieci i oprogramowanie są nie dla nas a dla użytkowników. To oni mają pieniądze i odkąd trochę poduczyli się zaczynają stawiać coraz większe wymagania (chociaż wciąż nie wiedzą czego chcą) a pieniędzy na niezbędne zakupy okropnie skąpią. Każdy zakup kwestionują i żądają szczegółowego uzasadnienia. Nauczyli się niewiele, ale mają coraz większe pretensje. Wciąż nie potrafią poprawnie opisać algorytmów, ale mają coraz większe wymagania i są wobec nas bardziej krytyczni. O ile odmawiają nam środków na niezbędne zakupy niezbędnego sprzętu i nowszego oprogramowania, to nie brakuje im środków na drogich doradców, których jedynym celem wydaje się kwestionowanie naszych koncepcji. Za namową owych doradców użytkownicy domagają się od nas wizji, strategii i planów. To może dobre jest gdzieś na zachodzie, ale 'nasza specyfika' jest zupełnie odmienna. U nas nigdy nie wiadomo, co się stanie za dwa lata i jaka będzie dostępna technologia.

Ostatnio niektórzy użytkownicy anonsują kolejną zmianę -- zapowiadają, że muszą wszystko robić tak by ich klientom 'dodać wartości'. Twierdzą, że organizację muszą zmieniać tak, iżby zmiany dawały korzyści i służyły interesom ich klientów. Posuwają się do twierdzenia, że zastosowania techniki informacyjnej mają być podporządkowane klientom. Czy nas ma to obchodzić? My jesteśmy fachowcami i trzeba nam dać dokładny opis sytuacji, wszystkie obowiązujące przepisy, struktury danych oraz opis algorytmów a oni mogą sobie w dowolny sposób zmieniać organizację. Nic nas nie interesują ich klienci ze swoimi grymasami. Nie możemy tego brać pod uwagę, należy to do użytkowników. Nas wystarczająco dużo kosztuje rozpoznanie potrzeb i wymagań użytkowników. Niech oni troszczą się i przejmują swoimi klientami.

Można obawiać się, że zmiany mogą sięgać dalej. Użytkownicy, zapatrzeni w Zachód, bezkrytycznie przyjmują wzorce obce naszej kulturze i mentalności. Czasami nawet, kiedy nie wiedzą, o co chodzi w tamtejszych podręcznikach, szukają wsparcia w firmach consultigowych (nie znających naszej specyfiki) i próbują bezkrytycznie wprowadzać niesprawdzone teorie na nasz grunt. Przede wszystkim w zupełnie nietrafny sposób charakteryzują otoczenie doszukując się w nim takich charakterystyk jak: (Keyes: 1995; Hoffman: 1994; Hogbin & Thomas: 1994; Lincoln: 1990):

- turbulencje,
- fluktuacje,
- globalizacja,
- deregulacja,
- hyper-konkurencja,
- decentralizacja,
- alianse strategiczne,

[3] Systemy informatyczne są dla informatyków.

- ultra-jakość,
- konwergencja,
- oczekiwania konsumentów,
- organizacje wirtualne,
- przełomy technologiczne i metodologiczne

W tej zachodniej literaturze powstają nawet, przeczące zdrowemu rozsądkowi i doświadczeniu całego pokolenia informatyków, teorie przeczące trafności i nieuniknności praw Murphy'ego (np. Chew i in.: 1991). Inne opracowania (np.: Bysinger & Knight: 1996) zalecają zmianę dotychczasowych reguł i przewrócenie do góry nogami sprawdzonego podziału obowiązków i odpowiedzialności. Autorzy ci twierdzą na przykład, że:

-- systemów informacyjnych nie można powierzać technikom (*techies*);

-- systemy informowania kierownictwa powinny być proste;

-- menedżerowie powinni rozpocząć (r)ewolucję w systemach informacyjnych pamiętając, że to oni są odpowiedzialni za procesy;

-- menedżer powinien sterować sposobem wykorzystania techniki informacyjnej;

-- menedżer jest klientem informatyka i może postępować jak typowy klient.

Można wyobrazić sobie, że podobne pomysły ktoś eksperymentalnie może wprowadzić w zachodnich organizacjach otwartych na zmiany, doświadczonych w wykorzystaniu informatyki i o ustalonej wysokiej pozycji na rynku. W 'naszych warunkach', przy 'naszej specyfice' może się to jednak okazać ryzykowne lub zupełnie niewykonalne.

4. Co jeszcze mogą wymyślić użytkownicy?

Kłopotliwy --Określenie Zwierzęcia (Małego lub Dużego), Rzeczy bądź Czynności, które z jakichś powodów zmuszają cię do zawołania tak Głośno, jak tylko twa Postura pozwala: 'Masz ci los!'

Milne: 1995; Melrose: 1997

Można mieć poważne obawy, że to nie koniec pomysłów, jakie użytkownicy zamierzają bezkrytycznie zaadaptować z firm działających w zupełnie odmiennych warunkach. Coraz częściej sami użytkownicy a także informatycy bajdurzą o misji, celach, strategii, skuteczności i efektywności, traktowaniu jako zwykłej inwestycji nakładów koniecznych na przełomie wieków na utrzymanie minimalnego poziomu cywilizacyjnego. Użytkownicy nie zdając sobie sprawy ze znaczenia i cywilizacyjnej roli techniki informacyjnej często stawiają jej nierealne wymagania oczekując szybkiego zwrotu nakładów i dodatkowych zysków.

Niektórzy użytkownicy twierdzą, że technika informacyjna rozwinęła się dzięki umiejętności jej zastosowania w celu sprawnego zorganizowania firmy, zyskania przewagi konkurencyjnej i w konsekwencji zarobienia pieniędzy na dalsze jej stosowanie. Uważają, że nie byłoby ich stać na poniesienie wielkich wydatków, jeśli jej wykorzystanie, nie przynosiłoby wielkich korzyści. Jednocześnie uważają, że nie powinni ponosić wydatków na prowizoryczne rozwiązania służące celom bieżącym, ale inwestować w infrastrukturę przyszłego systemu informacyjnego. Chcą w związku z tym, żeby informatycy przedstawili wizję przyszłego systemu i w dzisiejszych projektach uwzględniali możliwości rozbudowy i modernizacji. Chcą więc strategicznego planu budowy przyszłego systemu, podczas gdy informatycy uważają, że nie mogą dziś przewidzieć przyszłych potrzeb użytkowników ani przyszłych rozwiązań technologicznych.

Sytuacja staje się jednak coraz trudniejsza i musimy zacząć rozumieć wymagania użytkownika. Skoro użytkownik ma pieniądze i chce je wydać na coś, co możemy mu sprzedać powinniśmy przystać na jego warunki. Na wstępie menedżerowie i my musimy przyjąć jako aksjomat, że to ludzie dający usługę kreują wartość a nie TI. Menedżer i informatyk muszą przyjąć do wiadomości, że sukces wprowadzania TI wynika z psychologii w 95% i z technologii w 5% (Peters, wg: Davenporta: 1997). Menedżer rozumiejąc to musi wiedzieć na czym chce / może zarobić, żeby koszty inwestycji zwróciły się. Musi zaprojektować nową organizację, określić cele informatyki i potrzebne informacje. Musi też zaplanować

i wskazać, jakich oczekuje efektów i jakie może ponieść nakłady. Wspólnie z informatykami musi sformułować rzeczywiste problemy lub okazje do usprawnienia procesu, określić szanse strategiczne i/lub możliwości wygenerowanie większego dochodu (Bysinger & Knight: 1996). Wspólnie muszą wskazać krytyczne czynniki sukcesu (Rockart: 1979, 1982; Bullen: 1996). Powinni zainteresować się i ewentualnie zaadaptować doświadczenia innych (Cortada: 1998).

Wnioski

Nie Miejcie do Mnie Pretensji
(wypowiedzenie rozkazujące)
Zdanie, które należy wypowiedzieć,
gdy nie bierze się odpowiedzialności za to,
co może się stać,
choć uczestniczy się w tym, co się dzieje.

Milne: 1995; Melrose: 1997

Praca w informatyce to naprawdę ciężki kawałek chleba. Nasza sytuacja jest dodatkowo trudniejsza niż w innych krajach, gdyż wspomaganie zarządzania jest pozbawione podstaw, które mają nasi odpowiednicy w krajach, w których gospodarka rynkowa jest rzeczywistością. Menedżerowie mają tam wizje i pomysły ich urzeczywistnienia oraz potrafią formułować swoje potrzeby informacyjne. Nam menedżerowie zlecają wykonanie prostych aplikacji ewidencyjnych, potem oczekują, żeby przekształcić je w systemy informowania kierownictwa, systemy wspomagania decyzji i w ... co jeszcze nowego pojawi się w publikacjach profesjonalnych i w reklamach. Potwierdza się w naszej praktyce to, o czym pisał DeMarco: *użytkownicy próbują określać za nas technologię i nawet dostawców a nam zostawiają rozwiązanie ich problemów organizacyjnych.*

Mamy dwie strategie do wyboru. Pierwsza to opanowanie wiedzy i umiejętności menedżerskich oraz formułowanie dla organizacji misji, zdefiniowanie celów, strategii ich osiągania, stosownych miar, rozwiązań organizacyjnych – procesów i struktur, potrzeb informacyjnych strategii informacyjnej, strategii SI/TI, architektury SI i procesu jego tworzenia. Chyba wystarczy. Powinniśmy tworzyć wielkie firmy consultingowe. Druga strategia to oczekiwanie na lepsze czasy, kiedy to menedżer będzie zarządzał: zdefiniuje swój biznes. Nie, jakie dziś wykonuje się procedury, jakie gromadzi się dane i jakie generuje raporty, a co i jak powinno się robić, jakie dane gromadzić, jakie dane udostępniać, by zwiększać udział w rynku oraz wynikające z tego dochody. Jest też jeszcze strategia 'spalonej ziemi'. Brać każde zlecenie jak leci i robić, jak się da nie przejmując się nadziejami i trudnościami użytkowników i klientów. A normalność i tak kiedyś do nas przyjdzie. Wtedy trzeba będzie zacząć przejmować się regułami sztuki i od nowa szukać klientów.

Bibliografia:

Allen, R.E. (1995). Szkoła menedżerów Kubusia Puchatka. Poznań: Dom Wydawniczy REBIS.

Allen, R.E., Allen, S.D. (1995). (1999). Kubuś Puchatek i rozwiązywania problemów. Poznań: Dom Wydawniczy REBIS.

Bullen, C.V. (1996). Reexamining Critical Success Factors [in:] Umbaugh (1996).

Brown, M.G. (1996). Keeping Score. Using The Right Metrics to Drive World-Class Performance. New York: Quality Resources.

Bysinger, B., Knight, K. Investing in Information Technology (1996). New York: Van Nostrand Reinhold.

Cash, J.I.Jr., McFarlan, F.W., McKenney, J.L., Applegate, L.M. (1991). Corporate Information Systems Management. Text and Cases. 3rd Ed. Homewood, Ill.: Irwin.

Chew, W.B., Leonard-Barton, D., Bohn, R.E. (1991). Beating Murphy's Law. Sloan Management Review. No. 5.

Cortada, J.W. (1998). Best Practices in Information Technology: How Corporations Get the Most Value from Exploiting Their Digital Equipments. Upper Saddle River, NJ.: Prentice Hall.

DeMarco, T. (1982). Controlling Software Projects.Englewwod Cliffs, N.J.:Yourdon Press.

DeMarco, T., Lister, T. (1987). Peopleware: Productive Projects and Teams. New York: Dorset House Publishing Co.

Hammer, M. (1996). Beyond Reengineering. New York: Harper Collins Publishers.

Hoffman, G.M. (1994). The Technology Payofff. How to Ptofit with Empowered Workers in the Information Age. Burr Ridge, Ill.: IRWIN.

Hoff, B. (1995). Tao Kubusia Puchatka. Poznań: Zysk i S-ka Wydawnictwo s.c.

Hoff, B. (1998). Te Prosiaczka. Poznań: Zysk i S-ka Wydawnictwo s.c.

Hogbin, G., Thomas, D.V. (1994). Investing in Information Technology. Managing the Decision-making Process. London-New York: Mc Graw-Hill Book Company.

Karake, Z.A. (1992). Information Technology and Management Control. Westport, Ct.: Praeger.

Keyes, J. (1995). Solving the Productivity Paradox. TQM for Computer Professionals. New York: McGraw-Hill, Inc.

Lincoln, T. (1990). Managing Information Systems fo Profit. Chichester – New York: John Wiley & Sons.

Luftman, J.N. (ed.): 1996. Competing in the Information Age. New York–Oxford: Oxford University ress.

Melrose, A.R. (1997). Słownik Kubusia Puchatka. Poznań: Dom Wydawniczy Rebis.

Milne, A.A. (1995). Kubuś Puchatek. Chatka Puchatka. Warszawa: Nasza Księgarnia.

Rockart,, J.F. (1979). Chief Executives define Their Own Data Needs. Harvard Business Review (Sept.-Oct., 1979).

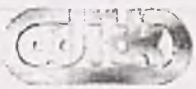
Rockart,, J.F. (1982). The Changing Role of the Information Systems Executive. Sloan Management Review (Fall, 1982).

Michał Hałas

Ośrodek Doradztwa i Treningu Kierowniczego
Gdańsk

Łańcuch Krytyczny

Ośrodek Doradztwa i Treningu Kierowniczego



Łańcuch Krytyczny

(Critical Chain)

Moduł oparty na pomyśle Eliyahu Goldratt'a

9 - 1

Ośrodek Doradztwa i Treningu Kierowniczego

Łańcuch Krytyczny

(Critical Chain)

Moduł oparty na pomyśle Eliyahu Goldratt'a

9 - 1

Czas zabija budżet

- Rozwój produktów - odpadnięcie z wyścigu, miejsce za peletonem konkurencji.
- Zaburzenia w płynności
- Zwiększenie budżetu o „współczynnik spalania"
- Rozkładanie budżetu brakiem terminowości (kary umowne)
- „Pociąg z popytem" odjechał

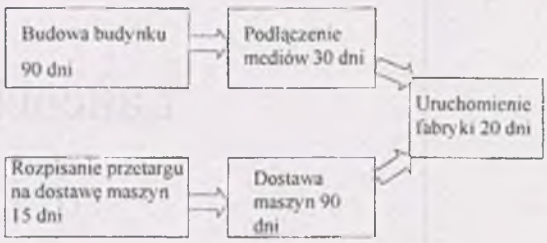
9 - 2

Skupienie uwagi kierownictwa

- Koncentrowanie się na wszystkim to to samo co koncentrowanie się na niczym

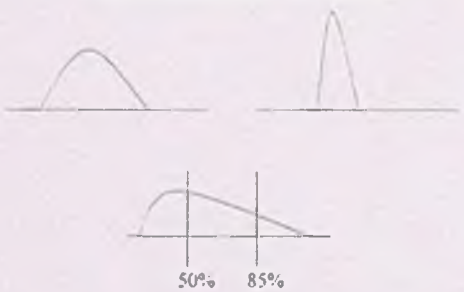
9 - 3

Przykład z budowy



9 - 4

Rozkład prawdopodobieństwa zakończenia zadania



9 - 5

„Zapasy" w zadaniach

- Estymowanie na 80-90% prawdopodobieństwa.
- Estymacja jest odzwierciedleniem przekroczenia na poprzednim projekcie.
- Każdy szczebel dodaje swoje zapasy 5+5=13.
- Zabezpieczanie się przed odgórnymi cięciami (dodatkowe 20%)

9 - 6

Przyczyny wewnętrznych zapasów

- Poczucie bezpieczeństwa
- Doświadczenia porażek
- Wygoda
- Oprócz zaangażowanych osób nikt nie wie ile coś ma naprawdę trwać
-
-

9 - 7

Moc zbliżającego się terminu zakończenia

Deadline - (ang. gra słów) linia po przekroczeniu której Menedżer Przedsięwzięcia jest martwy

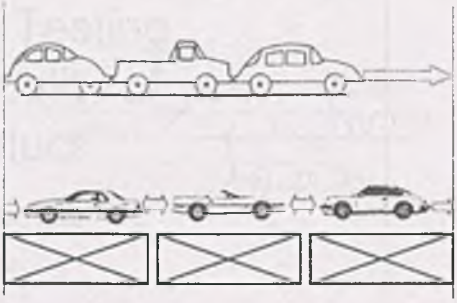
9 - 8

Skutki napompowanych zadań

- Brak motywacji do wczesnego ukończenia (socjalizm: następnym razem obetniemy twoją estymację)
- Ewentualne przyspieszenie nie przyspieszy następnych zadań - „kalendarzowa gotowość"
- Przy równoległych działaniach opóźnienia są przekazywane dalej. Przyspieszenia nie.
- Syndrom studenta
- Wielozadaniowość (multitasking)

9 - 9

Parkowanie po Parysku



9 - 10

Konsekwencje

- Przejadanie i marnotrawienie czasu
- Przekraczanie i tak napompowanego harmonogramu
- Dobre wyniki przez 90% czasu. Pozostałe 10% zadania trwa tyle co uprzednie 90%.

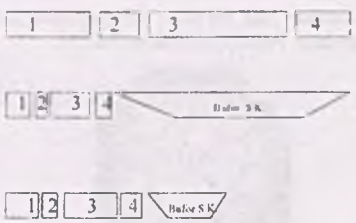
9 - 11

Propozycje Krytycznego Łańcucha (1 z 2)

- Przywrócić wagę ścieżce krytycznej
- Dobrowolnie obciąć estymacje o n.p. połowę
- Uzyskany zapas umieścić jako bufor na końcu ścieżki krytycznej
- Priorytet śledzenia - bufory

9 - 12

Skracanie Ścieżki



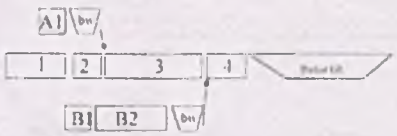
9 - 13

Propozycje Krytycznego Łańcucha (2 z 2)

- Codzienne raportowanie estymowanych zakończeń.
- „Parkowanie po Parysku"
- Sygnalizowanie wykonawcy nadchodzącego rozpoczęcia 7, 3 i 1 dni przed rozpoczęciem
- Dojścia do ścieżki krytycznej zabezpieczyć buforami.

9 - 14

Jeszcze BW - bufory wejściowe



9 - 15

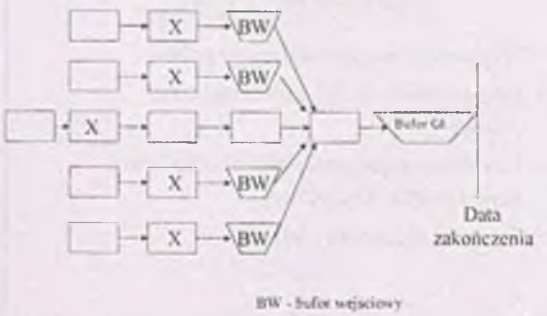
„Wąskie gardło"

W przypadku „wąskiego gardła" - działu, człowieka, maszyny:

- ustawić arbitralnie kolejkę zadań przed wąskim gardłem.
- Przeredagować sieci projektów uznając kolejność przed wąskim gardłem za priorytet.
- Dalej śledzić bufory zapasów (jest to odbicie sumy ryzyka).

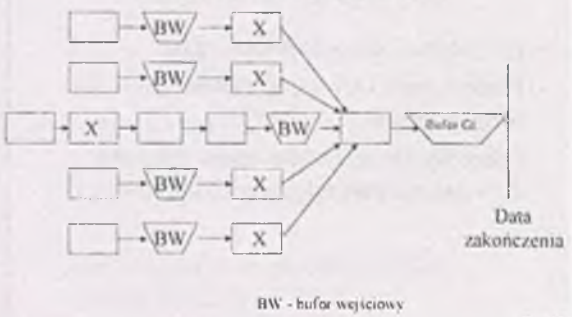
9 - 16

Wąskie gardła (jako X)



9 - 17

Łańcuch Krytyczny

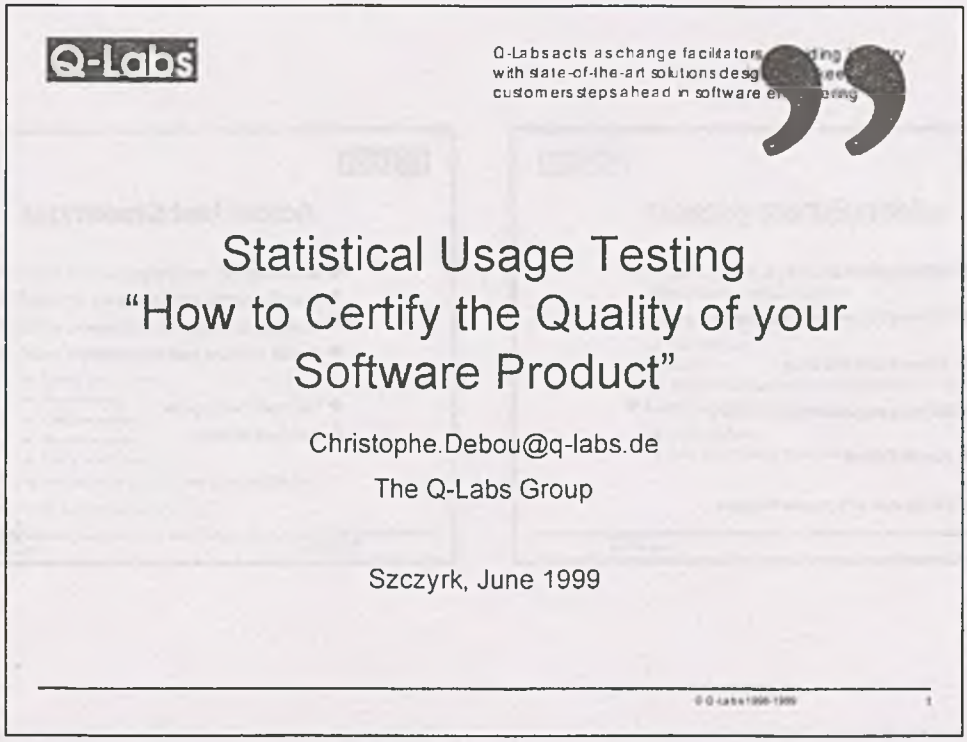


9 - 18

Christophe Debou

Q-Labs

Statistical Usage Testing „How to Certify the Quality of your Software Product"



Q-Labs

The Q-Labs Company

- Global Company with local presence in
  - Sweden (Lund, Linköping Goteborg)
  - Germany (Kaiserslautern)
  - USA, (College Park, Knoxville)
  - Ireland, Poland, Netherlands
- 70+ Consultants
- Highly qualified staff many with long term industrial background
- ISO9001 certified since 1993

Q-Labs

Q-Labs Company Profile

- Established 1989
  - Continuous growth , internationalization and local presence
- Strategic Alliances & Network
  - Collaboration with world leading industrial organizations and research institutes
- Advisory board
  - Prof. Rombach, prof. Basili, Prof. Fenton
- Working with the leading organisations
  - ABB, Alcatel, Ericsson, Bosch, Siemens, Volvo, IBM, Telia, Telenor, ...

Q-Labs

Main Service Areas

- Software Management
- Software Process Improvement
- Software Best Practices
- Software Acquisition Improvement
- Human Factors
- Certification of Software Products

Q-Labs

Actual Test Challenges

- Not testing the 'right' thing
- Seem to test the same things
- Quality goes down in each phase
- Cannot measure product 'reliability'

- Too many field reports
- Low Field reliability

Q-Labs

The Reality of the Sample

- Exhaustive testing is not practical due to variable inputs and infinite languages
- ALL SOFTWARE TESTING IS SAMPLING
- How to choose the best sample?

Q-Labs

The Implications of Sampling

- Don't know if the software contains no faults
- Don't know if software will never fail
- Two kinds of uses
  - Those executed during test
  - Those left for the user to execute
- How to predict what the user should expect to see?

Q-Labs

How We Normally Test

- We believe that the mental process for testing is largely the same for all people/organizations
- Understanding the Problem
- Planning the Test
- Building the Test Cases
- Running Tests
- Looking at Test Results

Q-Labs

Understanding the Problem

- Read through existing documentation
- Work through existing prototypes
- Talk with domain experts
- Build a mental model of the system
  - What it does
  - How it may do it
  - How one should think about testing it
- Document portions of the mental model as needed

Q-Labs

Planning the Test

- Define boundaries for test
- Gather up inputs, etc.
- Create mental model of system usage
- Define what will be tested
  - Normal cases (usage)
  - Requirements
  - Critical situations
  - Boundary cases
  - 'Likely to fail' cases
- Determine how much testing in each area
- Avoid looking at code

Q-Labs

Building the Test Cases

- Create a test case one at a time, consistent with mental model/documentation
- Reuse, as useful, from other tests
  - Test structure
  - Test steps
  - Right combination of reuse and new tests
- Compute expected results
  - Painful activity
  - Look for facilitating the work (easy data, etc.)

Q-Labs

Running the Test

- Execute tests as allowed by environment
- Determine if test passed/failed
- Record test results

Q-Labs

Looking at Test Results

- Organize results based on measures in organization
  - Failures per KLOC
  - Open TRs vs. Total TRs
- Make release decision based upon schedule and testing information

Q-Labs

How does SUT Relate?

- Approach is similar to the 'typical' testing approach
- However, we leverage performance by making approach more structured and formal
  - Focus is on building models and then generating test cases
  - Generate a sample of tests that allows us to draw a stronger conclusion
  - Leverage the statistical attributes of the model
  - Focus more heavily on usage, maximizing user perceived reliability

Q-Labs

Why SUT?

- To maximize field reliability
- To maximize the potential for objective measurement
  - Measurement for test planning
  - Measurement for test execution
  - Measurement for product certification
  - Measurement for process assessment
- To remove human bias from testing
- To test consistently with field usage
- To make software quality assessment more scientific

Q-Labs

Statistical Usage Testing

- Test the software the way users use it
  - Develop test without code knowledge
  - Provide maximum field reliability gain
  - Provide the correct focus for quality measurement: the user
- Testing as a statistical experiment
  - Statistical inference provides a scientific assessment for tests not executed
  - Testing decisions based on objective data

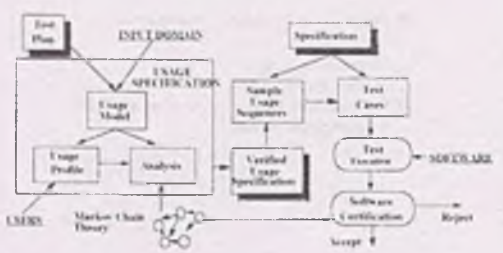
Q-Labs

SUT Process

Test Planning	Define Test Goals/Strata
	Define Boundary and Stimuli
	Define a Use
	Allocate Test Budget to Goals
	Allocate Test Approaches to Goals
Usage Model Development	Build Model Structure
	Assign Probabilities
	Analyze Model
Generate Tests	Generate Scripts
	Process into Test Cases
	Study Coverage
	Compute Expected Results
Test Execution	Run Tests
	Determine Success/Failure
	Record Pass/Fail
Certification	Compute Quality Measures
	Compute Stoppage Criteria
	Make Release Decisions

Q-Labs

Certification Process Overview



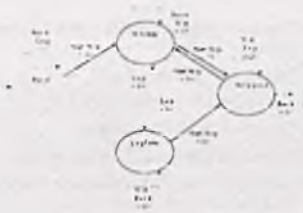
Q-Labs

Markov Chain

- Usage Model has states, transition arcs and transition probabilities
- The states must obey the Markov property, which means that the information necessary to determine a transition to a future state must be wholly contained in the current state or future transitions are independent of past transitions given the current state
- State definition based upon change in possibility/probability of next state/transition arcs

Q-Labs

Example Usage Model



Q-Labs

Evaluation Measures

- Quality
  - Reliability and Confidence
  - Mean Time to Failure
- Stoppage Criteria
  - Actual State and Arc Coverage
  - Actual States and Arcs not Covered
  - Discriminant
  - Distance

Q-Labs

Tool Support

- tool SET_Certify - a CASE tool to support SUT
- Automates all computational steps - test planning/model analysis, test generation, product certification
- Automatic generation of tests - random samples and coverage test
- Full C API - Allows integration of SUT into an automated testing environment

Q-Labs

Project Experiences

- Primarily Embedded Systems
- Domains include Telecommunications, Defense, Aerospace, Medical Devices, CASE Tools
- Three Example Projects
  - Alcatel
  - FAA
  - IBM SSD

Q-Labs

Alcatel

- Evaluate feasibility of Usage Specification and Functional Test
- Modeled three ISDN switch features
- Results
  - 30% more effective (defects per test case)
  - Large number of requirements issues noted and clarified early
  - Usage modeling more interesting than creating test specifications
  - Shift in thought process for testing
  - Reduced lead time due to earlier defect detection
- HTTP //www alcatel de/projekte/usst

Q-Labs

FAA

- Traffic Situational Display
- Numerous and major requirements issues observed and noted
- At least 7 major models (>350 states each), composed of at least 50 minor models - used Model Composition prototype
- Stimulus level models
- Modeled full system, organized by functional groups and incremental deliveries
- Results - Unbiased test Test to usage, Large numbers of failures found

Q-Labs

IBM-SSD Tucson

- Mass Storage Solutions - Devices, controllers
- Using approach since 1990, only way they test
- Many products released, usage and protocol models
- Correctness and Performance
- High degree of automation - full process + >>90% of tests evaluated automatically
- Release criteria 1000 consecutive passes in final version with representative sample
- Ideas being moved across division

Q-Labs

Applying the Techniques

- Concepts readily applicable with 3-3.5 days of training
- Tool handles mathematics
- One approach - model Features
- Can model components/objects
- Can model from a user perspective
- Anything that can be bounded and viewed as a black box can be tested using SUT

Q-Labs

Conclusion

- All testing is sampling - how do we find the best sample
- Usage models allow a system's to be characterized
- Focus is on usage model development - Tests come for free
- Stronger quantitative basis for release decision

Q-Labs

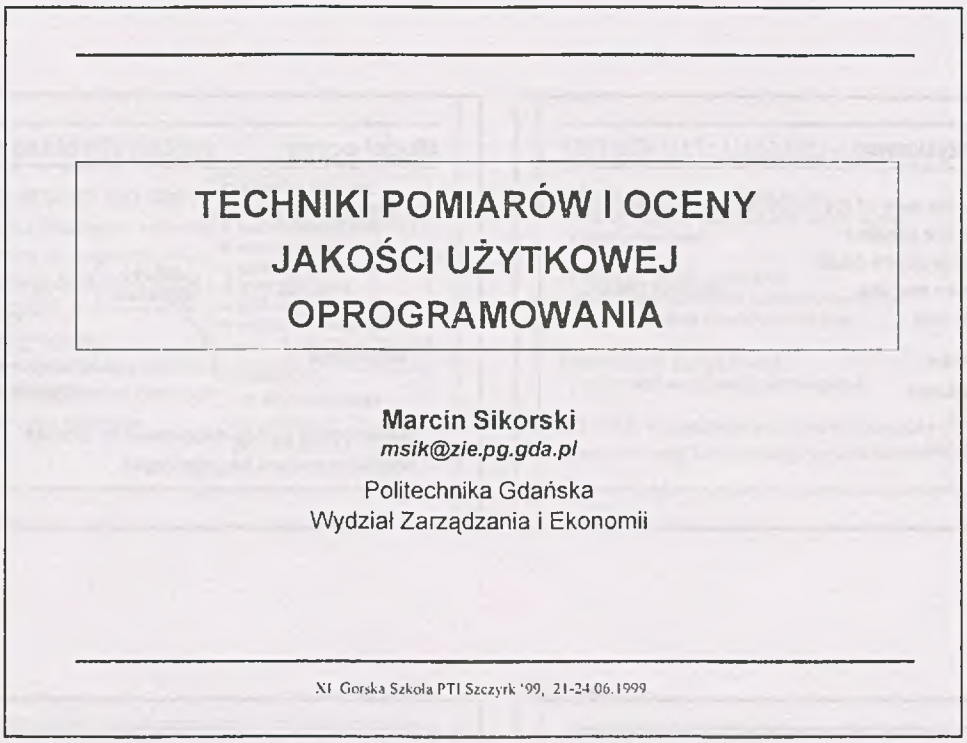
References

- H.D. Mills, M. Dyer and R.C. Linger, Cleanroom software engineering, IEEE Software. Sept 87
- J.A. Whittaker, M. G. Thomason. A Markov Chain Model for Statistical Usage Testing, IEEE Trans. Softw. Eng. 20, 10 (Oct 1994), pp. 812-824
- S. W. Sherer, A. Kouchakdjan, P. G. Arnold, Experience using Cleanroom Software Engineering, IEEE Software, may 1996
- A.Beck, H. Beer. ESSI Process Improvement Experiment 23843 - USST - Usage Specification and Statistical Testing, Proceedings of the Eurostar98, Munich

Marcin Sikorski

Politechnika Gdańska
Wydział Zarządzania i Ekonomii
msik@zie.pg.gda.pl

Techniki pomiarów i oceny jakości użytkowej oprogramowania



TECHNIKI POMIARÓW I OCENY JAKOŚCI UŻYTKOWEJ OPROGRAMOWANIA

Marcin Sikorski
msik@zie.pg.gda.pl
Politechnika Gdańska
Wydział Zarządzania i Ekonomii

Jakość produktu informatycznego

Jakość konstrukcyjna
- metryki - atrybuty kodu (ISO 9126)
- złożoność, ilość pętli, długość kodu, styl i struktura

⇐ ocena kodu źródłowego

Jakość ergonomiczna
- charakterystyki i wymagania ergonomiczne (ISO 9421)
- łatwość nauki i obsługi, intuicyjność, „przyjazność"

⇐ ocena interfejsu użytkownika

Jakość użytkowa
- oceny użytkowników i ekspertów
- relacja efekt/wysiłek, skuteczność wspomagania zadań roboczych

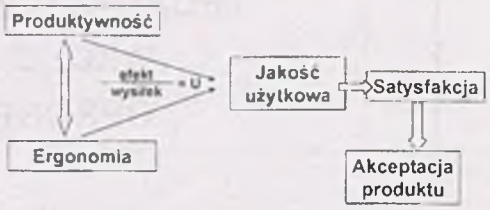
⇐ ocena produktu

Jakość użytkowa

Usability - the sum of the product characteristics which make the product:
- *productive in user's tasks*
- *easy to learn and use* *(Nielsen,1993)*
- *pleasant in use*

Terminy polskie:
- jakość użytkowa
- użyteczność
- użytkowalność

Model oceny



- zakres oceny należy dopasować do potrzeb
- oceniamy system lub jego część

Cel i zakres oceny

Ocenie podlega:
- prototyp produktu
- makieta interfejsu użytkownika
- gotowy produkt lub jego część
- dokumentacja i/lub wsparcie

Ocenę przeprowadza się :
- w laboratorium
- w firmie u użytkownika

Ocena:
- porównawcza
- diagnostyczna

Oceniamy: jakość użytkową (akceptowalność produktu)

Rodzaje oceny

- ocena na zgodność z zasadami, przepisami, specyfikacjami
- ocena akceptowalności produktu
- ocena porównawcza

Co oceniamy: produkt, prototyp czy interfejs ?

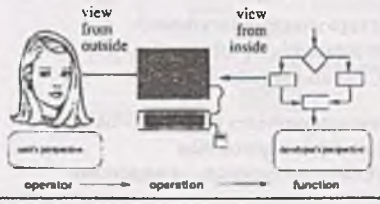
Ocena:	diagnostyczna	porównawcza
rozwojowa	...	...
podsumowująca	...	...

Ocena: ekspertów lub użytkowników

Interfejs użytkownika a jakość produktu

Interfejs użytkownika :

- ma duży wpływ na odbiór jakości użytkowej produktu
- może ułatwić lub utrudnić korzystanie z produktu
- kształtuje emocjonalny stosunek użytkownika do produktu
- buduje lub niszczy lojalność użytkownika



Kryteria oceny wg ISO9241

Norma ISO9241: Ergonomic Requirements for Office Work with Visual Display Terminals

7 kryteriów głównych:

- dopasowanie do zadania
- samo-opisywalność
- kontrola przez użytkownika
- zgodność z oczekiwaniami
- tolerancja na błędy
- możliwość indywidualizacji
- wspomaganie nauki

Poziomy oceny:

- ekran
- dialog
- interfejs użytkownika
- produkt

Wymagania dla dialogu

CZĘŚCI NORMY ISO 9241:

10 Dialogue principles
11 Guidance on usability
12 Presentation of information
13 User guidance
14 Menu dialogues
15 Command language dialogues
16 Direct manipulation dialogue
17 Form-filling dialogue

DIALOGUE DESIGN:

- suitability for the task
- self-decriptiveness
- controllability
- conform expectations
- error tolerance
- individuallzation
- suitability for learning

USABILITY:

- effectiveness
- efficiency
- satisfaction

ISO 9241/11: Usability components

Effectiveness (skuteczność):
- accuracy and correctness with which users achieve their goals

Efficiency (efektywność):
- resources expended in relation to accuracy and completness in reaching the goal

Satisfaction (satysfakcja):
- comfort and acceptability of the system

ISO 9241: Podstawowa norma dotycząca jakości ergonomicznej i użytkowej produktu informatycznego

Wybrane składniki jakości użytkowej

- Skuteczność-produktywność produktu w zadaniu
- Występowanie błędów podczas obsługi produktu
- Zdolność użytkownika do korekty skutków błędów
- Wysiłek potrzebny do nauczenia się obsługi systemu
- Szybkość utraty nabytych umiejętności
- Możliwość dopasowania systemu do nowych potrzeb
- Łatwość włączenia systemu do istniejących procedur
- Całkowite koszty użytkowania systemu
- Zadowolenie użytkowników (udziałowców) z systemu

Składniki te dotyczą większości typowych systemów

Ocena ekspertów (1/2)

Ocena ekspertów (*usability inspectors*):

- inspekcja funkcji produktu
- ocena przy pomocy listy kontrolnej
- ocena heurystyczna
- test zadaniowy

- Ocena zgodności z dokumentami
- Ocena zgodności z zasadami
- Ocena interfejsu
- Ocena podsumowujaca (kwalifikacyjna)

Ocena ekspertów (heurystyczna) (2/2)

heurystyki: ogólne zasady postępowania
źródło: Nielsen J. (1993): Usability Engineering, Academic Press, New York. *http://www.useit.com*

- stwórz prosty i naturalny dialog
- używaj języka użytkownika
- nie obciążaj pamięci użytkownika
- zachowaj spójność dialogu
- zapewnij potwierdzenia działań
- zapewnij dobrze oznakowane wyjścia
- zapewnij możliwość pracy na skróty
- zapewnij dobre komunikaty o błędach
- zapobiegaj powstawaniu błędów

13

Ocena użytkowników

1. Ocena laboratoryjna:
   - test produktu: pomiary, chronometraż, rejestracja video
   - wywiad: indywidualny lub grupowy
   - ankieta, kwestionariusz
2. Ocena eksploatacyjna (użytkowa):
   - wywiad: indywidualny lub grupowy
   - ankieta, kwestionariusz

- Ocena akceptowalności całego produktu
- Ocena interfejsu użytkownika
- Ocena rozwojowa produktu ze wsparciem

14

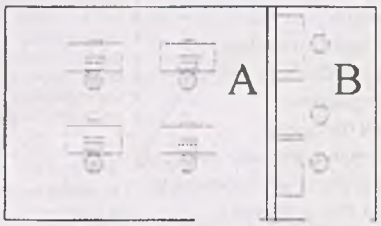
Techniki rejestracji pracy użytkownika

- obserwacja
- rejestracja na video
- rejestracja *logfile*
- chronometraz czynnosci
- protokoły pomiarowe
- rejestracja audio

Rejestrujemy:
- czasy wykonywania czynności
- popełnione przez użytkownika błędy
- inne istotne zdarzenia

15

Laboratorium



stanowiska testowe aparatura

16

Uczestnicy badania

Reprezentatywność:
- liczebność grupy
- dobór składu grupy
- homogeniczność grupy
- doświadczenie uczestników

Problemy:
- przygotowanie uczestników
- pozyskanie uczestników
- gratyfikacje dla uczestników
- organizacja sesji testowej

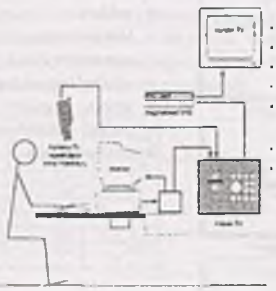
17

Przebieg sesji testowej

Plan testu:
1. Wprowadzenie
2. Wywiad wstępny
3. Zadania próbne
4. Zadania testowe
5. Ankieta o produkcie
6. Wywiad grupowy
7. Zakończenie

18

Budowa stanowiska rejestracyjnego



- system 2 kamer
- rejestracja na VHS
- zegar-timer LCD
- chronometraż
- protokoły pomiarowe
- rejestracja 1-2 osób
- czas nagrania do 30 min

19

Problem wystąpił jeśli:

- użytkownik określił cel działania, ale mimo wysiłków nie posuwa się w jego kierunku (wskutek czego nie będzie w stanie osiągnąć go w zadanym czasie (konieczna jest pomoc prowadzącego)
- użytkownik określił cel działania i musi wypróbować kilku podejść aby znaleźć rozwiązanie, gdyż system nie sugeruje jak zacząć ani nie podpowiada kolejnych kroków
- użytkownik określił cel działania, bezskutecznie próbuje kilku podejść, po czym rezygnuje z wykonywania zadania
- uczestnik nie ukończył zadania, gdyż wynik zadania jest inny niż wymagany
- uczestnik nie ukończył zadania, gdyż nie zmieścił się w zadanym czasie
- uczestnik wyraża zdziwienie zachowaniem się systemu
- uczestnik wyraża negatywną opinię lub stwierdza że coś stanowi problem
- uczestnik zgłasza propozycję zmian w oceniany programie

20

Wywiady grupowe (*focus group*)

- wywiad moderowany, 6-9 osób
- ujawnia potrzeby i oczekiwania ukryte, motywacje i nastawienia
- odkrywa istotne informacje rynkowe
- umożliwia porównanie doświadczeń
- odczucia, emocje, co się nie podobało
- daje tylko informacje jakościowe
- przebieg wywiadu nagrywać na kasetę audio

Technika bardzo skuteczna, ale wymaga moderatora

21

Wywiady indywidualne

- dla istniejącego, pracującego systemu
- cel wywiadu - konieczny scenariusz i „protokół"
- informacje „z pierwszej ręki", po testowaniu
- odczucia, emocje, co się nie podobało
- najlepiej prowadzić w miejscu pracy użytkownika
- trudności w opracowaniu wyników
- duży błąd w szacowaniu liczb

Tylko dla małej ilości użytkowników

22

Budowa ankiety

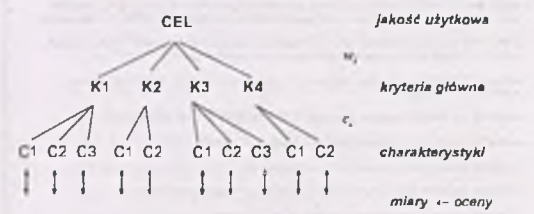
- ankieta odzwierciedla model jakości:
  - kryteria główne
  - podkryteria (charakterystyki)
  - miary
- części ankiety, pytania zamknięte i otwarte
- problem wyboru skali
  - nominalna < porządkowa < przedziałowa < ilorazowa
  - jakościowa | ilościowa
- problem wiarygodności odpowiedzi
- problem trafności i rzetelności narzędzia
- problem rozrzutu wyników

23

Przykład ankiety (fragment)

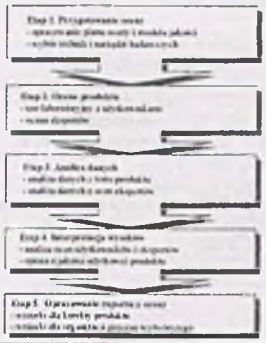
24

Drzewo jakości użytkowej produktu



Obliczanie wyników: $U = \frac{\sum_{i=1}^{n} K_i \cdot w_i}{\sum_{i=1}^{n} w_i}$

Ogólna metodyka oceny



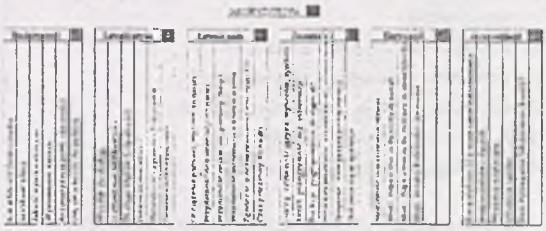
Techniki:
- ankieta
- test produktu
- rejestracja video
- wywiad indywidualny
- wywiad grupowy
- analiza hierarchiczna
- listy kontrolne

Przykład oceny oprogramowania z udziałem użytkowników

PRODUKT: Program finansowo-księgowy dla małej firmy
UCZESTNICY: Księgowe z małych przedsiębiorstw

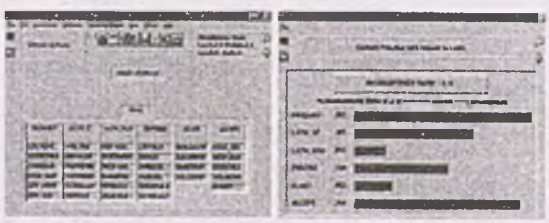
(1) Model jakości

Struktura modelu jakości:
Cel - Kryteria - Charakterystyki - Miary



(2) Określenie preferencji

- określenie wzajemnej dominacji kryteriów
- macierz porównań parami dla kryteriów głównych
- program *Expert Choice*: obliczenie wag istotności dla kryteriów



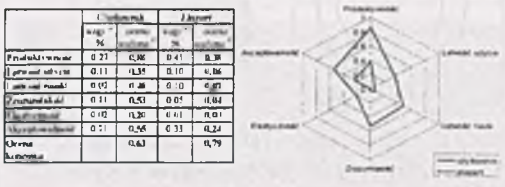
(3) Test produktu

Plan testu:
1. Wprowadzenie
2. Kalibracja modelu jakości
3. Zadania próbne
4. Zadania testowe
5. Ankieta o produkcie
6. Wywiad grupowy

Analiza danych:
1. Selekcja danych
2. Agregacja danych
3. Obliczenia
4. Interpretacja wyników

(4) Wyniki oceny

- pomiary, noty, oceny, opinie: normalizacja danych
- przeniesienie ocen na model jakości
- agregacja ocen cząstkowych, obliczenie oceny syntetycznej
- wizualizacja wyników dla projektantów



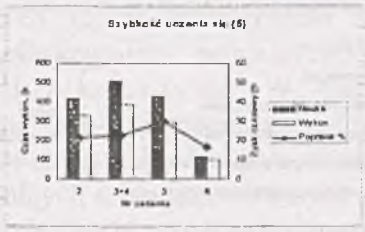
31

(5) Interpretacja wyników

→ wskazanie obszarów (charakterystyk) do poprawy

32

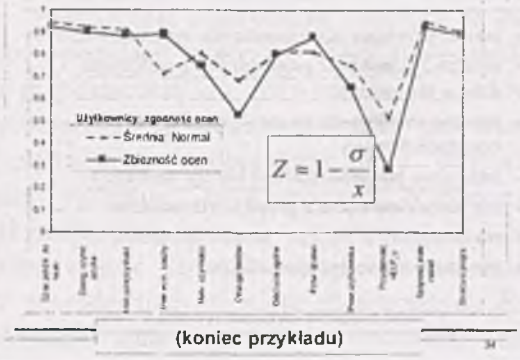
Wyniki uzupełniające - nauka



- zgodność przebiegów dla różnych osób
- zadawalająca wartość poprawy czasu ukończenia
- klasyfikacja i analiza popełnionych błędów

33

Analiza zbieżności ocen



(koniec przykładu)

34

Wnioski z oceny produktu

Zalecenia dla ulepszenia produktu:

- interfejs użytkownika, dialog, ekran
- produktywność, funkcje, elastyczność, bezpieczeństwo, ...

Zalecenia dla ulepszenia procesu wytwórczego:

- zbieranie danych od użytkowników
- regularne testowanie produktów z użytkownikami
- powiązanie satysfakcji użytkownika i jakości użytkowej z programami projakościowymi
- jakość użytkowa a dojrzałość procesu

35

Podsumowanie

Oceniamy:

- jakość całego produktu
- poziom akceptacji produktu
- odbiór jakości produktu przez użytkowników

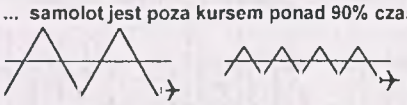
Wykorzystanie informacji od użytkowników:

- wykrycie wad użytkowych produktu
- atrakcyjność produktu, porównanie z konkurencją
- mechanizm decyzji wyboru produktu dla firmy
- integracja ze środowiskiem biznesowym
- reklama, marketing, dystrybucja, cena

36

Możesz wierzyć albo nie...

- ... samolot jest poza kursem ponad 90% czasu



- skąd wiesz czy Twój projekt idzie w dobrą stronę?

- trzeba często mierzyć kurs i korygować kierunek
- im częstszy pomiar, tym lepsze wyniki
- więcej iteracji - krótszy czas projektu, mniej błędów

37

Testowanie prototypów to podstawa

- Poprzednia wersja jest prototypem aktualnej wersji
- Aktualna wersja jest prototypem dla przyszłej wersji

- Każdy prototyp przechodzi przez te same etapy
- Potrzebujemy danych: szybciej, więcej, dokładniej
- Najlepsze dane uzyskamy od użytkowników

38

To nieprawda, kiedy słyszysz że ...

- jakość interfejsu użytkownika nie ma znaczenia
- wystarczy zapoznać projektantów z metodami a będą je stosowali
- świetne rozwiązania powstają tylko w dużych, bogatych firmach
- potrzebne jest laboratorium i droga aparatura
- nie poradzimy sobie z grupą użytkowników
- nie potrafimy „zmierzyć" jakości użytkowej
- nie mamy czasu ani specjalistów

39

Zapewnienie jakości użytkowej w firmie

Elementy systemu :

- zbieranie wymagań
- budowa i testowanie prototypów
- zbieranie danych z użytkowania
- przekazywanie danych projektantom

} user feedback

- nadzór nad stosowaniem przewodnika stylu

Kto ma to robić?

Job title: *Usability engineer*

40

Co jest potrzebne:

- określenie celów jakości użytkowej
- poparcie kierownictwa firmy
  - *usability* w misji firmy i w polityce jakości
  - integracja z reklamą i sprzedażą
  - przydzielenie zasobów
- wybór metod i technik do zastosowania
- przeprowadzenie testów, badań i ocen
- wykazanie efektów badań i testów produktu

User feedback w procesie wytwórczym

41

Zakończenie:

Pokaz nagrania video z testu produktu

- rejestracja pracy użytkownika (1-2 osoby)
- rejestracja zmian ekranu

- chronometraż wykonania zadań
- analiza popełnianych błędów
- analiza retrospektywna z uczestnikami
- zadanie próbne i zadanie testowe
- test produktu, wywiad i dyskusja
- wyniki liczbowe i wnioski projektowe

Metoda pracochłonna ale bardzo skuteczna

42

Piotr Jędrzejowicz

Katedra Informatyki,
Wyższa Szkoła Morska, Gdynia
pj@wsm.gdynia.pl

Rozwiązywanie problemów za pomocą algorytmów opartych na ewolucji populacji

1. Wstęp

Algorytmy oparte na ewolucji populacji należą do klasy technik optymalizacji, w których sposób poszukiwania rozwiązań jest naśladowaniem naturalnych procesów ewolucji i uczenia się. W algorytmach, o których mowa przedmiotem przetwarzania jest populacja tak zwanych osobników – potencjalnych rozwiązań lub innych obiektów, z których łatwo wygenerować rozwiązania danego problemu. Wykonywanie algorytmów opartych na ewolucji populacji nazywane jest mianem obliczeń ewolucyjnych (ang.: *evolutionary computing*). Mechanizmy ewolucji wykorzystywane w trakcie obliczeń ewolucyjnych polegają na wymianie informacji pomiędzy osobnikami oraz na uczeniu się osobników. Okresy współpracy (wymiany informacji) przeplatane są okresami samoadaptacji (uczenia się). Algorytmy oparte na ewolucji populacji posiadają, zazwyczaj, charakter iteracyjny i probabilistyczny zaś proces ewolucji populacji osobników odbywa się zgodnie z pewnymi, precyzyjnie zdefiniowanymi regułami.

Do szeroko znanych technik wykorzystujących koncepcję ewolucji populacji należą, między innymi, algorytmy genetyczne [Holland 1975], programowanie genetyczne [Koza 1992], w którym każde rozwiązanie jest programem komputerowym, czy też programowanie ewolucyjne [Michalewicz 1992]. Na ewolucji populacji mogą być także oparte systemy klasyfikujące [Goldberg 1989] oraz systemy sztucznego życia [Munakata 1998].

Popularność algorytmów opartych na ewolucji populacji wiąże się z ich skutecznością i przydatnością do rozwiązywania trudnych problemów, a w szczególności problemów optymalizacji kombinatorycznej o wykładniczej złożoności obliczeniowej. Wspomniana skuteczność, łatwość implementacji oraz elastyczność samego podejścia stosowanego z powodzeniem w wielu różnorodnych dziedzinach sprawiły, że w ostatnich latach pojawiły się dziesiątki (jeśli nie setki) technik, podejść i algorytmów, których autorzy proponują rozmaite reguły ewolucji populacji. W niniejszej pracy przedstawiono cechy charakterystyczne algorytmów opartych na ewolucji populacji, a na ich tle omówiono niektóre ważniejsze algorytmy i techniki.

2. Cechy charakterystyczne algorytmów opartych na ewolucji populacji

Do istotnych cech algorytmów omawianej klasy zalicza się sposób definiowania osobników, reguły procesu ewolucji, strukturę sąsiedztwa osobnika, źródła informacji, strategię intensyfikacji, strategię dywersyfikacji oraz sposób postępowania z osobnikami reprezentującymi rozwiązania niedopuszczalne [Hertz, Kobler 1998].

Osobnikiem w obliczeniach ewolucyjnych może być odpowiednio zakodowane rozwiązanie problemu (np. ciąg bitów lub wektor liczb lub łańcuch znaków reprezentujących wartość rozwiązania). Osobnikiem może być także element rozwiązania bądź zbiór rozwiązań. Osobnikiem, wreszcie, może być każdy dowolny obiekt pod warunkiem, że istnieje procedura, która pozwala na transformację tego obiektu na rozwiązanie rozważanego problemu. Zbiór osobników na danym etapie rozwiązywania problemu nazywany jest populacją bądź generacją osobników. Dla każdego osobnika w populacji daje się obliczyć wskaźnik jakości, którym zazwyczaj jest wartość reprezentowanego rozwiązania.

Dla przykładu, w problemie podróżującego komiwojażera poszukiwana jest trasa, która pozwoli komiwojażerowi wyruszyć z macierzystego miasta, odwiedzić jeden raz każde z odwiedzanych miast i powrócić do punktu wyjścia minimalizując koszt podróży. Niech miastem macierzystym jest Warszawa, a odwiedzanymi miastami Gdańsk, Szczecin, Poznań, Wrocław, Kraków, Rzeszów i Lublin. Rozwiązaniem problemu jest permutacja odwiedzanych miast, którą można przedstawić jako łańcuch znaków. Osobnikiem w odpowiednim algorytmie opartym na ewolucji populacji może być łańcuch znaków zbudowany z pierwszych liter odwiedzanych miast. I tak osobnik (G,S,P,W,K,R,L) reprezentuje rozwiązanie, w którym komiwojażer wyruszy z Warszawy i odwiedzi, po kolei, Gdańsk, Szczecin, Poznań, Wrocław, Kraków, Rzeszów i Lublin, a następnie powróci do Warszawy. Wskaźnikiem jakości osobnika jest, w rozważanym przypadku, łączny koszt podróży. Z kolei, w problemie szeregowania zbioru n niezależnych i niepodzielnych zadań o dowolnych czasach gotowości i przetwarzania na wielu identycznych procesorach celem minimalizacji czasu uszeregowania, osobnikiem może być wektor n numerów zadań, który jest transformowany na rozwiązanie problemu drogą zastosowania algorytmu listowego. Niech $n = 5$ i niech przykładowy osobnik jest wektorem (4,5,1,3,2). Rozwiązanie problemu generowane jest wówczas drogą przydzielenia zadania nr 4 do najmniej obciążonego procesora, potem zadania nr 5 - również do najmniej obciążonego procesora, potem zadania nr 1 itd., aż do wyczerpania wszystkich zadań z listy. Wskaźnikiem jakości osobnika może być moment zakończenia przetwarzania tego zadania ze zbioru zadań, które zostało zakończone najpóźniej.

W obliczeniach ewolucyjnych populacja osobników podlega ewolucji zgodnie z określonymi regułami. W każdej iteracji powstają nowi osobnicy. Strategia ewolucji wymaga określenia którzy spośród nich przejdą do nowej generacji, a którzy zostaną odrzuceni. Często obliczenia ewolucyjne opierają się na populacji o stałej wielkości m. W takim przypadku do kolejnej iteracji przechodzi m najlepszych osobników tworząc nową generację. Niektóre strategie ewolucji przewidują, że z jednej iteracji do kolejnej przechodzą wyłącznie nowi osobnicy. Strategia taka nazywana jest wymianą generacji. Z kolei strategia, w której przy przejściu do kolejnej iteracji tylko określona część osobników może być zastąpiona nowymi, jest nazywana ewolucją o stałym stanie.

Niektóre strategie ewolucji dopuszczają zmienną wielkość populacji w trakcie poszukiwania rozwiązania. Decyzja o zakwalifikowaniu (bądź nie) osobnika do nowej generacji może być oparta na mechanizmie losowym, choć prawdopodobieństwo takiej kwalifikacji może zależeć od wskaźnika jakości danego osobnika.

Populacja osobników, w której nie przewidziano żadnych ograniczeń na możliwości wzajemnej wymiany informacji pomiędzy osobnikami jest nazywana populacją z nieokreśloną funkcją sąsiedztwa. Jeśli natomiast dla każdego osobnika określony został pewien podzbiór osobników, z którymi możliwa jest wymiana informacji, wówczas mówi się o populacji ze zdefiniowaną funkcją sąsiedztwa.

Jeśli nawet osobnik posiada duży zbiór sąsiadów, z którymi wolno mu się komunikować to wcale nie znaczy, że w danej iteracji wykorzystane zostaną wszystkie możliwe źródła informacji. Najczęściej, wymiana informacji następuje pomiędzy dwoma losowo dobranymi osobnikami - rodzicami, którzy poprzez taką wymianę generują jednego nowego osobnika - potomka. Zdarzają się jednak rozwiązania, w których w tworzeniu potomka

uczestniczy więcej niż dwóch rodziców, bądź takie w których jeden osobnik jest źródłem informacji wystarczającej do stworzenia potomka. Wracając do przykładowego problemu podróżującego komiwojażera, niech (G,S,P,W, K,R,L) oraz (S,G,W,P,K,R,L) są dwoma osobnikami. Krzyżowanie osobników pozwala wygenerować potomka. Dla losowo wybranego punktu krzyżowania $p = 2$, potomkiem obydwu osobników jest nowy osobnik (G,S, W,P,K,R,L).

Zaobserwowano, że wielu przypadkach obliczenia ewolucyjne można w istotny sposób usprawnić jeśli uda się zastosować tak zwaną strategię intensyfikacji w stosunku do poszczególnych osobników tworzących populację (por. np. [Fleurent, Ferland 1996]). Strategia taka polega na zastosowaniu algorytmu poprawy, którego zadaniem jest polepszenie wskaźnika jakości osobnika bez korzystania z informacji pochodzącej od innych osobników. Większość algorytmów poprawy opiera się na intensyfikacji poszukiwań w pewnych "obiecujących" regionach obszaru dopuszczalnych rozwiązań.

Jedną z trudności, na które napotykają obliczenia ewolucyjne jest przedwczesne zatrzymywanie się obliczeń w pobliżu jakiegoś optimum lokalnego. Czynnikiem, który ma zapobiegać takim sytuacjom jest strategia dywersyfikacji polegająca na losowym zakłócaniu procesu poszukiwań celem odejścia od strefy optimum lokalnego. Przykładem strategii dywersyfikacji jest operacja mutacji, która polega na losowym zakłócaniu struktury osobnika. Przykładowo, w przytoczonym wcześniej problemie szeregowania mutacja może polegać na zamianie miejscami dwóch losowo wybranych zadań. Jeśli takiej zamianie podlegać mają zadania, odpowiednio pierwsze i drugie na liście (co ustalono w wyniku losowania), to osobnik reprezentowany przez wektor (4,5,1,3,2) w wyniku mutacji przyjmie postać (5,4,1,3,2).

W obliczeniach ewolucyjnych zdarza się, że w wyniku wymiany informacji lub stosowania strategii dywersyfikacji nowo powstały osobnik nie reprezentuje rozwiązania należącego do zbioru rozwiązań dopuszczalnych. Możliwe są wówczas trzy sposoby postępowania: osobnik taki jest, po prostu odrzucany, osobnik jest karany, a więc nadaje mu się bardzo niski wskaźnik jakości bądź wreszcie osobnik jest "reperowany" w wyniku czego znów reprezentuje rozwiązanie dopuszczalne.

Projektowanie algorytmów ewolucyjnych jest sztuką, w której projektant posiada znaczny stopień swobody w zakresie wyboru sposobu reprezentacji rozwiązania, reguł procesu ewolucji, struktury sąsiedztwa, źródeł informacji, strategii procesu ewolucji, intensyfikacji i dywersyfikacji, a także reakcji na rozwiązania niedopuszczalne. Tak duża dowolność sprawiła, że w ostatnich latach pojawiło się wiele różnych rodzajów algorytmów opartych na ewolucji populacji. W dalszych częściach pracy przedstawione zostały ważniejsze algorytmy omawianej klasy.

3. Algorytmy genetyczne

Najstarszą i szeroko znaną techniką obliczeń ewolucyjnych są algorytmy genetyczne. Pierwsze pomysły dotyczące omawianego podejścia pochodzą jeszcze z lat sześćdziesiątych [Holland 1962], [Fogel et al. 1966].

Ogólny pseudo-kod algorytmu genetycznego opartego na strategii wymiany generacji przedstawia tablica 1 (według [Hertz, Kobler 1998]):

Tab. 1. Algorytm genetyczny oparty na strategii wymiany generacji

```
Begin
    Wybierz liczbę całkowitą p≥2 i wygeneruj populację pierwotną P_o składającą się z p osobników
    Ustaw licznik iteracji: i := 0;
    While kryterium zatrzymania obliczeń nie jest spełnione do
        i := i + 1;
        P_i := ∅;
        While P_i zawiera mniej niż p osobników do
            Z P_{i-1} wybierz dwa osobniki I_1 oraz I_2;
            Do I_1 oraz I_2 zastosuj operator krzyżowania celem wygenerowania potomków O_1 oraz O_2;
            Dołącz O_1 oraz O_2 do P_i;
        EndWhile
        Do każdego osobnika w P_i zastosuj operator mutacji;
    EndWhile
End
```

W algorytmach genetycznych populacja pierwotna jest generowana w oparciu na mechanizmach losowych. Osobniki tworzące tą populację reprezentują rozwiązania dopuszczalne. Wybór osobników do reprodukcji poprzedzony jest obliczeniem wskaźników jakości. Prawdopodobieństwo wyboru jest zazwyczaj proporcjonalne do wartości wskaźnika jakości. Operatory krzyżowania i mutacji także wykorzystują mechanizmy losowe. W krzyżowaniu losowany może być punkt (bądź punkty) krzyżowania, zaś w przypadku mutacji zakres i miejsce wprowadzania ewentualnych zakłóceń.

W klasycznym programowaniu genetycznym osobnik jest reprezentowany przez łańcuch bitów. W przypadku reprezentacji opartej na innym schemacie, zamiast o algorytmach genetycznych mówi się o algorytmach ewolucyjnych.

4. Algorytmy wyspowe

Algorytmy wyspowe są szczególnym przypadkiem algorytmów ewolucyjnych. W algorytmie wyspowym występują dwa rodzaje osobników. Osobniki pierwszego rodzaju reprezentują rozwiązania dopuszczalne. Osobniki drugiego rodzaju są zbiorami takich rozwiązań nazywane wyspami. Ewolucja przebiega tu na dwóch poziomach. Na poziomie niższym ewolucja opiera się na identycznych zasadach jak w przypadku algorytmów genetycznych. Na poziomie wyższym odbywa się wymiana informacji między wyspami oraz migracja osobników pierwszego rodzaju. Projektowanie algorytmu wyspowego łączy się z wyborem topologii połączeń komunikacyjnych pomiędzy wyspami (np. pierścień skierowany, w którym najlepsze rozwiązania z danej wyspy są przesyłane do następnika lub sześcian gdzie komunikacja odbywa się pomiędzy sąsiadującymi wierzchołkami – wyspami). Innymi parametrami algorytmu wyspowego jest stopa migracji, która określa liczbę migrujących osobników pierwszego rodzaju, strategia ewolucji oraz częstotliwość migracji. Algorytmy wyspowe okazały się bardziej efektywne od algorytmów genetycznych [Cohoon et al. 1987], [Tanese 1987].

Pseudo-kod algorytmu wyspowego o topologii pierścienia przedstawiono w tablicy 2:

Tab. 2. Algorytm wyspowy o topologii skierowanego pierścienia.

```
Begin
    Określ liczbę wysp k oraz liczbę osobników tworzących wyspę p;
    Wygeneruj populację pierwotną składającą się z k * p dopuszczalnych osobników;
    Z populacji pierwotnej utwórz k wysp P1,...Pk;
    Ustaw licznik iteracji: i := 0;
    While kryterium zatrzymania obliczeń nie jest spełnione do
        i := i + 1;
        For each wyspa Pj do
            Z Pj wybierz dwa osobniki I1 oraz I2;
            Do I1 oraz I2 zastosuj operator krzyżowania celem wygenerowania potomków O1 oraz O2;
            Do O1 oraz O2 zastosuj operator mutacji;
            Zdecyduj czy O1 oraz O2 powinny wejść do Pj zastępując starsze osobniki;
        EndFor
        If i jest wielokrotnością danej liczby całkowitej n then
            Przenieś najlepsze rozwiązanie z każdej wyspy Pj do wyspy P(j mod k)+1;
    EndWhile
End
```

Warto zwrócić uwagę, że algorytmy wyspowe dają się w naturalny sposób realizować w wersji równoległej. Liczba dostępnych procesorów może wówczas odpowiadać liczbie wysp.

5. Kolonie mrówek

Algorytmy oparte na symulacji zachowania się mrówek poszukujących pożywienia stanowią szybko rozwijającą się technikę obliczeń ewolucyjnych (por. np. [Dorigo et al. 1996]). Pomysł opiera się na obserwacji mrówek, które używając prostego mechanizmu wzajemnej komunikacji potrafią znajdywać najkrótszą drogę pomiędzy dwoma punktami. W trakcie poszukiwań mrówki pozostawiają feromonowy ślad na powierzchni ziemi. Jego rolą jest kierowanie innych mrówek do punktu docelowego. Pojedyncza mrówka wybiera trasę w zależności od natężenia feromonu. Substancja ta rozkłada się w miarę upływu czasu. Ilość feromonu pozostawionego na trasie przez mrówki zależy od ilości znalezionego pożywienia, długości trasy oraz od liczby mrówek. Im więcej mrówek korzysta z trasy tym silniejszy jest ślad.

Reguły, na których oparta jest wymiana informacji między mrówkami są wykorzystywane do rozwiązywania kombinatorycznych problemów optymalizacyjnych. W tablicy 3 przedstawiono ogólny schemat algorytmu kolonii mrówek z intensyfikacją.

Tab. 3. Algorytm kolonii mrówek.

```
Begin
    Wygeneruj pierwotny ślad feromonowy;
    Wybierz liczbę mrówek p;
    While kryterium zatrzymania obliczeń nie jest spełnione do
        For j = 1 to p do
            Znajdź, dla kolejnej mrówki rozwiązanie opierając się na śladzie feromonowym;
            Zastosuj algorytm poprawy;
        EndFor
        Zaktualizuj ślad feromonowy;
    EndWhile
End
```

W pierwszym kroku algorytmu inicjalizowany jest ślad feromonowy. W trakcie kolejnych iteracji każda mrówka znajduje rozwiązanie problemu w oparciu na probabilistycznej funkcji zmiany stanów w zależności od śladu feromonowego. Kiedy dla wszystkich mrówek zostaną wygenerowane rozwiązania, uruchamiana jest globalna procedura aktualizacji śladu feromonowego. Obejmuje ona dwie fazy. Pierwsza jest fazą rozkładu, w której pewna część śladu zanika. Druga jest fazą wzmocnienia, w której każda mrówka wydziela pewną ilość feromonu proporcjonalnie do jakości uzyskanego rozwiązania. Proces jest powtarzany aż do spełnienia kryterium zatrzymania poszukiwań.

W modelu równoległym (wieloprocesorowym) kolonii mrówek jeden z procesorów pełni rolę serwera dla całego procesu poszukiwań, zaś pozostałe procesory są klientami tj. mrówkami poszukującymi rozwiązań [Talbi et al. 1999]. Na poziomie serwera zarządza się pamięcią centralną, w której zapamiętywany jest ślad feromonowy (np. w postaci macierzy). Serwer obsługuje procesy wymiany informacji w trakcie poszukiwań, a także przechowuje aktualnie najlepsze rozwiązanie. W każdej iteracji serwer przesyła macierz śladu feromonowego do wszystkich mrówek. Te ostatnie, korzystając z otrzymanej informacji generują rozwiązanie, a następnie starają się je polepszyć stosując jakiś algorytm poprawy. Uzyskane rozwiązanie jest przesyłane do serwera. Po otrzymaniu rozwiązań od wszystkich mrówek serwer uaktualnia ślad feromonowy i cykl jest powtarzany, aż do spełnienia kryterium zatrzymania poszukiwań.

6. Algorytm SLA

Algorytm SLA (ang.: *social learning algorithm*) jest nową techniką optymalizacji w klasie algorytmów opartych na ewolucji populacji. Algorytm ten, zaproponowany w [Jędrzejowicz 1998], symuluje procesy uczenia się w skali społecznej. Koncepcja algorytmu nawiązuje do wieloszczeblowych systemów edukacyjnych, w których naukę na niskich szczeblach rozpoczynana jest masowo, zaś na wyższe, coraz bardziej zaawansowane szczeble dociera coraz mniej coraz lepiej wykształconych jednostek. Zdobywanie wiedzy na każdym szczeblu opiera się m.in. na przekazie, ćwiczeniach praktycznych oraz wymianie informacji.

W algorytmie SLA, podobnie jak w poprzednio opisanych algorytmach, przedmiotem przetwarzania jest populacja osobników. Osobnikiem może być odpowiednio zakodowane rozwiązanie problemu bądź fragment rozwiązania bądź zbiór rozwiązań. Osobnikiem, może być także każdy inny dowolny obiekt pod warunkiem, że istnieje procedura, która pozwala na transformację tego obiektu na rozwiązanie rozważanego problemu.

Populacja pierwotna, generowana na początku procesu poszukiwania najlepszego rozwiązania jest, w przypadku SLA, możliwie liczna. Populacja, o której mowa powinna reprezentować w dostateczny sposób całą przestrzeń dopuszczalnych rozwiązań. Idealnie, w sąsiedztwie każdego optimum lokalnego powinien znaleźć się co najmniej jeden osobnik należący do populacji pierwotnej. Mechanizm generowania tej populacji posiada zazwyczaj charakter losowy.

Generowanie populacji pierwotnej poprzedza pierwszy etap nauki. Na tym etapie wszystkie osobniki poddawane są działaniu jakiegoś prostego, w sensie złożoności obliczeniowej, algorytmu poprawy. Po zakończeniu poprawiania obliczane są wskaźniki jakości osobników, a potem, zgodnie z przyjętą strategią selekcji, grupa osobników lepszych przechodzi do drugiego etapu. Cykl taki jest następnie powtarzany dla kolejnych etapów, przy czym stosowane algorytmy poprawy stają się coraz bardziej złożone, zaś liczba osobników docierających do wyższych etapów - coraz mniejsza. Na wybranych etapach dopuszcza się wymianę informacji między osobnikami (np. poprzez krzyżowanie). Proces poszukiwań może być prowadzony sekwencyjnie, w osobnych równoległych i niezależnych wątkach, bądź wreszcie w równoległych wątkach z wzajemną wymianą informacji.

W tablicy 4 przedstawiono ogólny pseudokod algorytmu SLA w wersji sekwencyjnej.

Tab.4. Algorytm SLA w wersji sekwencyjnej.

```
Begin
    Wybierz liczbę etapów m;
    Określ liczebność populacji pierwotnej J;
    For each etap i = 1,...,m do
        określ kryterium selekcji SCi;
        zaprojektuj procedurę uczenia-poprawy Li ;
    EndFor
    wygeneruj populację pierwotną osobników J ;
    For i = 1 to m do
        For each osobnik w J do
            Zastosuj procedurę uczenia-poprawy Li;
            Oblicz wskaźnik jakości;
        EndFor
            Usuń z J osobniki, których wskaźnik jakości nie spełnia kryterium selekcji SCi;
    EndFor
End
```

7. Programowanie genetyczne

Programowanie genetyczne jest przypadkiem szczególnym obliczeń ewolucyjnych, w którym każde rozwiązanie jest programem komputerowym. Ideę programowania genetycznego przedstawiono na przykładzie zaczerpniętym z pracy [Munakata 1998]. Poszukiwany jest sposób zapisu funkcji logicznej XOR przy użyciu trzech operatorów logicznych OR, AND oraz NOT. OR, AND i XOR są operatorami dwuargumentowymi zaś NOT jest operatorem jednoargumentowym. W rozważanym przykładzie operatory dwuargumentowe są traktowane jako operatory przedrostkowe. Tak więc używana jest notacja (AND P Q) zamiast (P AND Q). P oraz Q są tutaj zmiennymi logicznymi, a więc mogą przyjmować jedną z dwóch wartości - 0 (fałsz) lub 1 (prawda).

Rozwiązaniem rozpatrywanego problemu jest wyrażenie logiczne zbudowane z argumentów logicznych oraz operatorów OR, AND i NOT. Chociaż poszukiwana funkcja nie jest typowym programem to łatwo sobie wyobrazić poszerzenie użytej notacji o typowe instrukcje typu przyporządkowanie, pętla czy skoki warunkowe. Manipulowanie tymi obiektami jest możliwe przy użyciu dowolnego języka programowania, choć najwygodniejsze są języki symboliczne takie jak LISP lub PROLOG.

Ponieważ XOR jest funkcją dwuargumentową, w rozważanym przykładzie poszukiwane jest rozwiązanie zbudowane z operatorów OR, AND, NOT oraz dwóch argumentów P i Q. W tablicy 5 przedstawiono działanie funkcji XOR dla możliwych wartości jej argumentów:

Tab. 5. Funkcja XOR.

P	*Q*	(XOR *P Q*)
0	0	0
0	1	1
1	0	1
1	1	0

W programowaniu genetycznym, podobnie jak w przypadku algorytmów genetycznych, generowana jest losowo populacja pierwotna rozwiązań zbudowanych z dozwolonych operatorów i argumentów. Dla każdego rozwiązania obliczyć można wskaźnik jakości. Rozwiązania mogą być krzyżowane i można je poddawać mutacji. Jeśli, przykładowo, w trakcie obliczeń ewolucyjnych otrzymano dwa osobniki:

O_1: (AND(OR *P Q*) | (AND *P Q*))
O_2: (AND (OR (NOT P) *Q*) | (NOT (AND *P Q*)))

oraz jeśli symbol | oznacza punkt krzyżowania, to w wyniku operacji krzyżowania otrzymuje się dwa kolejne osobniki:

O_3: (AND(OR *P Q*) | (NOT (AND *P Q*)))
O_4: (AND (OR (NOT P) *Q*) | (AND *P Q*))

Każde rozwiązanie można tu interpretować jako drzewo. Każdy argument lub argument złożony (poddrzewo) jest węzłem drzewa. Punkt krzyżowania jest wybierany losowo i niezależnie dla każdego rozwiązania. Punktem tym może być jednak tylko korzeń każdego z poddrzew tworzących drzewo rozwiązania.

Wskaźnik jakości dla rozwiązania jest zdefiniowany jako liczba kombinacji *P* oraz *Q*, które spełniają warunek sformułowany w definicji XOR. Rozwiązanie z wskaźnikiem jakości 4 będzie rozwiązaniem optymalnym. Wskaźniki jakości dla przykładowych rozwiązań podano w tablicy 6.

Tab. 6. Wskaźniki jakości przykładowych rozwiązań.

P	*Q*	(XOR *P Q*)	O_1	O_2	O_3	O_4
0	0	0	0	1	0	0
0	1	1	0	1	1	0
1	0	1	0	0	1	0
1	1	0	1	0	0	1
		WJ =	1	2	4	1

W tablicy 6 osobnik O_3 reprezentuje rozwiązanie optymalne. Ponieważ istnieje wiele wyrażeń logicznych spełniających (XOR *P Q*) można by zmodyfikować sposób obliczania wskaźnika jakości tak aby wyeliminować rozwiązania długie z nadmiarowymi wyrażeniami. Przykładowo, do poprzednio używanego wskaźnika można dodawać wyrażenie: 3 - 0.1 * (liczba operatorów). Mutacja w rozpatrywanym przykładzie może polegać na negacji argumentu lub złożonego argumentu.

Rozszerzając repertuar wyrażeń o typowe instrukcje języków programowania w formacie operatorów przedrostkowych umożliwi generowanie programów konwencjonalnych. Przykładowo, przyporządkowanie "*x* := *a*b* + *c*" można zapisać jako (:= *x*(+(* *ab*)*c*)), zaś instrukcję "**if** *x* > 0, *y* = 1"

jako (**if** (> *x*0)(= *y*1). I te wyrażenia można interpretować jako drzewa i krzyżować w punktach, które są wierzchołkami poddrzew. Wskaźniki jakości będą odzwierciedlać odpowiednie cele związane z konstrukcją programu.

8. Zakończenie

W sensie najbardziej ogólnym obliczenia ewolucyjne są techniką, w której do dobrze znanej metody poszukiwania losowego dołączono funkcję inteligentnego sterowania procesem poszukiwań. W ten sposób działania losowe są, w określonym stopniu, koordynowane tak aby w sposób efektywny otrzymać rozwiązanie problemu. Obliczenia ewolucyjne bywają zagrożone dwoma czynnikami: trudnym do oszacowania z góry i być może długim czasem poszukiwania rozwiązania oraz niebezpieczeństwem zatrzymania się w pobliżu jakiegoś optimum lokalnego. W ogólności, algorytmy oparte na ewolucji populacji nie gwarantują uzyskania rozwiązań optymalnych. Można więc zadać pytanie dlaczego zawracać sobie głowę obliczeniami ewolucyjnymi zamiast skoncentrować się na klasycznych metodach optymalizacji jak np. technika mnożników Lagrange'a, programowanie dynamiczne czy też metody analizy matematycznej?

Odpowiedź na powyższe pytanie jest prosta. Praktyka dostarcza przykładów wielu problemów tak trudnych, że metody klasyczne okazują się nieprzydatne lub zbyt czasochłonne natomiast obliczenia ewolucyjne okazują się przydatne do rozwiązywania niektórych spośród takich trudnych problemów.

Obecnie sądzi się, że przyszłość, jeśli chodzi o rozwiązywanie trudnych problemów optymalizacyjnych, problemów uczenia się maszynowego, kopalnictwa danych, gromadzenia i przetwarzania wiedzy leży w podejściu hybrydowym, którego ważnym elementem będą obliczenia ewolucyjne wykorzystujące algorytmy oparte na ewolucji populacji rozwiązań.

Literatura

Cohoon, J., S.Hedge, W.Martin, D.Richards: *Punctuated equilibria: A parallel genetic algorithm,* Proc. Second International Conference on Genetic Algorithms, MIT, Cambridge, MA, 1987, s. 148 - 154.

Dorigo, M., V.Maniezzo, A.Colorni: *The ant system: Optimization by a colony of cooperating agents,* IEEE Transactions on Systems, Men, and Cybernetics, vol. 26 nr. 1, 1996, s. 29 - 41.

Fleurent, C., J.A.Ferland: *Genetic and Hybrid Algorithms for Graph Coloring,* In: "Metaheuristics in Combinatorial Optimization, G.Laporte, I.H.Osman (red.), Annals of Operations Research 63, Baltzer Science Publishers, 1996, s. 437 - 461.

Fogel, L.J., A.J.Owens, M.J.Walsh: *Artificial Intelligence through Simulated Evolution,* J.Wiley, New York, 1966.

Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning,* Addison-Wesley, Rewading, MA, 1989.

Hertz, A., D.Kobler: *A Framework for the Description of Population Based Methods,* Proc. 16th European Conference on Operational Research, vol. "Tutorials and Research Reviews", Brussels, 1998, s.1 - 23.

Holland, J.H.: *Outline for a Logic Theory of Adaptive Systems,* Journal of the ACM, nr 3, 1962, s. 297

Holland, J.H.: *Adaptation in Natural and Artificial Systems,* University of Michigan Press, Ann Arbor, 1975.

Jędrzejowicz, P.: *Social Learning Algorithm,* Katedra Informatyki WSM, Raport z badań KI/98/7, Gdynia, 1998.

Koza, J.R.: *Genetic Programming,* MIT Press, Cambridge, MA, 1992.

Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin, 1992.

Munakata, T.: *Fundamentals of the New Artificial Intelligence. Beyond Traditional Paradigms,* Springer, Berlin, 1998.

Tanese, R.: *Parallel geneticalgorithms for a hypercube*, Proc. Second International Conference on Genetic Algorithms, MIT, Cambridge, MA, 1987, s. 177 - 183.

Talbi, E., O.Roux, C.Fonlupt, D.Robillard: *Parallel Ant Colonies for Combinatorial Optimization Problems,* Lecture Notes in Computer Science, nr 1586, Springer, Berlin, 1999, s. 239 - 247.

Autor: ***Piotr Gawkowski***
Promotor: ***prof. dr hab. Janusz Sosnowski***

Instytut Informatyki,
Politechnika Warszawska, Warszawa

I nagroda w konkursie PTI

„Programowa symulacja błędów w środowisku Windows NT"

Badanie niezawodności sprzętu i oprogramowania jest zadaniem dość trudnym i wymagającym specjalistycznych narzędzi wspomagających ten proces. Do grupy takich narzędzi należą symulatory błędów. Pozwalają one symulować (wstrzykiwać, ang. *fault injection*) błędy w badanym systemie i obserwować wywołane przez nie efekty. W szczególnym przypadku technika ta pozwala na ocenę efektywności różnego rodzaju mechanizmów detekcji i tolerowania błędów oraz programów diagnostycznych. W literaturze dotyczącej symulacji błędów wyróżnić można dwie techniki:

- wprowadzanie błędów do modelu (symulowanego) badanego systemu
- wprowadzanie błędów do rzeczywistego systemu.

Koncepcja wstrzykiwania błędów do modelu testowanego systemu jest bardzo atrakcyjna. Symulacja testowanego systemu pozwala bowiem na pełną kontrolę nad czasem generacji, typem i miejscem błędu. Możliwości tej techniki testowania uzależnione są od poziomu szczegółowości stosowanego w niej modelu analitycznego. Wadą jest olbrzymi nakład pracy włożony w opracowanie modelu testowanego systemu oraz duży narzut czasowy na samą symulację. Stąd stosując tę technikę wprowadza się różnego rodzaju uproszczenia i ograniczenia przez co symulatory takie dedykowane są dla konkretnych aplikacji. Przykłady rozwiązań tego typu znaleźć można w [5, 10, 14].

Bardziej uniwersalne wydają się być techniki wstrzykiwania błędów do rzeczywistych systemów. Wyróżnić tu można trzy podejścia:

- poprzez wymuszanie błędów fizycznych (na poziomie sprzętu) w badanym systemie (np. [9])
- poprzez programowe symulowanie błędów w badanym systemie (np. [3])
- poprzez implementację hybrydową (np. [13]), wykorzystującą jednocześnie pierwsze dwie koncepcje.

Sprzętowe wstrzykiwanie błędów jest przede wszystkim szybkie i realistyczne - w testowanym systemie generowane są rzeczywiste błędy. Przykłady takich rozwiązań znaleźć można w [1, 7, 8, 9, 11]. Mają one jednak szereg wad. Po pierwsze, wymagają one dodatkowego sprzętu, który najczęściej jest dedykowany jedynie do testowanego systemu, a to z kolei podnosi znacznie koszty eksperymentów. Duża złożoność i szybkość współczesnych procesorów sprawia, że projektowanie tego dodatkowego sprzętu jest bardzo trudnym zadaniem. Główny problem tkwi nie tylko w samej generacji błędu, ale w umożliwieniu późniejszej obserwacji

testowanego systemu i kontroli nad nim. Często zachodzi konieczność porównywania wszystkich wyjść układów scalonych w testowanym systemie (cykl po cyklu) z wyjściami układów w systemie wzorcowym.

Alternatywą dla sprzętowego wstrzykiwania błędów jest programowa ich symulacja (np. [2, 3, 6, 12]). W najprostszym ujęciu realizowana ona jest poprzez przerwanie wykonywania się testowanego programu (zazwyczaj wykorzystując do tego instrukcję pułapki programowej) i wykonanie kodu, który emuluje błędy sprzętu. Emulacja błędów polega na zmianie stanów różnych modułów systemu komputerowego takich jak rejestry procesora, pamięć danych czy kod testowanego programu. Zaletami implementacji programowej są: niski koszt (nie jest wymagany z reguły żaden dodatkowy sprzęt), elastyczność rozwiązania i możliwość przenoszenia na inne platformy. Metoda taka nie jest jednak pozbawiona wad. Podstawową wadą jest nakład czasowy potrzebny na zasymulowanie błędów sprzętu. Inną wadą jest często konieczność ingerencji w kod aplikacji poddawanej testowaniu. Poza tym, istnieją różnego rodzaju ograniczenia, które mogą nie pozwolić na symulację pewnych klas błędów. Efektywność emulacji błędów można istotnie zwiększyć wykorzystując mechanizmy monitorowania i uruchamiania spotykane w nowych procesorach [4].

W przedstawianej tu pracy dyplomowej zrealizowany został programowy symulator błędów. Istotną cechą opracowanego systemu, wyróżniającą go spośród innych tego typu rozwiązań, jest przyjęcie standardowego środowiska systemów IBM PC działających pod kontrolą rodziny systemów operacyjnych Win32: Windows 95, 98 oraz Windows NT. Wydaje się, iż jest to jedyny tego typu system symulacji błędów opracowany dla tego środowiska. Wybrana platforma programowa zapewnia wygodny w użyciu interfejs użytkownika oraz łatwość w przenoszeniu na inne platformy sprzętowe przez nią wspierane (PowerPC, MIPS, Alpha). Całość opracowanego systemu została napisana w języku C++, co pozwala na łatwy jego rozwój. Oprócz pełnej kontroli projektanta eksperymentu nad momentem generacji błędu, miejscem i rodzajem wstrzykiwanego błędu, symulator pozwala również na analizę wyników eksperymentu oraz danych o samym programie testowanym. System umożliwia także przeprowadzanie eksperymentów całkowicie pod jego kontrolą - bez udziału użytkownika. Eksperymenty można tak skonfigurować, by zapewnić pełną ich powtarzalność. Do przeprowadzania eksperymentów nie potrzeba dodatkowego komputera monitorującego, jak ma to miejsce w większości tego typu systemów (np. [3]). Niektóre z możliwości opracowanego systemu symulatora błędów są nieco ograniczone w porównaniu z potencjalnymi możliwościami stosowanej platformy sprzętowej i programowej. System jest jednak otwarty na dalszy rozwój. W pracy przedstawiono zarówno możliwości funkcjonalne jak i zasadę działania symulatora błędów. Szczegółowo opisany został mało znany a wykorzystywany przez symulator błędów interfejs programowy będący podstawą jego działania - tzw. *Debugging API* systemu Win32. Praca zawiera też szczegółowe informacje o implementacji symulatora błędów. Wykorzystując opracowany system symulacji błędów przeprowadzono wiele eksperymentów z różnymi programami aplikacyjnymi. Wykonane eksperymenty miały głównie na celu zweryfikowanie pracy symulatora a także zbadanie efektywności różnych mechanizmów systemowych oraz niektórych programowych mechanizmów detekcji i tolerowania błędów. Dodatkowo przeanalizowano wpływ różnych klas błędów na testowane aplikacje. W pracy dyplomowej zaprezentowano i omówiono najciekawsze wyniki z przeprowadzonych eksperymentów.

Opracowany system symulacji błędów pozwala na dużą dokładność specyfikacji eksperymentów oraz szczegółowość zbierania danych co z kolei pozwala na wnikliwą analizę efektów błędów układowych w środowisku systemowym. Symulacja błędów okazała się bardzo przydatna przy badaniu efektywności różnych programowych mechanizmów detekcji i tolerowania błędów. Ponadto wykorzystano ją również do oceny i optymalizacji profesjonalnego testu koprocesora arytmetycznego procesora Intel Pentium. Przewidywane są dalsze badania eksperymentalne. Funkcjonalność zaś symulatora błędów podlega ciągłemu rozwojowi.

Literatura

1. J.Arlat, *Fault Injection for Dependability Validation: A Methodology and Some Applications*, IEEE Transactions on Software Engineering, vol. 16, no. 2, pp. 166-182, February 1990.

2. A.Bens, P.Prinetto i inni, *A Fault Injection approach for embedded Microprocessor-based Boards*, The Proc. of Ninth European Workshop on Dependable Computing: testing methods and tools for modern computer systems and networks, Gdańsk, 14-16 May, 1998, pp. 2-6.

3. Joao Carreira, Henrique Madeira, Joao Gabriel Silva, *Xception: Software Fault Injection and Monitoring in Processor Functional Units*, DCCA-5, Conference on Dependable Computing for Critical Applications, September 1995, pp. 135-149.

4. Joao Carreira, Henrique Madeira, Joao Gabriel Silva, *Xception: A Technique for the Experimental Evaluation of Dependability in Modern Computers*, IEEE Transactions on Software Engineering, Vol. 24, No. 2, February 1998, pp. 125-136.

5. D.Gil, J.C.Baraza, J.V.Busquets, P.J.Gil, *Fault Injection into VHDL Models: Analysis of the Error Syndrome of a Microcomputer System*, Proc. of the 24th EUROMICRO Conference, Västerås, Sweden, August 25-27 1998, EUROMICRO 98, Volume I, pp. 418-425.

6. G.A.Kanawati, N.A. Kanawati, J.A. Abraham, *FERRARI: A Tool for the Validation of System Dependability Properties*, Proc. 22nd. Int. Symp. on Fault-Tolerant Computing (FTCS-22), pp. 336-344.

7. J.Karlsson, P.Lidén, P.Dahlgren, R.Johansson, U.Gunneflo, *Using Heavy-ion Radiation to Validate Fault-Handling Mechanisms*, IEEE Micro, Vol. 14, No. 1, pp. 8-32, 1994.

8. H.Madeira, M.Rela, F.Moreira, J.Silva, *RIFLE: A General Purpose Pin-Level Fault Injector*, Proc. First European Dependable Computing Conf., Berlin, pp. 199-216, October 1994.

9. G.Miremadi, J.Karlsson, U.Gunneflo, J.Torin, *Two Software Techniques for On-Line Error Detection*, Proc. FTCS-22, pp. 328-335,1992.

10. S.Racek, P.Herout, J.Hlavicka, *Fault Injection as a Tool for FT Process Controller Design*, Proc. of Design and Diagnostics of Electronic Circuits and Systems, Workshop, Szczyrk, 2-4 September 1998, pp. 123-128.

11. J.R.Samson, W.Moreno, F.Falquez, *A Technique for Automated Validation of Fault Tolerant Designs Using Laser Fault Injection (LFI)*, FTCS-28, IEEE Computer Society, 1998, pp. 162-167.

12. Z.Segall, D.Vrsalovic, D.Sieviorek, D.Yaskin, J.Kownacki, J.Barton, D.Rancey, A.Robinson, A.Lin, *FIAT-Fault Injection based Automated Testing Environment*, FTCS-18, pp. 102-107.

13. L.T.Young, R.K.Iyer, K.K.Goswami, *A Hybrid Monitor Assisted Fault Injection Environment*, Proc. Third IFIP Working Conf. Dependable Computing for Critical Applications, Italy, Sept. 1992.

14. Charls R. Yount, Daniel P.Siewiorek, *A Methodology for the Rapid Injection of Transient Hardware Errors*, IEEE Transactions on Computers, Vol. 45, No. 8, August 1996

Tomasz Kaźmierczuk
Andrzej Miazga

Tomasz Kaźmierczuk, email: Tomasz.Kazmierczuk@ComArch.Pl
Andrzej Miazga, email: Andrzej.Miazga@ComArch.Pl

II nagroda w konkursie PTI

Konspekt prezentacji pracy magisterskiej

Przeglądanie i testowanie aplikacji pracujących w środowisku CORBA

Praca związana jest z tematyką projektowania i budowy trójwarstwowych architektur nowoczesnych systemów informatycznych. Systemy te są obecnie najczęściej budowane przy użyciu przemysłowego standardu CORBA, który specyfikuje mechanizmy budowy rozproszonych, obiektowych systemów. Przedmiot pracy stanowi projekt i realizacja narzędzia dostarczającego funkcjonalność umożliwiającą przeglądanie i testowanie aplikacji stworzonych zgodnie ze standardem CORBA.

Cele pracy:

1. Stworzenie narzędzia umożliwiającego przeglądanie i testowanie jak największych obszarów funkcjonalności systemów informatycznych zbudowanych zgodnie ze standardem CORBA.
2. Zaprojektowanie komponentowej i rozszerzalnej architektury narzędzia.
3. Stworzenie interfejsu graficznego „przyjaznego" dla użytkownika pozwalającego na intuicyjne korzystanie z narzędzia.

Podstawą dla budowy wykonanego narzędzia jest mechanizm refleksji istniejący w standardzie CORBA. Mechanizm ten umożliwia uzyskanie informacji o wewnętrznej budowie systemu w trakcie jego działania. Narzędzie zaprojektowano i zrealizowano według architektury komponentowej, czyli poprzez wyodrębnienie komponentów, z których każdy odpowiedzialny jest za określony etap testowania. Każdy z tych komponentów posiada wyraźnie wyspecyfikowane interfejsy wejściowy i wyjściowy w sposób umożliwiający ich ścisłą współpracę. Pozwala to również na wykorzystanie każdego z nich w innych programach potrzebujących funkcjonalności dostarczanej przez dany komponent. Stworzone komponenty intensywnie wykorzystują zdolność refleksji systemów działających w architekturze CORBA wspieraną przez metawiedzę zawartą w ich bazach wiedzy. Zasadnicza część pracy dotyczy opisu zdolności refleksji i sposobów wykorzystania jej do celów testowania systemów CORBA.

Przyjęto następujący porządek prezentacji pracy:

- Opis trzypiętrowej architektury budowy nowoczesnych systemów informatycznych,
- Podstawowe cechy standardu CORBA budowy rozproszonych aplikacji obiektowych,

- Opis języka IDL służącego do definiowania funkcjonalności obiektów CORBA,
- Prezentacja rozproszonych baz metawiedzy o stanie systemu zdefiniowanych w standardzie CORBA, dzięki którym realizowany jest mechanizm refleksji w tym środowisku,
- Zdefiniowanie problemu przeglądania i testowania aplikacji stworzonych w systemie CORBA,
- Projekt komponentowej architektury narzędzia o rozbudowanej funkcjonalności,
- Przykłady funkcjonowania narzędzia,
- Wnioski końcowe i zastosowania.

Pracę wykonano w Katedrze Informatyki Akademii Górniczo – Hutniczej w Krakowie, na sieci lokalnej komputerów SUN. Wykonane narzędzie zostało wykorzystane przy implementacji złożonego systemu ABng w Katedrze Informatyki AGH.

Jerzy Nawrocki

Instytut Informatyki, Politechnika Poznańska
ul. Piotrowo 3a, 60-965 Poznań,
fax: 0-61 8771525
Jerzy.Nawrocki@put.poznan.pl

Szacowanie rozmiaru oprogramowania

Streszczenie

Szacowanie rozmiaru oprogramowania odgrywa ważną rolę w planowaniu przedsięwzięć programistycznych. Upraszczając można powiedzieć, iż planowanie sprowadza się do trzech czynności: szacowania rozmiaru tworzonego oprogramowania, na tej bazie szacowania pracochłonności i wreszcie - w oparciu o kalendarz - opracowania harmonogramu przedsięwzięcia.

W dobrze prowadzonym przedsięwzięciu programistycznym przed przejściem do kolejnej fazy przedsięwzięcia aktualizuje się plany w oparciu o dane z poprzedniej fazy. Planowanie jest więc czynnością, którą wykonuje się wielokrotnie. Jednakże w miarę upływu czasu zmienia się kontekst planowania. Inna jest sytuacja na samym początku projektu, gdy nie ma nawet dokładnych wymagań użytkownika, a inna po fazie projektowania, gdzie o tworzonej aplikacji wiemy już bardzo dużo. Ten zmieniający się kontekst planowania wymaga od kierownika przedsięwzięcia znajomości różnych metod szacowania rozmiaru oprogramowania. Niektóre z metod bardziej pasują do wczesnych faz, inne zaś wymagają danych, które są dostępne dopiero po opracowaniu projektu.

W trakcie wykładu przedstawione zostaną podstawowe metody szacowania oprogramowania: metoda Delphi zaproponowana przez Boehma, metoda rozmytych wartości i metoda standartowego składnika (obie opracowane przez Putnama), metoda punktów funkcyjnych Albrechta oraz metoda PROBE wprowadzona przez Humphrey'a.

XI Górska Szkoła PTI

Szczyrk, 21-25 czerwca 1999

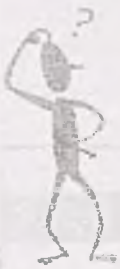
Software Size Estimation

Jerzy Nawrocki

Jerzy.Nawrocki@put.poznan.pl

Copyright © 1999 © Jerzy R. Nawrocki

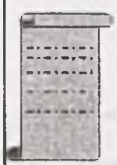
Introduction



What to estimate software size for ?

- To make plans.
- To estimate the effort.
- To create a reliable schedule.

Plan of the lecture



- Introduction
- Wideband-Delphi method
- Fuzzy-logic method
- Standard-component method
- Function-point method
- The Probe method
- Summary

Wideband-Delphi method

Rand Corporation, Boehm'81

- A few experts individually produce size estimates.
- A Delphi process is used to reach a consensus.

Wideband-Delphi method

The Wideband-Delphi procedure

1. Experts get the specification and an estimation form
2. They meet for discussion (project goals, assumptions, estimation issues)
3. Each expert anonymously lists the tasks and estimates the size
4. The estimates go to the estimate moderator. He tabulates the results and returns them to the experts.

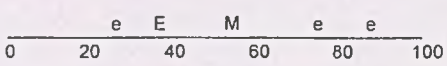
The estimate moderator

Wideband-Delphi method

Estimator: Jerzy Nawrocki Date: 22.06.1999

Project: Sorting routine

The estimates from the 1st round:



e - estimates, E - your estimate, M - median estimate

Your estimate for the next round: ..... LOC

A rationale for your estimate: .....

.....

Wideband-Delphi method

The Wideband-Delphi procedure

5. The experts meet to discuss the results. They review the tasks they have defined but not their size estimates.
6. The procedure is repeated from step 3 until the estimates are acceptably near

The estimate moderator

Fuzzy-logic method

Putnam'92:

- We need accurate, not necessary precise size estimates.
- Let's relate the estimate to the historical data

Tall Middle Short

Fuzzy-logic method

Fuzzy-logic size ranges

Range	Low	Middle	High
V. small			
Small			
Medium			
Large			
V. large			

Fuzzy-logic method

Tall Middle Short ?

Remarks:

- Historical data are required on a large number of projects.
- Don't change the existing ranges. Add new ranges when required.

Standard-component method

House
2 bathrooms
4 bedrooms
1 kitchen
1 living room

Putnam'92

- We have a system decomposed into a set of standard components (screens, files, modules etc.)
- For each standard component we have historical data about its size

Standard-component method

You have to provide for each component type:

S - the smallest number of components of that type;

L - the largest number

M - the most likely number

Estimated number of components of a given type:

(S+4*M+L) / 6

For each component type we know average LOC.

Function-point method

Albrecht, IBM, 1979

Basic functions:

- Inputs
- Outputs
- Inquiries
- Data files
- Interfaces

Function-point method

Counts	Fun types	Weights	Total

Function-point method

FP = UT * CM

FP - Function points

UT - Unadjusted total (previous slide)

CM - Complexity multiplier (0.65 .. 1.35)

CM = 0.65 + 0.01 * Σ Influence_factors

14 influence factors, 0 - 5 points each

Function-point method

Influence factors

Data communications	End-user efficiency
Distributed functions	Transaction rate
Performance objectives	On line update
Heavily used configuration	Installation ease
On line data entry	Operational ease
Complex processing	Multiple sites
Reusability	Facilitate change

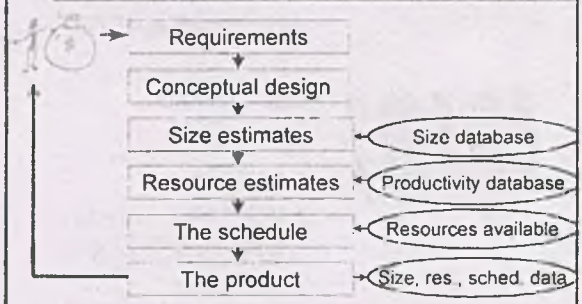
The Probe method

Humphrey, CMU, 1995

PROxy-Based Estimating

Objects as proxies

The Probe method



The Probe method

Conceptual design

Identify objects

Number of methods | Object type | Relative size | Reuse categories

Calculate projected and modified LOC

Estimate program size

Calculate prediction interval

The Probe method

Linear regression:

Expected_size= β_0 + Guessed_size * β_1

Prediction interval: Student's t distribution

Range= $t(\alpha/2, n-2)\sigma \sqrt{(1 + 1/n + M)}$

$M= (x_k - x_{avg})^2 / \Sigma(x_i - x_{avg})^2$

$t(\alpha/2, n)$ for p=.70 is $t(\alpha, n)$ for p'=(1+p)/2=.85

Summary

Size estimating methods:

- Wideband-Delphi method
- Fuzzy-logic method
- Standard-component method
- Function points
- The Probe method

Rafał Juraszek

PROJEKT ALSO MPiPS

ComputerLand®

System PULS, Wersja: 1
Zatwierdzone przez:
ComputerLand
Wojciech Ziółkowski

Wdrażanie Systemu Zapewnienia Jakości w projekcie ALSO-PULS

1. Podstawowe informacje o projekcie ALSO-PULS

W maju 1991 Minister Pracy w imieniu polskiego rządu podpisał umowę z Bankiem Światowym o udzielenie Polsce pożyczki na Projekt Promocji Zatrudnienia i Usług. W ramach projektu wyodrębniono 9 grup zadaniowych (TOR - Terms Of Reference), których zadaniem było zbadanie poszczególnych obszarów rynku pracy i przedstawienie koncepcji zmian strukturalnych i prawnych, a także sformułowanie celów, metod zarządzania i monitorowania tegoż rynku.

Efektem pracy grup było powołanie projektu "Automatyzacji Systemu Urzędów Pracy i Organizacji Pomocy Społecznej" - ALSO (Automation of the Labour and Social Welfare Organisation), który finansowany jest ze środków pożyczki z Banku Światowego (TOR #3).

Projekt ALSO koordynuje zespół utworzony w tym celu przez Biuro Informatyki Ministerstwa Pracy i Polityki Socjalnej (MPiPS).

Celem Projektu ALSO jest wsparcie technologią informatyczną zdolności Ministerstwa Pracy i Polityki Socjalnej w realizacji jego misji w wybranych obszarach jego działania.

Projekt ALSO składa się z trzech wzajemnie powiązanych podprojektów (w dalszej części opracowania zwanych projektami):

1. Podprojekt Systemu Pomocy Społecznej (SPS) - **Projekt POMOST.**
2. Podprojekt Systemu Urzędów Pracy (SUP) - **Projekt PULS.**
3. Podprojekt Systemu Wsparcia MPiPS.

Wykonawcą oprogramowania w ramach projektu POMOST i PULS jest firma ComputerLand.

Celem projektu ALSO-PULS jest:

- usprawnienie pośrednictwa pracy poprzez efektywne kojarzenie bezrobotnych z wolnymi miejscami pracy oraz usprawnienie zbieranie informacji o rynku pracy
- usprawnienie kontroli finansowej, tak by:
  - efektywniej gospodarować środkami finansowymi
  - zoptymalizować wydatki
  - poprawić alokację środków
  - ograniczyć możliwość nadużyć finansowych
- poprawienie organizacji pracy poprzez usprawnienie systemu informacyjnego jednostek SUP, planowanie, badanie efektywności itp.

- usprawnienie rutynowych czynności w jednostkach SUP poprzez zautomatyzowanie tych czynności i zmniejszenie obciążenia pracowników

Oto bardzo uproszczona historia projektu PULS:

1. We wrześniu 1995 roku został przez MPiPS ogłoszony przetarg na wykonanie systemu PULS.
2. Przetarg wygrały dwie firmy: Andersen Consulting i CSBI.
3. CSBI podpisało kontrakt w lipcu 1996 roku.
4. Dokumentacja projektowa wersji 1.0 systemu PULS została wstępnie zaakceptowana w listopadzie 1996 roku.
5. W tym roku ogłoszono bardzo poważną nowelizację ustawy o zatrudnieniu i przeciwdziałaniu bezrobociu, będącej podstawą do realizacji systemu. Co spowodowało konieczność przebudowy systemu PULS.
6. W listopadzie 1997 roku CSBI uzyskało pozytywny wynik testu wstępnego przeprowadzonego przez MPiPS.
7. Od lutego do grudnia 1998 roku trwały wdrożenia pilotowe w województwach: siedleckim, kaliskim, leszczyńskim i częstochowskim.
8. W maju 1998 roku powołano Zespół ds. Zapewnienia Jakości Projektu ALSO-PULS.
9. W listopadzie 1998 roku została zatwierdzona przez MPiPS dokumentacja projektu szczegółowego wersji 1.4 oprogramowania aplikacyjnego PULS.
10. W lutym 1999 CSBI zostało włączone do firmy ComputerLand, która przejęła wszystkie zobowiązania firmy CSBI.
11. W marcu 1999 rozszerzono zakres umowy z MPiPS o obszar przyznany wcześniej firmie Andersen Consulting, która wcześniej zrezygnowała z realizacji projektu.
12. W marcu 1999 podpisano pięcioletnią umowę na serwis oprogramowania PULS z Krajowym Urzędem Pracy.
13. W marcu 1999 wersja 1.4 oprogramowania aplikacyjnego PULS otrzymała świadectwo akceptacyjne, co pozwoliło na rozpoczęcie procesu wdrożeń masowych .
14. Od kwietnia trwają wdrożenia masowe (378 jednostek Systemu Urzędów Pracy).

Plany w zakresie realizacji projektu ALSO-PULS są następujące:

1. Dalsza realizacja pięcioletniej umowy na utrzymanie systemu PULS.
2. Zatwierdzenie dokumentacji projektowej wersji 1.7 systemu PULS.
3. Test akceptacyjny wersji 1.7 systemu PULS.
4. Zatwierdzenie dokumentacji projektowej wersji 2.0 systemu PULS.
5. Test akceptacyjny wersji 2.0 systemu PULS.
6. Zakończenie wdrożeń masowych wersji 1.4 systemu PULS.

W ramach projektu PULS na podstawie wymagań określonych przez MPiPS firma ComputerLand zrealizowała następujące aplikacje składające się na wersję 1.4 systemu PULS:

- Formalna Obsługa Osób (obsługa Powiatowego Urzędu Pracy – PUP w zakresie obsługi bezrobotnych)
- Pośrednictwo Pracy (obsługa PUP w zakresie kojarzenia pracodawców z kandydatami do pracy)
- Szkolenia i Poradnictwo zawodowe (obsługa PUP w zakresie rejestracji instytucji szkolących, organizowanych przez nie szkoleń, kwalifikacji kandydatów na szkolenia itp.)
- Prace Interwencyjne, Roboty Publiczne i Refundacje (obsługa PUP w zakresie tworzenia miejsc pracy w ramach prac okresowych, np. prac interwencyjnych, robót publicznych itp.)
- Zasiłki (obsługa PUP w zakresie przygotowania wypłaty zasiłków dla bezrobotnych)
- Finanse (obsługa PUP, Wojewódzkiego Urzędu Pracy - WUP, Krajowego Urzędu Pracy – KUP w zakresie prowadzenia

rachunkowości jednostki, realizacji budżetu, wykorzystania funduszy celowych)

- Statystyka (obsługa PUP, WUP i KUP w zakresie prowadzenia statystyki i sprawozdawczości na potrzeby wewnętrzne i urzędów statystycznych)
- Zarządzanie systemem (obsługa PUP, WUP i KUP w zakresie zarządzania użytkownikami systemu, słownikami, archiwizacji itp.)

Aktualnie są opracowywane wersje 1.7 i 2.0 systemu PULS. Wersja 1.7 zawiera dodatkowo następujące aplikacje:

- Majątek Trwały (obsługa PUP, WUP i KUP w zakresie zarządzania majątkiem trwałym jednostki, to znaczy jego ewidencji, naliczania umorzeń, inwentaryzacji itp.)
- Kadry (obsługa PUP, WUP i KUP w zakresie rejestracji danych pracowników jednostki SUP)
- Płace (obsługa PUP, WUP i KUP w zakresie naliczania wynagrodzeń)

Wersja 2.0 ma ponadto zawierać następujące aplikacje:

- Pożyczki (obsługa PUP w zakresie udzielania pożyczek bezrobotnym i pracodawcom)
- Zatrudnianie Cudzoziemców w Polsce (obsługa WUP w zakresie indywidualnych i grupowych wniosków o zatrudnianie cudzoziemców)
- Zatrudnianie Polaków za Granicą (obsługa WUP w zakresie imiennych i grupowych umów sezonowych itd.)
- Zastępcza Służba Wojskowa (obsługa WUP w zakresie pozyskiwania miejsc pracy dla poborowych, kierowania poborowych do odbycia służby zastępczej itp.)
- Odwołania, Skargi i Wnioski (obsługa PUP, WUP i KUP w zakresie rejestrowania skarg, odwołań i wniosków napływających do jednostki SUP)
- Kontrola Legalności Zatrudnienia (obsługa PUP w zakresie zbierania informacji o planowych i przeprowadzonych kontrolach pracowników i pracodawców)
- Prawodawstwo (obsługa PUP, WUP i KUP w zakresie rejestrowania i dystrybuowania interpretacji aktów prawnych, które zostały wykonane przez KUP)
- Wskaźniki Efektywności (obsługa PUP, WUP i KUP w zakresie wyliczania wskaźników określających skuteczność i celowość wydatkowania środków finansowych)

System PULS jest realizowany i wdrażany przez około 40 specjalistów (analitycy, projektanci, programiści, specjaliści ds. wdrożeń, specjaliści ds. zapewnienia jakości, specjaliści ds. testów) przy ścisłej współpracy z kilkoma podwykonawcami. Wielkość systemu w kwietniu 1999 roku była następująca:

- aplikacje - 10
- projekt - 7400 stron
- moduły - 1000
- encje (tablice) - 460
- atrybuty - 3900
- powiązania - 860
- linie kodu - 3,6 mln.

System PULS został zrealizowany w architekturze klient–serwer za pomocą narzędzi firmy PROGRESS. Po stronie serwera, baza danych (RDBMS) działa na platformach systemów operacyjnych Novell Netware, Unix i Windows NT. Po stronie klienta wykorzystywany jest system operacyjny Windows 3.1x, Windows 95, Windows 98 lub Windows NT Workstation. Połączenie klienta z serwerem jest realizowane w oparciu o sieciowe protokoły IPX/SPX i TCP/IP.

Do wymiany danych pomiędzy jednostkami SUP mogą być wykorzystywane sieci rozległe (INTERNET, INTRANET).

System obejmuje ponad 400 instalacji (jednostek SUP). Jedna instalacja obejmuje przeciętnie 25 stacji roboczych.

2. Definicja jakości w projekcie i plan wdrażania systemu zapewnienia jakości

Elementy jakości istniały w projekcie od podpisania kontraktu. Już w ofercie CSBI istniały zapisy dotyczące planów jakości, kontroli, zespołu jakości. Początkowo podejście do problematyki jakości nie było systemowe.

W maju 1998 roku wyodrębniono Zespół ds. Zapewnienia Jakości, który miał za zadanie uporządkować sprawy jakości w projekcie. Działanie to zbiegło się z auditem projektu wykonanym przez MPiPS.

W odpowiedzi na wyniki auditu rozpoczęto prace nad Księgą Jakości Projektu, która miała stać się konstytucją projektu ALSO-CSBI. Pod koniec czerwca wysiłkiem kierownictwa projektu powstała „Księga Jakości Projektu ALSO-CSBI". Dokument ten został opracowany według normy ISO-9001, a w szczególności wytycznych dotyczących oprogramowania zapisanych w normie ISO-9000-3. W Księdze Jakości zostały zebrane i uporządkowane sprawy dotyczące projektu. Księga Jakości zawiera następujące elementy:

1. Definicję jakości w projekcie i cele projektu.
2. Wykaz cech produktu, które umożliwią realizację tych celów.
3. Strukturę organizacyjną i kluczowe stanowiska w projekcie.
4. Szczegółowe zakresy działania (funkcje) poszczególnych stanowisk i jednostek organizacyjnych.
5. Kompetencje poszczególnych stanowisk w projekcie (zakres odpowiedzialności i uprawnień).
6. Sposób weryfikacji jakości w projekcie (Komitet Sterujący, Rada Projektu, audity).
7. Strukturę dokumentacji projektu i produktu.
8. Wykaz procedur projektu.
9. Sposób planowania jakości w projekcie.
10. Procedurę ustalania i dokumentowania wymagań klienta.
11. Procedurę przeglądu ofert i umów.
12. Opis 3 etapowego (projekt koncepcyjny, funkcjonalny, szczegółowy) procesu zarządzania pracami projektowymi z uwzględnieniem metody inspekcji.
13. Wykaz dokumentów projektu i produktu, które są objęte nadzorem. Sposób prowadzenia nadzoru nad dokumentacją.
14. Sposób nadzoru nad pracami wykonywanymi przez podwykonawców w zakresie opracowywania oprogramowania i jego wdrożenia.
15. Cykl życia projektu w podziale na ścieżki, fazy i procesy.
16. Sposób w jaki klient uczestniczy w zarządzaniu projektem.
17. Procedurę testowania oprogramowania w trakcie jego wykonania.
18. Procedurę testu końcowego produktu.
19. Sposób postępowania w przypadku wykrycia błędu w produkcie w różnych fazach jego wytwarzania.
20. Definicję działań korygujących i zapobiegawczych w projekcie.
21. Wyjaśnienie na czym polegają wewnętrzne audity jakości projektu.
22. Metodę zapewnienia odpowiednich kwalifikacji zespołu realizującego projekt.
23. Plan wdrażania systemu zapewnienia jakości w projekcie.

Jako wzorzec Systemu Zapewnienia Jakości (SZJ), do którego należy dążyć przyjęto zapisy zawarte w uzgodnionej z MPiPS „Księdze Jakości Projektu ALSO-PULS", normy ISO-9001 i ISO-9000-3 oraz zawarte w ofercie CSBI zapisy dotyczące jakości.

System Zapewnienia Jakości miał być wdrażany w następujących etapach i terminach:

1. Upowszechnienie polityki jakości (szkolenia personelu dotyczące norm ISO i planu wdrażania systemu zapewnienia jakości) - czerwiec 1998
2. Analiza procesów w projekcie – czerwiec 1998
3. Opracowywanie, zatwierdzenie i przekazanie dokumentów systemu zapewnienia jakości – sierpień 1998
4. Przeszkolenie pracowników z obowiązujących ich dokumentów – wrzesień 1998
5. Wdrożenie procedury auditu wewnętrznego – październik 1998

6. Wdrożenie działań korygujących i zapobiegawczych – listopad 1998
7. Wdrożenie przeglądów kierowniczych - listopad 1998

Co wynikło z tych ambitnych założeń zostanie przedstawione w następnym rozdziale.

3. Wdrażanie systemu zapewnienia jakości w projekcie

Zostały przyjęte następujące założenia dotyczące jakości:

„W projekcie informatyzacji SUP konieczne jest zapewnienie wysokiej jakości produktów i usług. Uzyskane w ten sposób korzyści to, przede wszystkim spełnienie wymagań użytkownika (zgodność ze specyfikacją i zatwierdzonym projektem) oraz terminowe wykonanie prac (zgodność z przyjętym harmonogramem, brak nawrotów). Wysoka jakość zostanie osiągnięta poprzez działania na kilku płaszczyznach:

- *określenie struktury organizacyjnej przedsięwzięcia, w której podział zadań i odpowiedzialności jest jednoznaczny i klarowny oraz zdefiniowane są procedury organizacyjne i komunikacyjne*
- *konsekwentne utrzymywanie właściwego klimatu wokół spraw jakości - każdy członek zespołu realizującego projekt musi zdawać sobie sprawę z rangi tych spraw*
- *przyjęcie standardowych wymagań na wytwarzane produkty (pośrednie i końcowe) i usługi*
- *określenie procedur kontrolnych, wykonywanych w celu sprawdzenia, czy produkty i usługi spełniają przyjęte wymagania*
- *wykonywanie procedur kontrolnych przez powołany do tego Zespół d.s. Zapewnienia Jakości."*

Powyższe zasady udało się upowszechnić i w dużej części wdrożyć.

Kolejnym krokiem jakiego podjął się Zespół Jakości było graficzne przedstawienie procesu produkcyjnego. W wyniku tych prac wyróżniono ścieżki głównych procesów oraz ich fazy.

W ramach ścieżki organizacji i zarządzania projektem zdefiniowano proces zarządzania projektem, zarządzania zmianami, zarządzania zmianami w oprogramowaniu oraz proces zarządzania jakością i dokumentacją. Każdy z tych procesów wpływa na pozostałe procesy.

Ścieżka opracowania systemu dzieli się na fazy:

- przygotowania i projektowania
- implementacji
- weryfikacji i walidacji
- testowania.

Ścieżka wdrożeń rozpoczyna się od opracowania strategii wdrożeń. Następne fazy, to opracowanie strategii szkoleń. Po wykonaniu tych faz nastąpiło opracowywanie materiałów wdrożeniowych, przygotowywanie szkoleń i wdrożeń (opracowywanie harmonogramów). Realizacja wdrożeń rozpoczyna się od wdrożeń pilotowych (zbierania doświadczeń). Ostatnia faza to wykonywanie wdrożeń masowych. Wdrożenia te rozpoczynają się po akceptacji systemu przez MPiPS.

Po zakończeniu procesu rozwoju oprogramowania następuje proces serwisu (utrzymania) oprogramowania.

Oczywiście w każdym projekcie cykl życia projektu będzie nieco inny. Graficzna prezentacja cyklu życia projektu ułatwiła zrozumienie idei Systemu Zapewnienia Jakości, konieczności dokumentowania ważnych procesów oraz ogromu pracy jaką należy wykonać aby system opisać.

Gdy zakończono analizę procesów jakie się dzieją w projekcie przystąpiono do tworzenia dokumentacji Systemu Zapewnienia Jakości, która miała opisać stworzony model produkcji oprogramowania. W lipcu 1998 doprowadzono do uzupełnienia i zatwierdzenia Księgi Jakości Projektu. Opracowano i zatwierdzono procedurę zarządzania systemem zapewnienia jakości (SZJ), instrukcje opracowywania dokumentów SZJ, procedurę weryfikacji dokumentów, procedurę szkoleń, procedurę zarządzania dokumentacją (biblioteka SZJ) itp. W ten sposób powstał szkielet systemu zapewnienia jakości, na którego bazie miały być tworzone pozostałe elementy systemu.

Zatrudniono osobę, która zaczęła opiekować się dokumentacją SZJ oraz powołano przedstawiciela jakości w zespole rozwoju i utrzymania systemu PULS.

Założono, że przedstawiciel jakości w każdym zespole będzie w 50 % pracować dla Zespołu Jakości i w 50 % będzie wykonywać normalne działania operacyjne swojego zespołu. Przedstawiciel jakości miał łączyć znajomość tematyki jakości z problematyką normalnych działań operacyjnych zespołu z którego pochodził. Okazało się, że przedstawicieli jakości w poszczególnych zespołach pochłonęła realizacja bieżących zadań, co spowodowało, że nie mieli oni już czasu na realizację zadań związanych z jakością. Z tego powodu zaniechano łączenia dwóch ról w jednej osobie i jedna osoba albo miała się zajmować działalnością operacyjną albo działalnością pomocniczą, jaką jest zapewnianie jakości.

Pomimo tych trudności wiele istotnych procedur instrukcji i formularzy powstało i zostało wdrożonych. Niestety nie udało się w pełni udokumentować cyklu życia projektu we wszystkich jego obszarach.

Równocześnie z opracowywaniem procedur, instrukcji, formularzy były prowadzone szkolenia z powstających dokumentów. Pierwszym szkoleniem, które odbyło się w czerwcu 1998 roku było szkolenie wprowadzające w tematykę systemu zapewnienia jakości. Na tym szkoleniu została zdefiniowana jakość w projekcie. Określono przyczyny wdrażania systemu zapewnienia jakości. Została dokonana prezentacja normy ISO-9001 jako wzorca według którego miał być wdrażany System Zapewnienia Jakości. Zaprezentowano strukturę organizacyjną zespołu jakości.

Następne szkolenia były organizowane sukcesywnie w miarę powstawania dokumentów systemowych. Nie opracowano planu szkoleń, tak jak założono. Nie udało się wykonać szkoleń z wszystkich dokumentów. Przyczyny były różne. Niektóre dokumenty zaraz po opracowaniu były już nieaktualne. Niestety okazało się, że niektóre dokumenty były "martwe" i niepotrzebne.

Kolejnym krokiem miało być wdrożenie auditów wewnętrznych. Działania związane z auditami rozpoczęto od przeszkolenia auditorów. Szkoleń nie udało się wykonać w założonych terminach. Aktualnie mamy trzech przeszkolonych przez BSI auditorów wewnętrznych. Dopiero w maju bieżącego roku opracowano założenia do auditów wewnętrznych i plan auditów wewnętrznych. Zostały wyróżnione dwa rodzaje auditów: audity zespołów i audity procesów. Przykładem auditu procesów będzie zaplanowany na koniec czerwca br. Audit umowy serwisowej z KUP. Audity będą wykonywane zgodnie z procedurami ogólnie stosowanymi w firmie ComputerLand w obszarach w których firma ComputerLand posiada certyfikat ISO-9001 (dawny ComputerLand bez nowo wchłoniętych firm, np. CSBI).

W zakresie działań korygujących i zapobiegawczych zostaną wykorzystane doświadczenia firmy ComputerLand, a w szczególności zostanie do tego celu użyta funkcjonująca już baza działań korygujących i zapobiegawczych zrealizowana przy pomocy Lotus Notes.

W strukturach projektu ALSO-PULS stworzono Radę Projektu. W skład Rady wchodził Dyrektor Projektu i podlegli mu kierownicy. Radę zwoływał dyrektor w celu wymiany informacji pomiędzy członkami Rady, oceny wyników pracy, rozwiązywania pojawiających się problemów. W szczególności Rada oceniała działania zespołu jakości i zlecała zespołowi wykonywanie tak zwanych projektów jakościowych. Celem tych projektów było porządkowanie najważniejszych spraw dla projektu. Przykładem projektów jakościowych było opracowanie systemu bezpieczeństwa wyników pracy oraz dostosowanie organizacji projektu do umowy serwisowej z KUP. Prace Rady Projektu nie były jednak klasycznym przeglądem kierowniczym.

Oprócz działań wynikających z planu wdrażania Systemu Zapewnienia Jakości Zespół Jakości zajmował się następującymi ważnymi dla jakości zadaniami:

- Realizacją projektów jakościowych definiowanych przez Radę Projektu
- Zatwierdzaniem wersji oprogramowania
- Koordynacją testów końcowych oprogramowania

- Opieką nad systemem OPUS, który wspomaga proces wytwarzania oprogramowania.

Zespół Jakości kontroluje formalnie poprawność procesu wytwarzania kolejnych wersji oprogramowania. Czynności kontrolne są wplecione w proces wytwórczy. W uproszczeniu proces dokonywania zmian w oprogramowaniu wygląda następująco:

1. Jest rejestrowany zgłaszany problem. Problem ten jest odpowiednio kategoryzowany.
2. Dla każdego zgłoszonego problemu jest opracowywana diagnoza.
3. Następnie dokonywana jest zmiana w oprogramowaniu, dokumentacji projektowej i użytkownika.
4. Każda zmiana jest weryfikowana i następnie testowana.
5. Przygotowywany jest pakiet aktualizacyjny oprogramowania.
6. Po przygotowaniu tegoż pakietu są wykonywane testy integracji według wcześniej przygotowanych planów.

Cały proces jest dokumentowany w dokumencie nazwanym "Specyfikacją zmian". Zatwierdzanie wersji polega na sprawdzeniu dokumentacji procesu (czy wszystkie czynności wykonano zgodnie z przyjętymi zasadami). Podczas zatwierdzania wykonywana jest również próbna instalacja oprogramowania. Jeżeli pakiet aktualizacyjny oprogramowania zawiera wszystkie elementy zgodnie z standardem i instaluje się, to jest formalnie zatwierdzany i przesyłany do klienta. Żadna wersja oprogramowania nie jest przekazywana MPiPS bez wcześniejszego jego zatwierdzenia.

Przed oddaniem oprogramowania do testu akceptacyjnego Zespół Jakości organizuje test końcowy. Celem takiego testu jest zbadanie stanu systemu według określonych kryteriów akceptacji i dyskwalifikacji oprogramowania. Kryteria te są surowsze niż kryteria określone w kontrakcie. W teście uczestniczą Specjaliści ds. Wdrożeń, którzy znają tematykę testowanych aplikacji. Wcześniej przygotowywane są plany testu. Prowadzone są z reguły trzy rodzaje testów:

- Test użytkownika polegający na powtórzeniu testów integracji
- Test aplikacji według planów testów opracowanych na podstawie dokumentacji użytkownika
- Test statystyczny polegający na dokładnym sprawdzeniu zgodności wylosowanej części projektu szczegółowego z oprogramowaniem

Każdy znaleziony błąd jest rejestrowany i kategoryzowany. Błędy są poprawiane i ponownie tester wykonuje retest.

Wyniki testu są na bieżąco obserwowane przez koordynatora testu końcowego, a na ich podstawie sporządza się zestawienia obrazujące postęp testowania, jego skuteczność i poziom jakości oprogramowania. W czasie testu końcowego MPiPS ma prawo wykonać inspekcję.

Po zakończeniu testu opracowywane są jego wyniki. Na podstawie tych wyników podejmowana jest decyzja o przystąpieniu do testu akceptacyjnego przeprowadzanego przez MPiPS.

Wcześniej opisano procesy dokonywania zmian w oprogramowaniu, zatwierdzania wersji oprogramowania, testowania. Realizacja tych czynności bez odpowiednich narzędzi byłaby praktycznie nie możliwa. Dlatego opracowano system OPUS, który pierwotnie miał służyć do rejestrowania błędów. Z czasem jego funkcjonalność była poszerzana i aktualnie jest on podstawowym narzędziem wspomagającym proces wytwarzania oprogramowania. Można śmiało powiedzieć, że w systemie OPUS zawarto wszystkie kluczowe procedury jakie wyróżniono w ścieżce opracowania oprogramowania. W skrócie system OPUS wspomaga proces wytwórczy w zakresie:

- Zarządzania wersjami oprogramowania
- Ewidencji zgłaszanych problemów
- Ewidencji zmian dokonywanych w oprogramowaniu (żadna zmiana nie może być wykonana poza systemem OPUS)
- Ewidencji zmian dokonywanych w dokumentacji projektowej i użytkownika
- Ewidencji wyników wykonanych czynności kontrolnych (weryfikacje, plany testów, wyniki testów)

- Tworzenia systemu pomocy kontekstowej
- Wspomagania wytwarzania pakietów aktualizacyjnych oprogramowania
- Kontroli jakości (zatwierdzania wersji oprogramowania)

Dzięki systemowi OPUS udało się zrównoleglić proces zatwierdzania wersji z procesem dokonywania zmian w oprogramowaniu. Poprzez system OPUS nad każdą zmianą pracuje autor zmiany (projektant, programista), weryfikator i tester (inny projektant lub programista), specjalista ds. zapewnienia jakości, a przed oddaniem oprogramowania do testu akceptacyjnego również specjalista ds. testów (wdrożeniowiec).

4. Bilans Zysków i Strat

Wdrażanie systemu zapewnienia jakości i systematyczne działania związane z jakością, w które byli zaangażowani pracownicy z wszystkich zespołów przyniosły wymierne efekty:

- Nastąpił znaczny postęp w realizacji kontraktu (oprogramowanie otrzymało pozytywny wynik testu akceptacyjnego, wdrożenia masowe są realizowane, jest realizowana umowa dotycząca serwisu oprogramowania)
- Oprogramowanie jest rozwijane w sposób systematyczny i kontrolowany
- Czynności kontrolne zostały wplecione w proces wytwórczy
- System Zapewnienia Jakości został prawie wdrożony
- Personel wie jak ważna jest sprawa zapewniania jakości na czym polega System Zapewnienia Jakości
- Powstała biblioteka dokumentacji
- Został opracowany system OPUS

Jest to zasługą całego zespołu projektu ALSO-PULS.

Oczywiście nie wszystko udało się zrealizować i nie zawsze wykonywano to w terminach. Nie jesteśmy zadowoleni z:

- Błędów w szacowaniu pracochłonności i planowaniu
- Ograniczonych zasobów do wykonywania zadań związanych z jakością
- Niezrealizowania do końca harmonogramu wdrażania systemu zapewnienia jakości (nie sprawdziła się idea przedstawicieli ds. zapewnienia jakości, nie udało się systematycznie szkolić pracowników, nie do końca wdrożono audity wewnętrzne, działania korygujące i zapobiegawcze nadal nie funkcjonują w sposób systemowy)

5. Projekt ALSO-PULS po zmianach organizacyjnych w ComputerLand

W firmie ComputerLand istnieją trzy sektory: przemysłowy, bankowo-finsowy i publiczny. Sektory te działają w 8 regionach ComputerLand. W ramach każdego sektora i regionów tworzone są Centra Kompetencyjne (CK), które mają dostarczać regionom odpowiednich produktów i usług. W sektorze publicznym zostało w kwietniu 1999 roku utworzone CK nazwane Działem Produkcji Oprogramowania (DPO). W skład DPO weszły zespoły realizacyjne wykonujące przedsięwzięcia informatyczne w ramach sektora publicznego między innymi projekt ALSO-PULS.

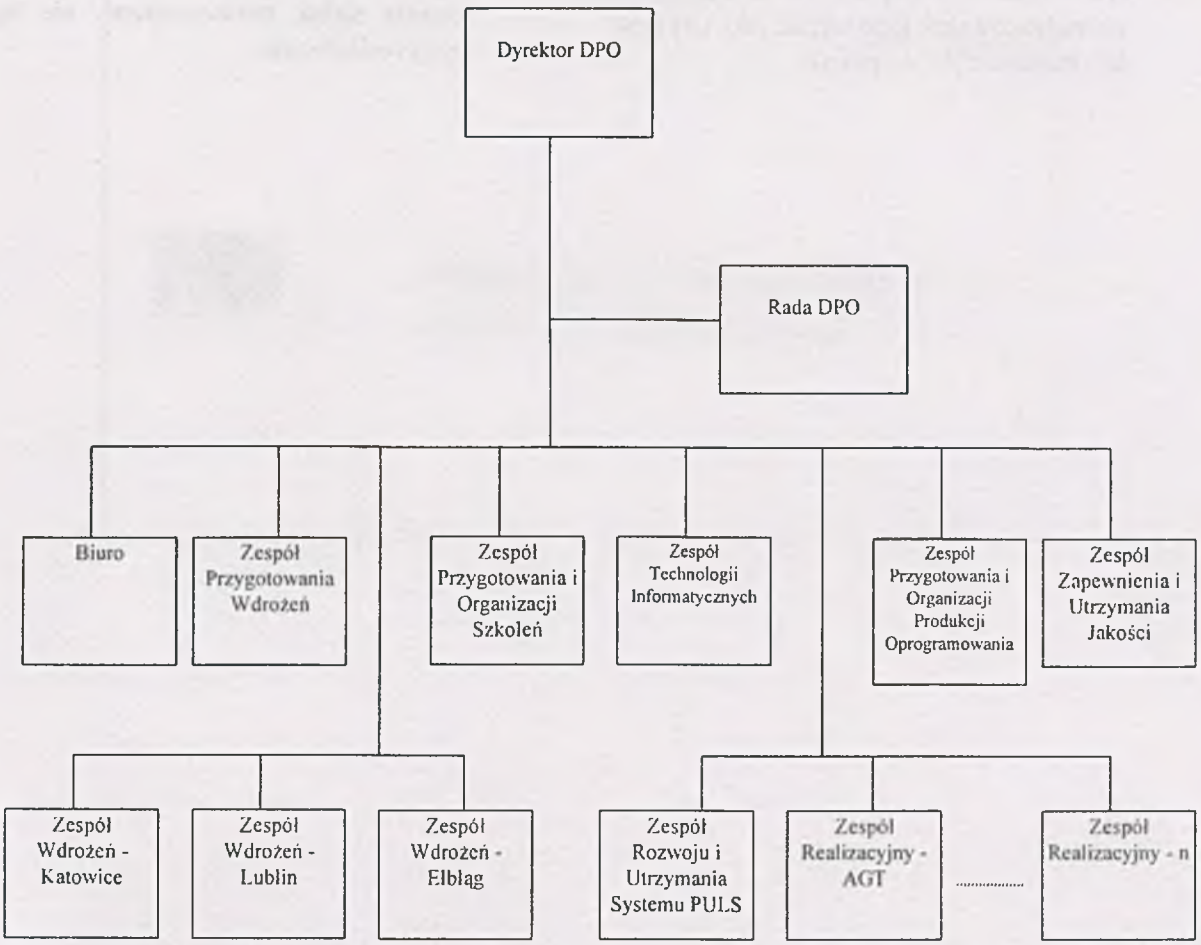
Dział Produkcji Oprogramowania dostarcza wysokiej jakości produkty i świadczy wysokiej jakości usługi, w szczególności w sposób zgodny z wymaganiami klienta wewnętrznego i zewnętrznego. Realizuje zadania i wytwarza produkty w ramach ustalonego budżetu i w zaplanowanych terminach.

Celem Działu Produkcji Oprogramowania (DPO) jest tworzenie zaplecza dla realizacji projektów informatycznych w ramach Sektora Publicznego w ComputerLand S.A. (CL). Dział Produkcji Oprogramowania odpowiada za realizację projektów informatycznych w warstwach:

- przygotowania założeń techniczno – ekonomicznych projektów
- analizy i weryfikacji wymagań użytkownika
- organizacji procesu produkcji oprogramowania
- zarządzania procesem produkcji oprogramowania

- opracowywania i serwisu oprogramowania
- opracowywania i przygotowywania systemu szkoleniowego
- organizacji i prowadzenia szkoleń (z wyjątkiem szkoleń masowych dla użytkowników)
- przygotowania procesu wdrożeń masowych (w tym przeprowadzanie wdrożeń pilotowych)
- świadczenia usług dodatkowych związanych z tworzonymi produktami.

W skład Działu Produkcji Oprogramowania wchodzą zespoły pomocnicze i realizacyjne. Zespoły pomocnicze wspomagają i kontrolują prace zespołów realizacyjnych. Zespoły pomocnicze zostały utworzone w oparciu o kadrę zaangażowaną dotychczas w projekt ALSO-PULS. Również doświadczenia Zespołu Jakości są przenoszone na cały Dział Produkcji Oprogramowania. Oto struktura organizacyjna DPO:

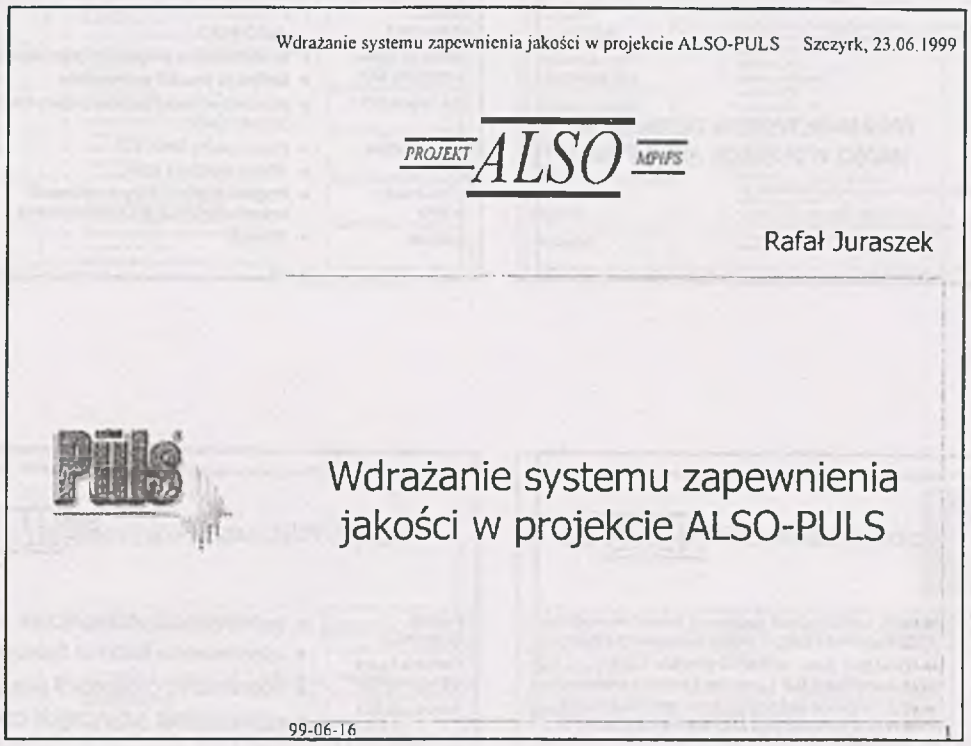


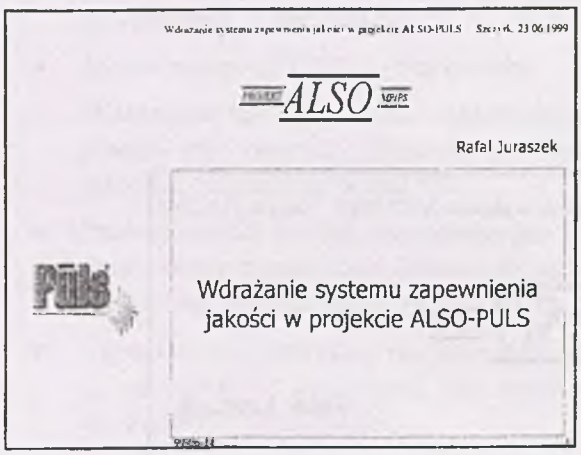
Wnioski

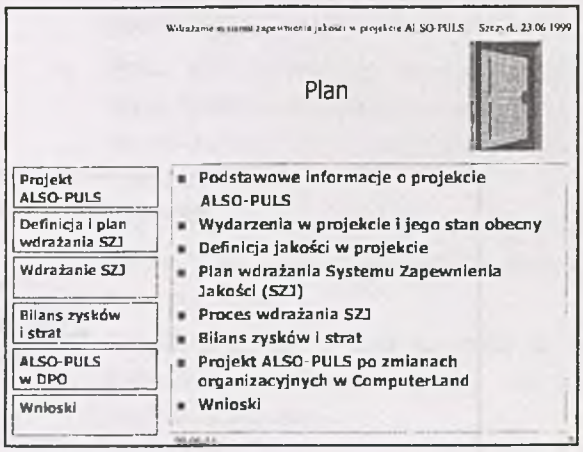
Na podstawie rocznego praktycznego doświadczenia z jakością w projekcie ALSO-PULS mogę udzielić paru prostych rad:

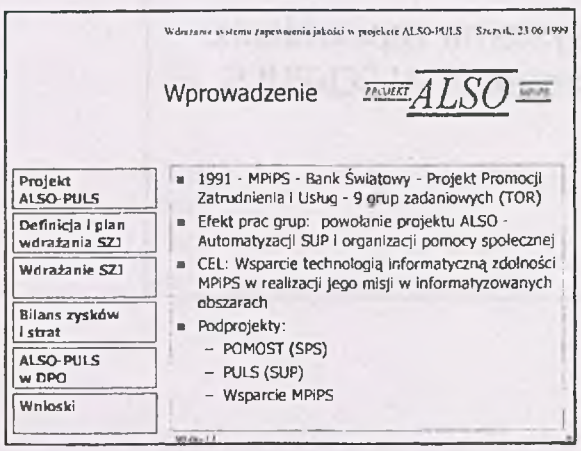
- Zanim zabierzecie się do pracy nad Systemem Zapewnienia Jakości należy odpowiedzieć na pytanie o cele tego systemu. Dowiedzieć się od kierownictwa jak ono rozumie ten system i jakie ma on mu przynieść korzyści. Im dokładniej zostaną zdefiniowane kryteria odbioru tego systemu, tym będzie mniej problemów. Najgorzej jest wtedy, gdy nie wiadomo co należy zrobić.
- Bez faktycznego poparcia kierownictwa (alokowania środków i zasobów) niewiele da się zrobić

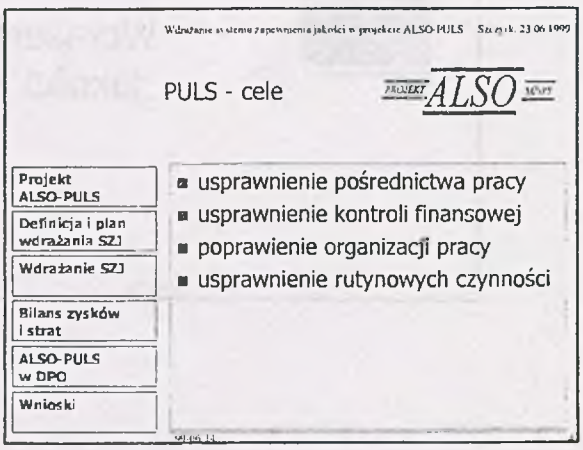
- Ludzie nie lubią być kontrolowani. Kontrole są potrzebne, lecz nie tędy droga. Należy pracować z ludźmi, pomagać, uświadamiać, a nie straszyć.
- Jakość zależy od każdego pracownika.
- Ważniejsze jest planowanie i zapewnienia jakości niż kontrola. Kontrola ma być tylko tak zwaną kropką nad "i".
- Procesy muszą być tak skonstruowane by umożliwiały zapewnienie jakości. Kontrola musi być wbudowana w proces.
- Kierownictwa zazwyczaj nie interesuje ile się napracowałeś lecz wynik jaki uzyskałeś. Podobnie jak w sporcie.
- Tylko systematyczne podejście do rozwiązywania problemów gwarantuje sukces (zweryfikowana koncepcja, plan i dopiero realizacja).
- Życie tak szybko się zmienia, że wdrażanie Systemu Zapewnienia Jakości nigdy się nie kończy i jest ciągłym dążeniem do doskonałości.
- Nie można przesadzić z biurokracją, gdyż ludzie będą tylko wypełniać dokumenty a nie pracować.
- Bez alokowania na stałe zasobów do wykonania zadań, harmonogramy nie będą nigdy realizowane.

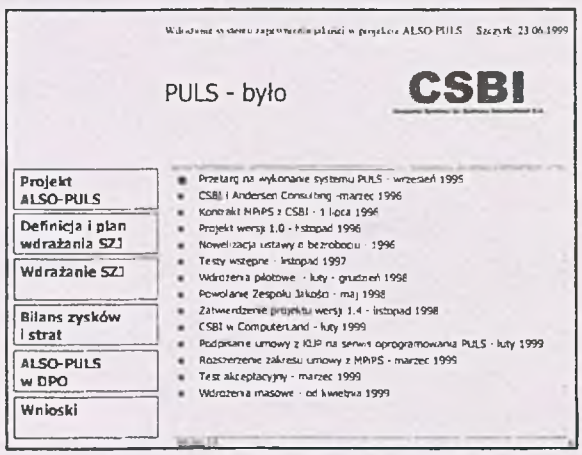

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
PROJEKT ALSO MPI/FS
Rafał Juraszek
Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS
99-06-16
1

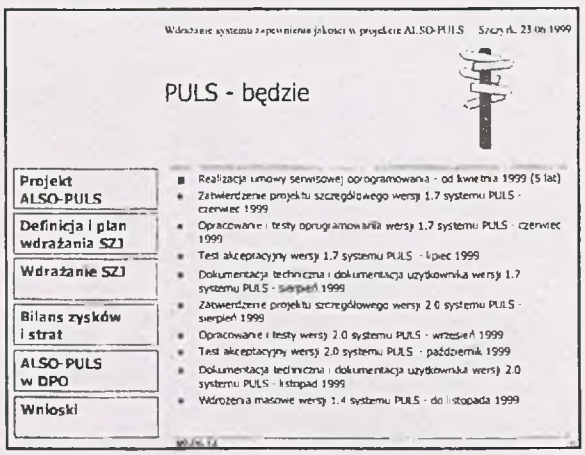

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
ALSO
Rafał Juraszek
PULS
Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS

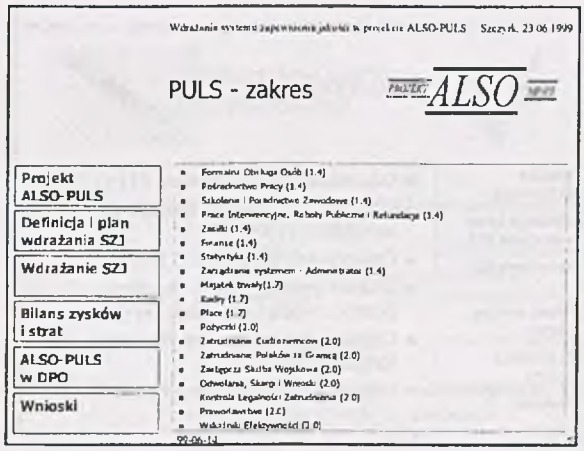

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
Plan
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski
Podstawowe informacje o projekcie ALSO-PULS
Wydarzenia w projekcie i jego stan obecny
Definicja jakości w projekcie
Plan wdrażania Systemu Zapewnienia Jakości (SZJ)
Proces wdrażania SZJ
Bilans zysków i strat
Projekt ALSO-PULS po zmianach organizacyjnych w ComputerLand
Wnioski

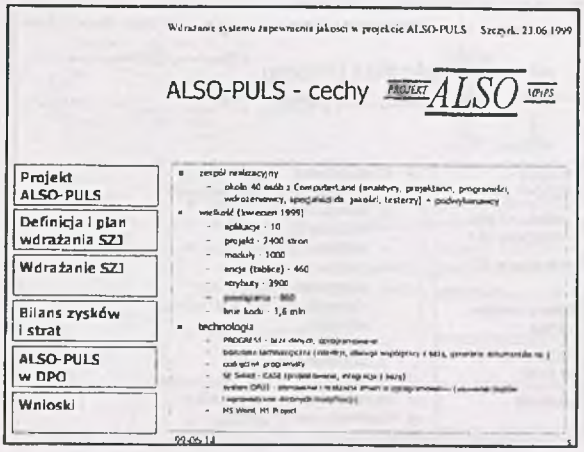

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
Wprowadzenie
ALSO
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski
1991 - MPiPS - Bank Światowy - Projekt Promocji Zatrudnienia i Usług - 9 grup zadaniowych (TOR)
Efekt prac grup: powołanie projektu ALSO - Automatyzacji SUP i organizacji pomocy społecznej
CEL: Wsparcie technologią informatyczną zdolności MPiPS w realizacji jego misji w informatyzowanych obszarach
Podprojekty:
– POMOST (SPS)
– PULS (SUP)
– Wsparcie MPiPS

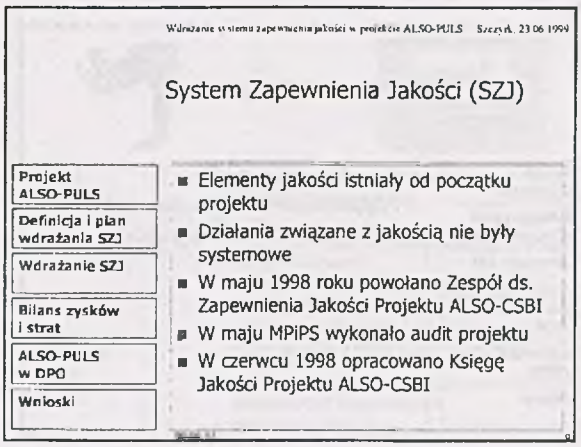

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
PULS - cele
ALSO
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski
usprawnienie pośrednictwa pracy
usprawnienie kontroli finansowej
poprawienie organizacji pracy
usprawnienie rutynowych czynności

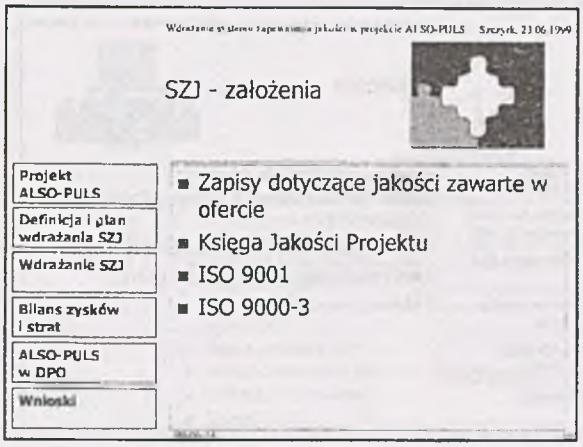

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
PULS - było
CSBI
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski
Przetarg na wykonanie systemu PULS - wrzesień 1995
CSBI i Andersen Consulting -marzec 1996
Kontrakt MPiPS z CSBI - 1 lipca 1996
Projekt wersji 1.0 - listopad 1996
Nowelizacja ustawy o bezrobociu - 1996
Testy wstępne - listopad 1997
Wdrożenia pilotowe - luty - grudzień 1998
Powołanie Zespołu Jakości - maj 1998
Zatwierdzenie projektu wersji 1.4 - listopad 1998
CSBI w ComputerLand - luty 1999
Podpisanie umowy z KUP na serwis oprogramowania PULS - luty 1999
Rozszerzenie zakresu umowy z MPiPS - marzec 1999
Test akceptacyjny - marzec 1999
Wdrożenia masowe - od kwietnia 1999

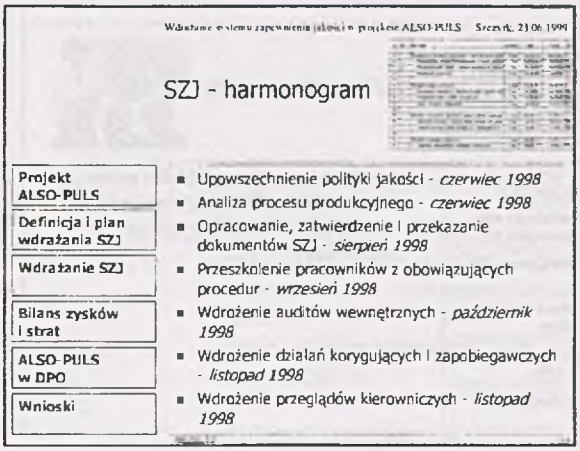

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
PULS - będzie
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski
Realizacja umowy serwisowej oprogramowania - od kwietnia 1999 (5 lat)
Zatwierdzenie projektu szczegółowego wersji 1.7 systemu PULS - czerwiec 1999
Opracowanie i testy oprogramowania wersji 1.7 systemu PULS - czerwiec 1999
Test akceptacyjny wersji 1.7 systemu PULS - lipiec 1999
Dokumentacja techniczna i dokumentacja użytkownika wersji 1.7 systemu PULS - sierpień 1999
Zatwierdzenie projektu szczegółowego wersji 2.0 systemu PULS - sierpień 1999
Opracowanie i testy wersji 2.0 systemu PULS - wrzesień 1999
Test akceptacyjny wersji 2.0 systemu PULS - październik 1999
Dokumentacja techniczna i dokumentacja użytkownika wersji 2.0 systemu PULS - listopad 1999
Wdrożenia masowe wersji 1.4 systemu PULS - do listopada 1999

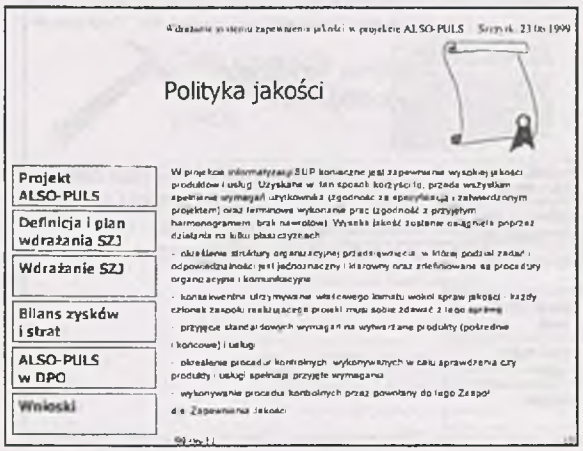

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
PULS - zakres
ALSO
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski
Formularz Obsługa Osób (1.4)
Pośrednictwo Pracy (1.4)
Szkolenie i Poradnictwo Zawodowe (1.4)
Prace Interwencyjne, Roboty Publiczne i Refundacja (1.4)
Zasiłki (1.4)
Świadczenia (1.4)
Statystyka (1.4)
Zarządzanie systemem - Administrator (1.4)
Majątek trwały(1.7)
Kasy (1.7)
Płace (1.7)
Pożyczki (2.0)
Zatrudnianie Cudzoziemców (2.0)
Zatrudnianie Polaków za Granicą (2.0)
Zastępcza Służba Wojskowa (2.0)
Odwołania, Skargi i Wnioski (2.0)
Kontrola Legalności Zatrudnienia (2.0)
Prawodawstwo (2.0)
Wskaźniki Efektywności (3.0)
99-06-14


Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
ALSO-PULS - cechy
ALSO
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski
zespół realizacyjny
- około 40 osób z ComputerLand (analitycy, projektanci, programiści, wdrożeniowcy, specjaliści ds. jakości, testerzy) + podwykonawcy
wielkość (kwiecień 1999)
- aplikacje - 10
- projekt - 3400 stron
- moduły - 1000
- encje (tabelce) - 460
- atrybuty - 3900
- [illegible]
- linie kodu - 1,6 mln
technologia
- PROGRESS - baza danych, oprogramowanie
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- MS Word, MS Project
99-06-14


Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
System Zapewnienia Jakości (SZJ)
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski
Elementy jakości istniały od początku projektu
Działania związane z jakością nie były systemowe
W maju 1998 roku powołano Zespół ds. Zapewnienia Jakości Projektu ALSO-CSBI
W maju MPiPS wykonało audit projektu
W czerwcu 1998 opracowano Księgę Jakości Projektu ALSO-CSBI


Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
SZJ - założenia
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski
Zapisy dotyczące jakości zawarte w ofercie
Księga Jakości Projektu
ISO 9001
ISO 9000-3


Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
SZJ - harmonogram
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski
Upowszechnienie polityki jakości - czerwiec 1998
Analiza procesu produkcyjnego - czerwiec 1998
Opracowanie, zatwierdzenie i przekazanie dokumentów SZJ - sierpień 1998
Przeszkolenie pracowników z obowiązujących procedur - wrzesień 1998
Wdrożenie auditów wewnętrznych - październik 1998
Wdrożenie działań korygujących i zapobiegawczych - listopad 1998
Wdrożenie przeglądów kierowniczych - listopad 1998


Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
Polityka jakości
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski
W projekcie informatycznym SUP konieczne jest zapewnienie wysokiej jakości produktów i usług. Uzyskane w ten sposób korzyści, przede wszystkim spełnienie wymagań użytkownika (zgodność ze specyfikacją i zatwierdzonym projektem) oraz terminowe wykonanie prac (zgodność z przyjętym harmonogramem, brak nakładów). Wysoką jakość zostanie osiągnięta poprzez działania na kilku płaszczyznach
- określenie struktury organizacyjnej przedsięwzięcia, w której podział zadań i odpowiedzialności jest jednoznaczny i klarowny oraz zdefiniowane są procedury organizacyjne i komunikacyjne
- konsekwentne utrzymywanie właściwego kontaktu wokół spraw jakości - każdy członek zespołu realizującego projekt musi sobie zdawać z tego sprawę
- przyjęcie standardowych wymagań na wytwarzane produkty (pośrednie i końcowe) i usługi
- określenie procedur kontrolnych wykonywanych w celu sprawdzenia czy produkty i usługi spełniają przyjęte wymagania
- wykonywanie procedur kontrolnych przez powołany do tego Zespół d.s. Zapewnienia Jakości

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999

Analiza procesu produkcyjnego

- Projekt ALSO-PULS
- Definicja i plan wdrażania SZJ
- Wdrażanie SZJ
- Bilans zysków i strat
- ALSO-PULS w DPO
- Wnioski

- Ścieżki procesów:
  - organizacja i zarządzanie projektem
  - opracowanie systemu PULS
  - wdrożenie systemu PULS
  - utrzymanie systemu PULS
- Fazy projektu:
  - przygotowanie i projektowanie
  - implementacja, weryfikacja i walidacja
  - testowanie
  - wdrażanie
  - utrzymanie
- Wyróżnienie procesów w każdej ze ścieżek
- Oznaczanie i opisywanie procesów
- Uzgadnianie powiązań z MPiPS

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999

Dokumentacja SZJ

- Projekt ALSO-PULS
- Definicja i plan wdrażania SZJ
- Wdrażanie SZJ
- Bilans zysków i strat
- ALSO-PULS w DPO
- Wnioski

- Uzupełnienie i zatwierdzenie KJ (+)
- Biblioteka SZJ (zasady, procedura, wdrożenie) (+)
- Opracowanie szkieletu SZJ (+)
- Budowa zespołu jakości - asystent, przedstawiciele i specjaliści (+/-)
- Opracowywanie procedur, instrukcji, formularzy (+/-)
- Przegląd i zatwierdzanie dokumentów (+)

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999

Szkolenia

- Projekt ALSO-PULS
- Definicja i plan wdrażania SZJ
- Wdrażanie SZJ
- Bilans zysków i strat
- ALSO-PULS w DPO
- Wnioski

- Przeprowadzenie szkolenia wprowadzającego z Systemu Zapewnienia Jakości (+)
- Przygotowanie planu szkoleń (+/-)
- Przeszkolenie pracowników (+/-)

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999

Audity wewnętrzne

- Projekt ALSO-PULS
- Definicja i plan wdrażania SZJ
- Wdrażanie SZJ
- Bilans zysków i strat
- ALSO-PULS w DPO
- Wnioski

- Przeszkolenie auditorów (+)
- Przygotowanie planów auditów (+)
- Audity wewnętrzne według planów (+/-)

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999

Działania korygujące i zapobiegawcze

- Projekt ALSO-PULS
- Definicja i plan wdrażania SZJ
- Wdrażanie SZJ
- Bilans zysków i strat
- ALSO-PULS w DPO
- Wnioski

- Opracowanie procedury w zakresie działań korygujących i zapobiegawczych (+/-)
- Wdrożenie procedury działań korygujących i zapobiegawczych (-)

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999

Przeglądy kierownicze

- Projekt ALSO-PULS
- Definicja i plan wdrażania SZJ
- Wdrażanie SZJ
- Bilans zysków i strat
- ALSO-PULS w DPO
- Wnioski

- Organizacja i przeprowadzanie przeglądów kierowniczych - Rada Projektu (+/-)
- Wypełnianie postanowień przeglądów kierowniczych (+/-)

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999

Zatwierdzanie wersji oprogramowania

- Projekt ALSO-PULS
- Definicja i plan wdrażania SZJ
- Wdrażanie SZJ
- Bilans zysków i strat
- ALSO-PULS w DPO
- Wnioski

- Wersjonowanie (a.b.c.d)
- Etapy procesu dokonywania zmian:
  - zgłoszony problem
  - diagnoza
  - zmiana w oprogramowaniu i dokumentacji
  - weryfikacja
  - test
  - pakiet aktualizacyjny
  - test integracji
- Dokumentowanie procesu -> specyfikacja zmian
- Kontrola, próbna instalacja, zatwierdzenie

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999

Testy końcowe

- Projekt ALSO-PULS
- Definicja i plan wdrażania SZJ
- Wdrażanie SZJ
- Bilans zysków i strat
- ALSO-PULS w DPO
- Wnioski

- Cele testu
- Organizacja testu (zasoby, przygotowania do testów, szkolenia, narzędzia, plany testów)
- Zasady i przebieg testu
  - rodzaje testów (użytkownika, aplikacji, statystyczny)
  - rejestrowanie błędów (kryteria, kategorie)
  - poprawianie błędów
- Monitorowanie testu
  - bilans otwarcia
  - pomiary postępu prac, jakości
- Inspekcje Nabywcy
- Wyniki testu

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999

System OPUS

- Projekt ALSO-PULS
- Definicja i plan wdrażania SZJ
- Wdrażanie SZJ
- Bilans zysków i strat
- ALSO-PULS w DPO
- Wnioski

- Autorskie narzędzie wspomagające proces wytwarzania oprogramowania
- Funkcje:
  - zarządzanie wersjami oprogramowania
  - ewidencja problemów
  - ewidencja zmian w oprogramowaniu
  - wyniki weryfikacji, plany testów, wyniki testów
  - tworzenie systemu pomocy kontekstowej
  - ewidencja zmian w dokumentacji projektowej i użytkownika
  - wspomaganie wytwarzania pakietów aktualizacyjnych oprogramowania
  - zatwierdzanie wersji

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999

Zyski

- Projekt ALSO-PULS
- Definicja i plan wdrażania SZJ
- Wdrażanie SZJ
- Bilans zysków i strat
- ALSO-PULS w DPO
- Wnioski

- Postęp w realizacji kontraktu:
  - pozytywny wynik testu akceptacyjnego
  - trwające wdrożenia masowe
  - serwis oprogramowania
- Systematyczne i kontrolowane dokonywanie zmian w oprogramowaniu
- Planowe testowanie wplecione w proces wytwórczy
- Prawie wdrożony SZJ
- Wzrost świadomości zespołu w zakresie jakości
- Biblioteka dokumentacji
- System OPUS

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999

Straty

- Projekt ALSO-PULS
- Definicja i plan wdrażania SZJ
- Wdrażanie SZJ
- Bilans zysków i strat
- ALSO-PULS w DPO
- Wnioski

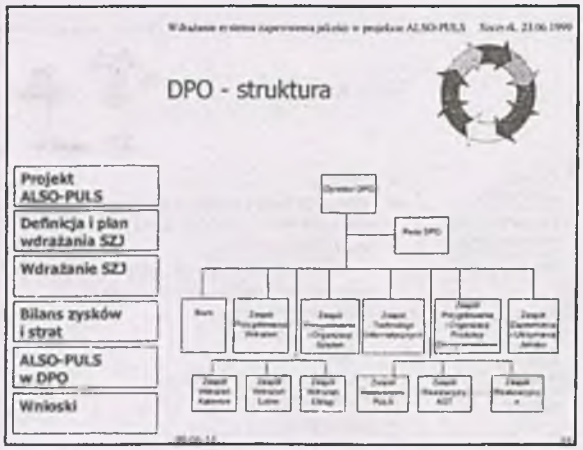
- Źle oszacowano pracochłonność zadań
- Ograniczone zasoby
- Nie udało się w pełni i w terminie wykonać harmonogramu wdrażania SZJ:
  - przedstawiciele ds. zapewnienia jakości
  - szkolenia
  - audity, działania korygujące i zapobiegawcze

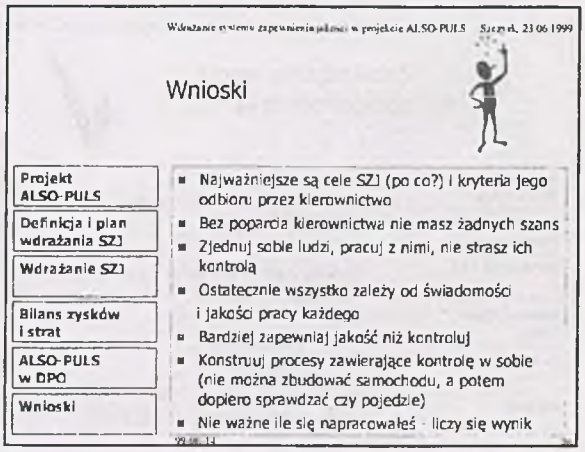
Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999

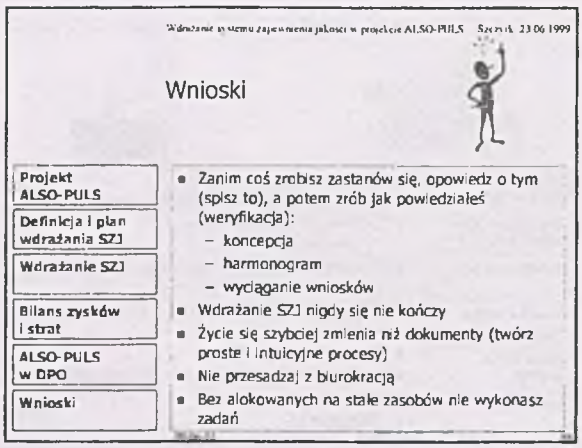
Dział Produkcji Oprogramowania (DPO)

- Projekt ALSO-PULS
- Definicja i plan wdrażania SZJ
- Wdrażanie SZJ
- Bilans zysków i strat
- ALSO-PULS w DPO
- Wnioski

- Dział Produkcji Oprogramowania dostarcza wysokiej jakości produkty i świadczy wysokiej jakości usługi, w sposób zgodny z wymaganiami klienta wewnętrznego i zewnętrznego. Realizuje zadania i wytwarza produkty w ramach ustalonego budżetu i w zaplanowanych terminach.
- Celem Działu Produkcji Oprogramowania (DPO) jest tworzenie zaplecza dla realizacji projektów informatycznych w ramach Sektora Publicznego w Computerland SA (CL). Dział Produkcji Oprogramowania odpowiada za realizację projektów informatycznych w warstwach:
  - [illegible]


Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
DPO - struktura
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski


Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
Wnioski
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski
Najważniejsze są cele SZJ (po co?) i kryteria jego odbioru przez kierownictwo
Bez poparcia kierownictwa nie masz żadnych szans
Zjednuj sobie ludzi, pracuj z nimi, nie strasz ich kontrolą
Ostatecznie wszystko zależy od świadomości i jakości pracy każdego
Bardziej zapewniaj jakość niż kontroluj
Konstruuj procesy zawierające kontrolę w sobie (nie można zbudować samochodu, a potem dopiero sprawdzać czy pojedzie)
Nie ważne ile się napracowałeś - liczy się wynik

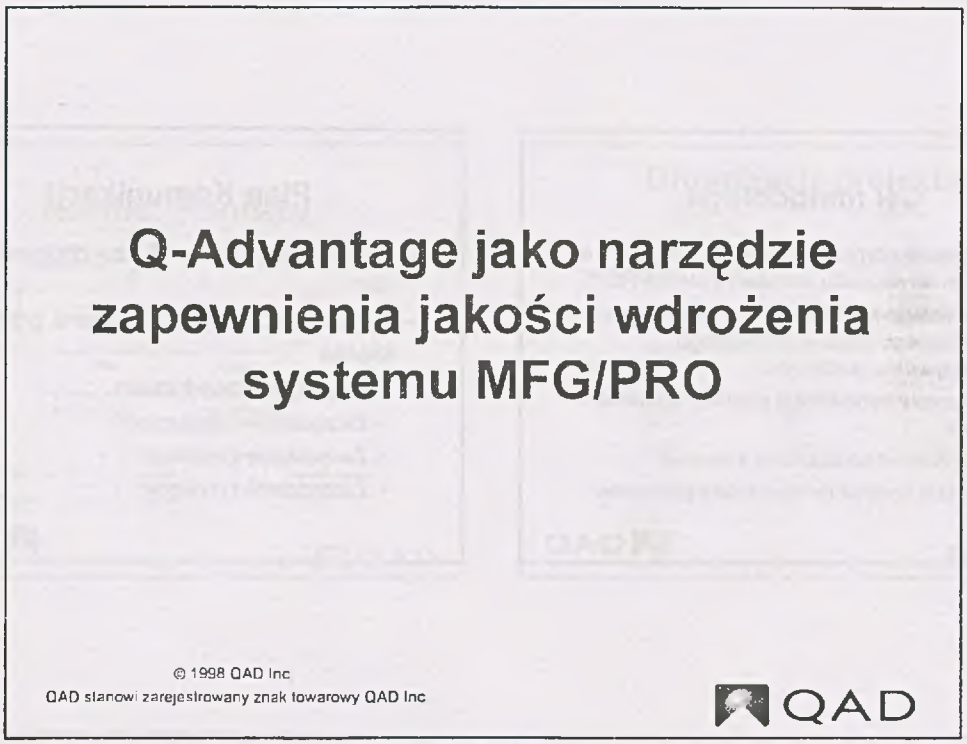

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
Wnioski
Projekt ALSO-PULS
Definicja i plan wdrażania SZJ
Wdrażanie SZJ
Bilans zysków i strat
ALSO-PULS w DPO
Wnioski
Zanim coś zrobisz zastanów się, opowiedz o tym (spisz to), a potem zrób jak powiedziałeś (weryfikacja):
– koncepcja
– harmonogram
– wyciąganie wniosków
Wdrażanie SZJ nigdy się nie kończy
Życie się szybciej zmienia niż dokumenty (twórz proste i intuicyjne procesy)
Nie przesadzaj z biurokracją
Bez alokowanych na stałe zasobów nie wykonasz zadań

Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS Szczyrk, 23.06.1999
PROJEKT ALSO
PULS

Tomasz Koniecko

QAD Polska

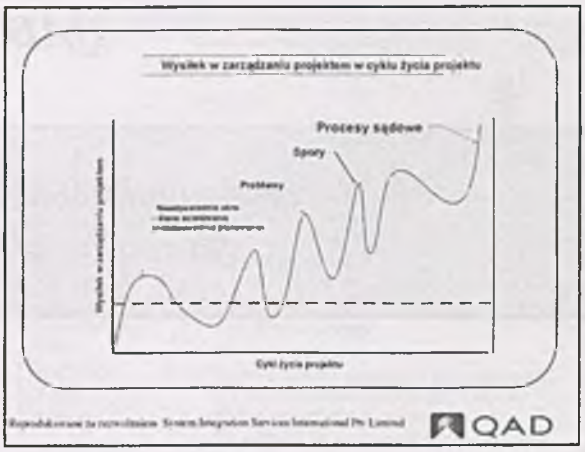
Q-Advantage jako narzędzie zapewnienia jakości wdrożenia systemu MFG/PRO



Polityka zapewnienia jakości

"Funkcją systemu zapewnienia jakości będzie zagwarantowanie realizacji celów projektu poprzez wdrożenie i konsekwentne stosowanie procesów i procedur systemu Q-Advantage opisanych w planach projektu i zatwierdzonych przez Komitet Sterujący Wdrożeniem."

QAD



QAD

Cel metodologii

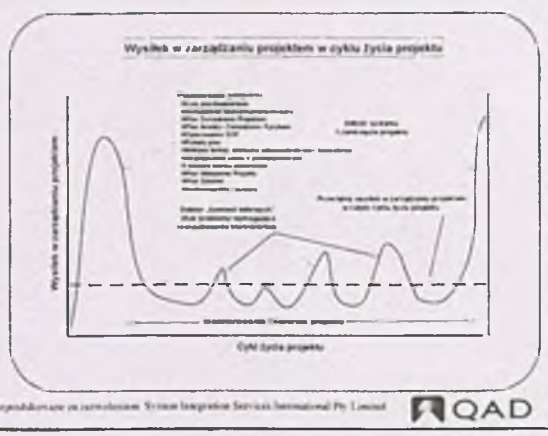
- Osiągnięcie porozumienia między klientem a QAD co do sposobu wdrożenia MFG/PRO™
- Minimalizacja konfliktów poprzez stosowanie uzgodnionego procesu wczesnego rozwiązywania problemów
- Sprzyjanie elastyczności poprzez spójność działań
- Danie możliwości działania klientowi
- Narzędzie kontroli do certyfikacji partnerów

QAD

Plan Komunikacji

- Określanie oczekiwań i zarządzanie nimi
- Regularne oceny dokonywane przez klienta
- Dyscyplina proceduralna
  - Zarządzanie wdrożeniem
  - Zarządzanie zmianami
  - Zarządzanie ryzykiem

QAD



QAD

Kluczowe cechy

- Własność
  - Komitet Sterujący
  - Organizacja projektu
  - Funkcje, zakresy odpowiedzialności i kompetencje
- Dokumentacja

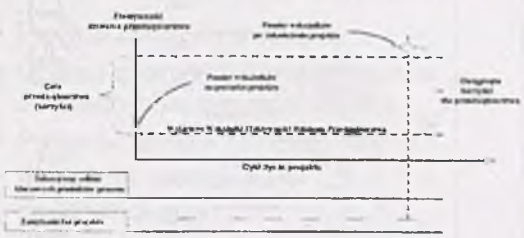
QAD

Elementy funkcjonalne

- Zakres
  - Struktura organizacji prac i słownik
  - Lista produktów faz wdrożenia
- Wyjściowe Wskaźniki Efektywności (WWE)
  - WWE Działania Przedsiębiorstwa
  - WWE Projektu
- Kryteria odbioru

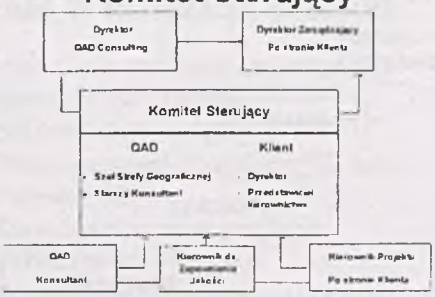
QAD

Proces odbioru



QAD

Komitet Sterujący



QAD

Organizacja projektu

QAD

Rola Kierownika Projektu

Kierownik Projektu ponosi odpowiedzialność za realizację projektu zgodnie z jego celami, w ustalonym terminie i zgodnie z budżetem.

QAD

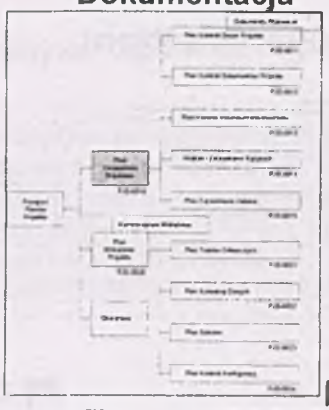
Rola Konsultanta QAD

Stanowi jedyny punkt kontaktu QAD w procesie wdrażania modułów MFG/PRO.

- Bezpośrednio kieruje Zespołem QAD ds. Projektu
- Ściśle współpracuje z Kierownikiem Projektu ze strony klienta
- Składa kierownictwu sprawozdania ze stanu zaangażowania

QAD

Dokumentacja



Jak to robi QAD

Plan Zarządzania Projektem (PZP)

- Proces zarządzania projektem
- Struktura organizacyjna
- Funkcje, zakresy odpowiedzialności i kompetencje
- Oceny i zatwierdzenia
- Przekazywanie problemów do rozwiązania na wyższym szczeblu

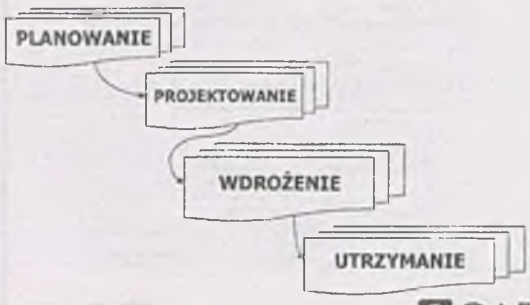
QAD

Plan Wdrożenia Projektu (PWP)

- Strategia wdrożenia
- Wyjściowe wskaźniki efektywności projektu
- Struktura organizacji prac, oszacowania i budżety
- Harmonogram projektu
- Zestawienie kosztów osobowych
- Matryca zakresów odpowiedzialności
- Sprawozdawczość projektu

QAD

Wdrożenie: podejście modułowe



QAD

Faza 0: Określenie potrzeb przedsiębiorstwa

- Udokumentowanie wyjściowych procedur operacyjnych przedsiębiorstwa
- Uzgodnienie kluczowych mierników efektywności
- Określenie celów przedsiębiorstwa
- Ustalenie organizacji pracy

QAD

Faza 1: Ustalenia i planowanie

- Określenie zakresu prac
- Uzyskanie zatwierdzenia przez klienta
  - Ról
  - Zakresów odpowiedzialności
  - „Kamieni milowych"
  - Produktów poszczególnych faz
- Powołanie podstawowego zespołu ds. projektu

QAD

Faza 2: Prototyp

- Rozwinięcie wniosków kierownictwa
- Przetestowanie dotychczasowych i nowych procesów stosowanych w przedsiębiorstwie pod względem
  - trwałości
  - efektywności
- Określenie zmian w procedurach przedsiębiorstwa
- Określenie modyfikacji oprogramowania

QAD

Faza 3: Przygotowanie systemu

- Przygotowania do wdrożenia
- Ocena i udokumentowanie procesów stosowanych w przedsiębiorstwie
- Przeprowadzenie konwersji danych
- Finalizacja dostosowania oprogramowania

QAD

Faza 4: Realizacja pilotowa

Testowanie i weryfikacja ostatecznego systemu przez Kluczowych Użytkowników

- Testowanie efektywności i czasów reakcji
- Formalny odbiór
- Opracowanie strategii wdrożenia
- Finalizacja Planu Przejścia
- Opracowanie Planu Serwisu i Pomocy

QAD

Faza 5: Wdrożenie

- Szkolenie użytkowników
- Wdrożenie Planów Przejścia oraz Serwisu i Pomocy
- Ocena gotowości

Przejście na nowy system

QAD

Faza 6: Zamknięcie projektu

Weryfikacja osiągnięcia celów określonych względem Wyjściowych Wskaźników Efektywności

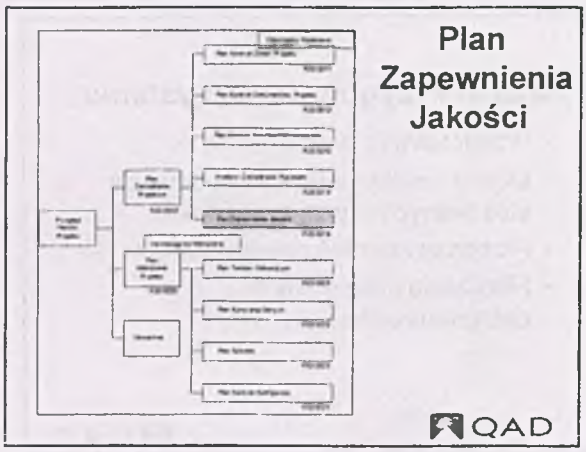
- Działanie przedsiębiorstwa
- Projekt

QAD

Faza 7: Serwis

Wdrożenie Planu Serwisu i Pomocy

QAD



Strategia

Celem strategii zapewnienia jakości jest:

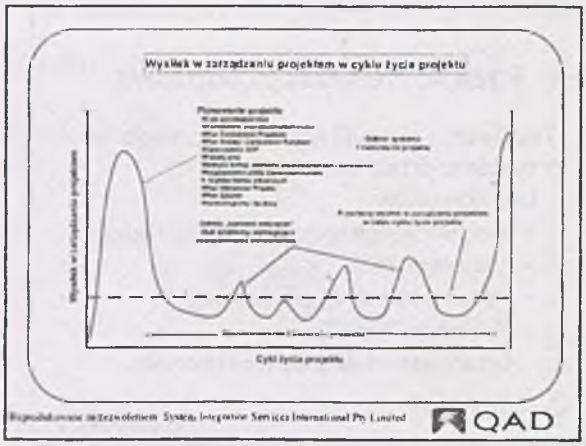
1. Określenie Krytycznych Czynników Powodzenia
2. Zapoznanie uczestników projektu z założeniami Planów Projektu
3. Przeprowadzenie okresowych kontroli jakości
4. Przeprowadzenie działań korygujących
5. Przekazywanie Komitetowi Sterującemu informacji niezbędnych do podejmowania decyzji

QAD

Zadania Kierownika ds. Zapewnienia Jakości

1. Zapewnienie wdrożenia Strategii Zapewnienia Jakości
2. Upowszechnianie zasad metodyki Q-Advantage
3. Przeprowadzanie kontroli jakościowych
4. Informowanie Kierownika Projektu o zauważonych niezgodnościach
5. Ustalanie w porozumieniu z Kierownikiem Projektu odpowiednich działań korygujących
6. Składanie okresowych sprawozdań Komitetowi Sterującemu na temat zgodności z Q-Advantage

QAD



Borys Stokalski

InfoViDE

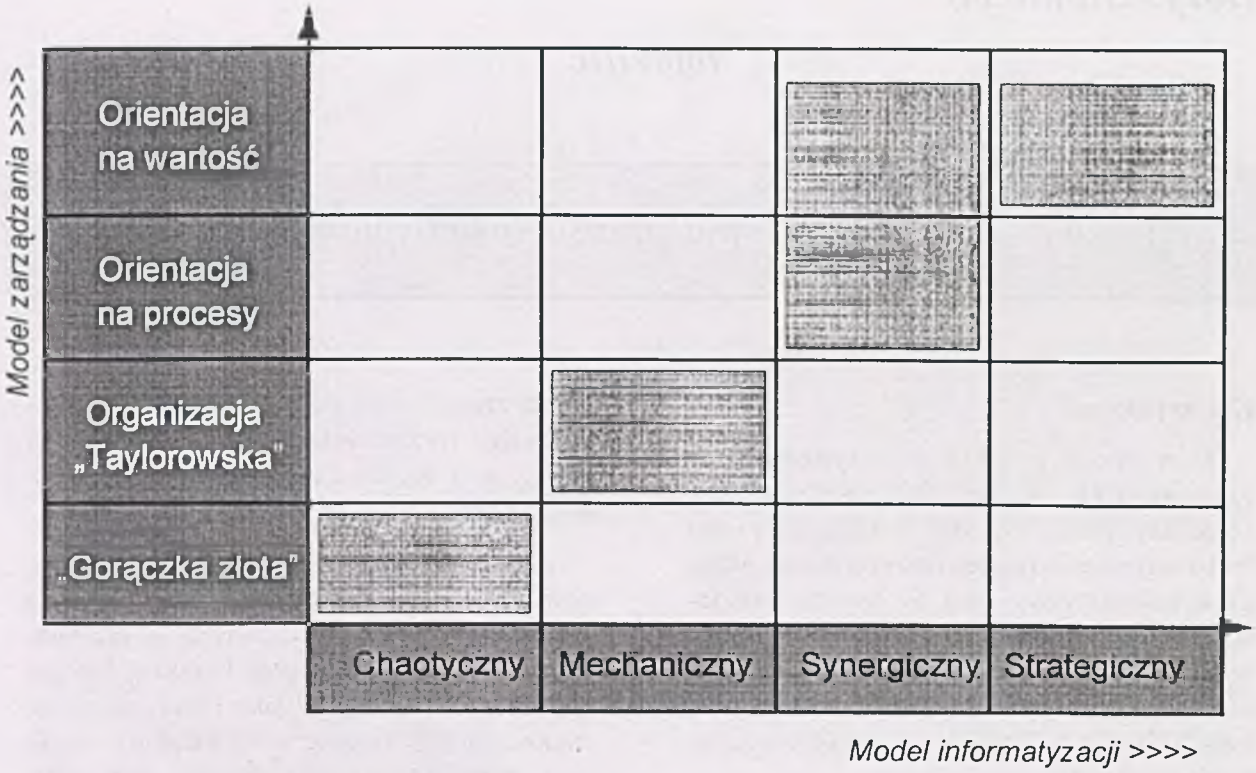
Zliczyć do trzech – analiza opłacalności inwestycji informatycznych

Kto to policzy?

Mówienie o projektowaniu systemów informatycznych w sposób uwzględniający konkretny bilans kosztów i korzyści często budzi wzruszenie ramion nawet u doświadczonych informatyków. Jest to swoiste świadectwo bezradności naszej branży wobec wyzwania jakim jest analiza opłacalności przedsięwzięć informatycznych, a zwłaszcza kwantyfikacja korzyści wypływających z informatyzacji.

Problem kwantyfikacji korzyści można uznawać za akademicki tak długo, jak poruszamy się w świecie w miarę oczywistych potrzeb niedużych grup użytkowników i stosunkowo mało złożonych, spójnych wewnętrznie rozwiązań. Niestety, skala przedsięwzięć realizowanych w naszym kraju coraz wyraźniej wykracza poza ten prosty świat. Sfera potrzeb obejmuje różnorodne strategie i cele biznesowe, które – jeśli rozpatrujemy je niezależnie – mogą być źródłem niespójnych bądź nawet sprzecznych wymagań. Sfera rozwiązań, to skomplikowane systemy informatyczne, o rosnącej złożoności technicznej i funkcjonalnej. Informatyzacja to dziś proces kapitałochłonny, jego koszty to setki tysięcy, a coraz częściej i miliony dolarów rocznie. Poważne traktowanie problemu analizy opłacalności inwestycji informatycznych, i nie poprzestawanie na stwierdzeniu, że „tego się nie da zrobić" staje się dziś jednym z podstawowych wyznaczników odpowiedzialnego traktowania swoich zadań przez menedżerów informatyki.

Analiza opłacalności wymaga jednak środowiska w którym możliwa jest kontrola finansowania zadań związanych z informatyzacją. Oznacza to jawność kosztów, i to nie tych które obserwujemy jako koszty *de-facto*, stanowiące nie zawsze mile widziany wynik prac nad rozwojem systemów informatycznych. Koszty o których mowa to świadomie planowany składnik budżetów organizacji. W pierwszej kolejności chodzi o budżety obrazujące plan działania operacyjnego, w których istotnym składnikiem są koszty eksploatacji oraz modyfikacji systemów. Równie ważne miejsce alokacji kosztów związanych z technologią informatyczną to budżety związane ze strategicznymi inicjatywami organizacji - restrukturyzacją, wprowadzeniem nowych produktów, rozwojem ważnych funkcji gospodarczych, takich jak marketing czy nowoczesne mechanizmy zarządzania. Krótko mówiąc, organizacje poważnie myślące o stosowaniu nowoczesnych rozwiązań informatycznych muszą przejść niełatwą drogę od informatyzacji ad-hoc, po informatyzacji rozumianej jako jeden ze strategicznych procesów inwestycyjnych.



Informatyzacja w stylu wolnym

Główną cechą informatyzacji chaotycznej jest brak widoczności tego nurtu inwestycyjnego na poziomie kierownictwa organizacji oraz towarzyszący temu brak synchronizacji inicjatyw „informatyzacyjnych". Działania inicjowane są one w sposób przypadkowy przez osoby mające silną pozycję w organizacji i wierzące w to, że komputer powinien być w nowocześnie zorganizowanej firmie używany.

Bilans opłacalności takich przedsięwzięć rzadko bywa jednoznacznie pozytywny. Brak zrozumienia dla ograniczeń technologii, trudności związanych z procesem wdrożenia, kosztów wytworzenia, a zwłaszcza kosztów utrzymania przy życiu raz wytworzonego rozwiązania powodują, że z czasem taka informatyzacja grzęźnie w mnożących się problemach. Część rozwiązań okazuje się bezużytecznych, te które działają coraz trudniej adaptować do zmian w otoczeniu biznesowym. Jeśli początkowo istnieje dobre porozumienie pomiędzy kadrą informatyczną a biznesową, z czasem, na skutek trudności w zaspokajaniu potrzeb użytkowników relacje te stają się coraz trudniejsze.

W tej sytuacji często pojawia się chęć do ujawnienia całości kosztów związanych z technologią informatyczną, w sposób umożliwiający centralne planowanie i kontrolę tych wydatków przez kierownictwo. Ponieważ zaś planowanie wymaga jakiegoś modelu odniesienia, modelem tym staje się nierzadko struktura organizacyjna firmy. Rozpoczyna się etap informatyzacji który można nazwać informatyzacją biurokratyczną, lub mechaniczną.

Komputeryzacja zadekretowana czyli podejście mechaniczne

Podejście to polega na systematycznej automatyzacji funkcji biznesowych. Typowy sposób myślenia przy tego rodzaju podejściu do informatyzacji wygląda mniej więcej tak: Jesteśmy firmą produkcyjną, stawiamy na nowoczesność i komputery, *ergo* musimy wdrożyć system lub systemy obsługujące księgowość, produkcję, magazyn oraz finanse i sprzedaż. Podstawowym celem jest to, aby funkcje te realizowane były od teraz „komputerowo". No i najważniejsze jest to aby system był „zintegrowany".

Główną korzyścią jaka jest osiągana w tym scenariuszu jest integracja danych w ramach

wdrożonych modułów wybranego systemu. Poprawa efektywności procesów biznesowych może lecz nie musi nastąpić, jako że koncentrujemy się na wyizolowanych fragmentach działania organizacji, próbując niezależnie uwzględniać cele i priorytety każdego z nich. To zaś nierzadko jest źródłem sprzecznych wymagań i sporów kompetencyjnych – dane wprowadzane w produkcji nie satysfakcjonują księgowości, sprzedaż potrzebuje innego zakresu danych na temat klientów niż finanse, raporty kontrolingowe nie satysfakcjonują kierownictwa, nie wiadomo jak podzielić obowiązki związane z rejestracją i przetwarzaniem informacji pomiędzy pracowników różnych pionów funkcjonalnych.

Sam proces wdrażania rozwiązań również może obfitować w problemy związane z koniecznością znaczących zmian w już wdrożonych modułach na skutek rozpoczęcia wdrożenia kolejnych. Wreszcie, następuje w znacznej mierze zakonserwowanie pewnej organizacji pracy.

W rezultacie bilans ponoszonych kosztów i uzyskiwanych korzyści może okazać się wielce problematyczny, co najmniej tak długo, jak fakt posiadania zintegrowanych, „uspójnionych" danych w skali całej organizacji nie zostanie wykorzystany w bardziej strategicznych zastosowaniach informatyki – na przykład w systemach wspomagania decyzji.

Władza dla ludu, czyli podejście synergiczne

Informatyzacja biurokratyczna w niewielu wypadkach daje efekty satysfakcjonujące kierownictwo. Korzyści niewspółmierne do wydatków, oraz spory i dyskusje dotyczące skuteczności działania informatyków, w połączeniu z wdrażaniem nowoczesnych zasad kontroli efektywności owocują zmianą ośrodków podejmowania decyzji związanych z informatyzacją. Autorem i dysponentem budżetów stają się osoby odpowiedzialne za kluczowe procesy gospodarcze w organizacji – nadchodzi era synergii inwestycji informatycznych.

Podejście synergiczne ukierunkowane jest na usprawnianie procesów gospodarczych – zwiększanie ich kosztów, poprawę jakości, skrócenie cykli obsługi podstawowych zdarzeń gospodarczych. Wyrasta ono wprost z potrzeb użytkowników i wymaga dostrzeżenia przez nich procesu gospodarczego jako mechanizmu generowania wartości interesujących klientów organizacji. Myślimy o „łańcuchu wartości" łączącym zadania realizowane przez różne fragmenty „funkcjonalne" organizacji – stąd krok do traktowania technologii informatycznej jako środka pozwalającego usunąć te elementy procesu, które generują koszty nie tworząc wartości dla klienta, środka skracającego czas realizacji podstawowych cykli biznesowych, korzystnie wpływających na jakość.

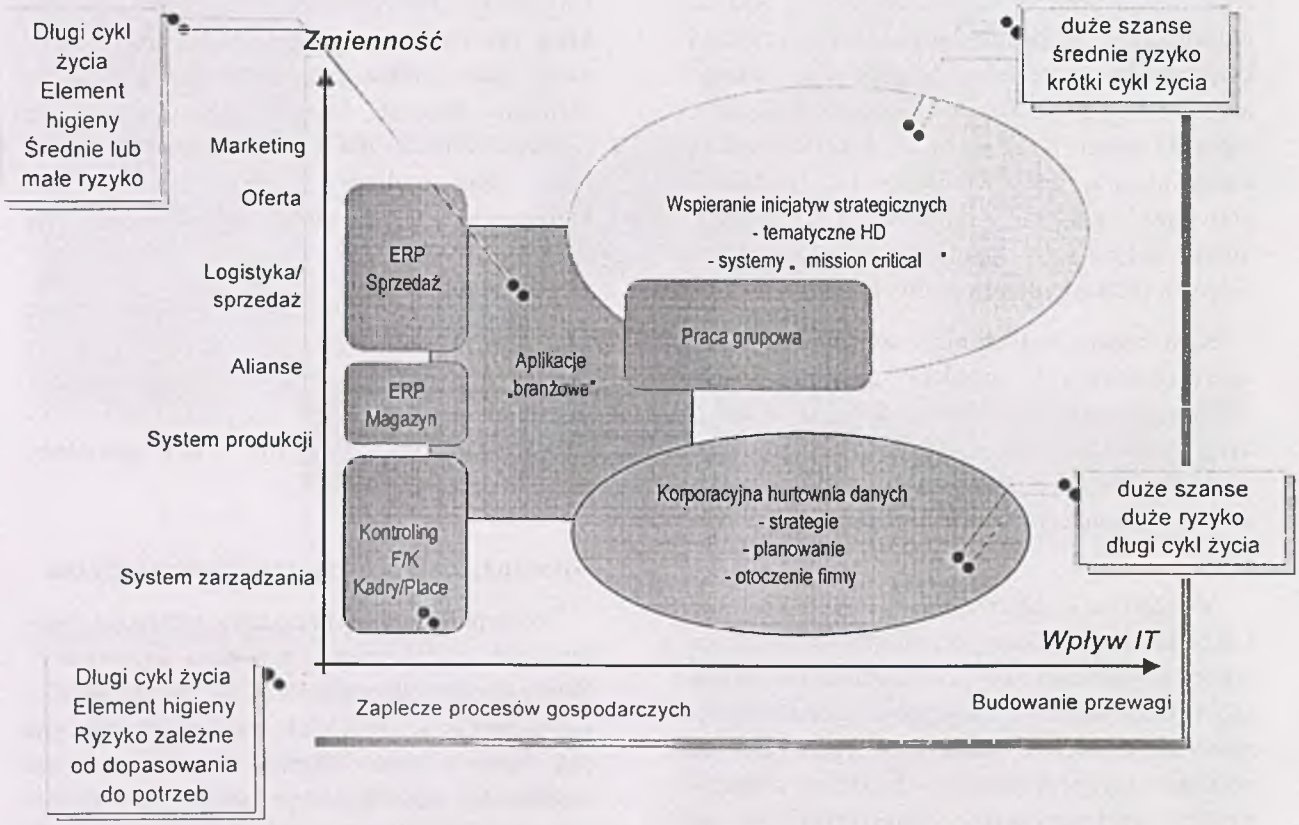
W tym podejściu informatyka może stać się istotnym składnikiem wartości oferowanej klientowi w ramach produktu danego procesu gospodarczego, tak jest na przykład w wypadku usług „*home banking*", czy sprzedaży przez Internet.

Nirwana, czyli informatyzacja strategiczna

Informatyzacja synergiczna poprawia skuteczność codziennego działania organizacji. Wyrasta ona jednocześnie z dojrzałego myślenia o organizacji, jako o bycie funkcjonującym dla dobra swoich klientów oraz satysfakcji właścicieli, posiadającego własną tożsamość, specyficzną kulturę i wartości. Dostrzeganie konkurencyjnego otoczenia firmy, w połączeniu z nieuchronnym dojrzewaniem rynków rodzi potrzebę myślenia o długofalowym rozwoju – pojawia się potrzeba planowania strategicznego. Naturalnym elementem tego procesu staje się planowanie wykorzystania technologii informatycznych dla strategicznych inicjatyw.

Ten model respektuje, a nawet wzmacnia biznesowy punkt widzenia na informatykę w organizacji. Staje się ona normalnym składnikiem kosztów rozwoju, takim jak badania marketingowe, czy koszty restrukturyzacji. Jednocześnie, dojrzałość planowania, opartego na rzetelnej wiedzy o rynku, świadomie artykułowanych, wyrażanych w konkretnych liczbach zamierzeniach gospodarczych, znakomicie ułatwia określenie oczekiwanych korzyści i dopuszczalnych kosztów związanych z informatyką. Weryfikacja wykonalności, oraz przekładanie tych planów gospodarczych

na konkretne przedsięwzięcia związane z budową aplikacji wymaga funkcjonowania w organizacji dobrze zdefiniowanej architektury systemów informatycznych. Pozwala ona kontrolować proces budowy, rozwoju i integracji systemów na wielu poziomach szczegółowości – od ich związków z procesami biznesowymi, po konkretne zasoby informacyjne, komponenty i elementy infrastruktury współdzielone przez poszczególne aplikacje.



Strategiczna architektura systemów informatycznych w firmie

Po pierwsze nie ma armat...

Wspomniane wyżej modele informatyzacji wynikają z obserwacji tego co dzieje się w organizacjach działających na polskim rynku zastosowań informatyki. Wymaganie od polskich firm, aby realizowały najbardziej dojrzały model informatyzacji jest oczywiście nierealistyczne. W istocie niewiele jest dziś polskich firm, posiadających jasno zdefiniowaną strategię, rozumianą przez ogół pracowników i wdrażaną w codziennych działaniach. Dojrzałość kadry menedżerskiej – zarówno informatycznej jak i biznesowej – nie rodzi się z dnia na dzień, również budowanie właściwych relacji pomiędzy dostawcami a odbiorcami usług informatycznych jest procesem raczej długotrwałym. Nie zmienia to faktu, iż wydatkowanie ogromnych kwot pieniędzy na informatyzację zmusza nas dzisiaj do postępowania odpowiedzialnego, zmierzającego w stronę coraz bardziej efektywnego ale i lepiej uporządkowanego biznesu, którego nierozłącznym składnikiem są dziś na świecie systemy informatyczne.

Model	Dysponent budżetu	Inicjatywa	Korzyści	Zalety	Wyzwania
Ad-hoc	Użytkownik (rezerwy budżetowe).	Przypadkowa	Osiągane na poziomie bezpośrednich użytkowników systemów	Bliski kontakt z użytkownikiem i konkretne potrzeby	Synchronizacja przypadkowych inicjatyw. Utrata sponsora.
Mechaniczny	Centralny dział informatyki	Informatyzacja jako element „politycznej poprawności" Potrzeba uporządkowania wydatków na informatyzację.	Ujednolicenie i „uspójnienie" informacji wykorzystywanych w różnych funkcjach biznesowych.	Uproszczone planowanie zakresu. Widoczność na poziomie zarządu – możliwa koordynacja decyzji biznesowych i planów wdrożenia.	Uzyskanie realnej wartości z realizowanych inwestycji informatycznych. Określenie priorytetów informatyzacji. Korzyści związane z efektywnością podziału funkcjonalnego. Petryfikacja zasad działania organizacji.
Synergiczny	Użytkownik – ośrodek decyzyjny związany z zarządem firmy	Wynikająca z potrzeb związanych z poprawą procesów gospodarczych.	Eliminacja składników procesów gospodarczych nie przynoszących wartości.	Możliwość optymalizacji zakresu pod kątem przyjętych celów biznesowych. Optymalne ROI dla projektów.	
Strategiczny	Zarząd firmy na poziomie wyznaczania oczekiwanego ROI Użytkownik, na poziomie szczegółów realizacji.	Związana ze świadomym planowaniem rozwoju firmy na bazie zdefiniowanej strategii biznesowej.	Umożliwienie osiągania celów wyznaczanych w ramach kluczowych inicjatyw strategicznych	Optymalny zwrot z kapitału (ROCE) zainwestowanego w budowę systemów informatycznych.	Konieczność funkcjonowania planowania stratgicznego jako praktyki zarządzania

Marek J. Maniecki

Inter-Design Tessel Systems
Warszawa

Projekt INSPIRE – wstępne podsumowanie

Z początkiem kwietnia 1997 roku rozpoczęliśmy prace w projekcie **INitiative for Software Process Improvement - Régions Extérieuers (INSPIRE)**. Projekt ten, sponsorowany przez Unię Europejską w ramach programu COPERNICUS, jest realizowany w strukturach Europejskiej Inicjatywy d/s Systemów i Oprogramowania (ESSI). Celem jego jest przekazanie doświadczeń i wiedzy, dostępnych w krajach zachodnich, na temat standaryzacji, certyfikacji i doskonalenia procesu produkcji oprogramowania małym i średnim firmom w Europie Centralnej i Wschodniej.

Z powodu rezygnacji z udziału w projekcie w grudniu 1997 roku głównego partnera – angielskiej firmy MARI Group Ltd, projekt był zawieszony przez Komisję Europejską przez okres 10 miesięcy. W wyniku pozytywnej oceny założeń i wstępnych rezultatów, dokonanej przez trzech niezależnych ekspertów w kwietniu 1998 roku, KE podjęła w końcu 1998 roku decyzję o kontynuowaniu projektu ustalając nowy termin zakończenia prac na 30 września 1999 roku.

Projekt jest realizowany obecnie przez konsorcjum złożone z pięciu firm:

- HIGHWARE GmbH - Niemcy,
- PZ Inter-Design Tessel Systems - Polska,
- SIVECO Romania SRL - Rumunia,
- Information Technology Ltd - Węgry,
- Estonian Software Society - Estonia.

Poniżej przedstawimy główne akcje i konkretne działania projektu INSPIRE, już zrealizowane bądź będące jeszcze w trakcie realizacji.

1. Zbadanie stanu świadomości na temat jakości wytwarzania oprogramowania u polskich producentów

Pierwszym zadaniem projektu było przeprowadzenie ankiety dotyczącej stosowania metod zapewnienia jakości w wytwarzaniu oprogramowania w polskich firmach. Ankieta została rozesłana do ponad 400 firm, głównie członków Polskiej Izby Informatyki i Telekomunikacji oraz Stowarzyszenia Polski Rynek Oprogramowania PRO.

Otrzymaliśmy 45 wypełnionych ankiet – wyniki były przedstawione w czasie poprzedniej Szkoły w Szczyrku. Ogólnie można stwierdzić, że większość odpowiedzi była bardzo podobna do rezultatów otrzymanych w czasie analogicznego badania w 17 krajach Europy Zachodniej z 1996 roku. Wyniki badania posłużyły do stworzenia Regionalnego Planu Szkoleniowego.

2. Sponsorowane akcje szkoleniowe

Głównym zadaniem projektu INSPIRE jest przeprowadzenie serii spotkań szkoleniowych, dla Polski wyznaczono limit minimum 10 sesji. Spotkania te mają różny charakter, od uświadamiających do spotkań roboczych dla specjalistów informatyków.

- Seminaria uświadamiające – zwrócenie uwagi środowisk odpowiedzialnych za tworzenie nowoczesnych systemów informatycznych na potrzebę dosko-

nalenia procesu wytwarzania oprogramowania i osiągane przy tym korzyści. Dla jak najefektywniejszego sposobu wydatkowania przyznanych kwot podjęliśmy decyzję o współorganizowaniu takich seminariów z innymi organizacjami, jak na przykład Polskie Towarzystwo Informatyczne (szkoły w Szczyrku i w Mrągowie).

- Sesje szkoleniowe i warsztaty mające na celu wymianę doświadczeń – celem jest przekazanie wiedzy oraz umożliwienie wymiany doświadczeń z już wdrożonych projektów. Odbyło się 5 takich spotkań; wykłady na nich prezentowali m.in. Janusz Górski, Chris Debou, Tomasz Byzia, Michał Halas, Grzegorz Prochowski. Spotkania były oczywiście bezpłatne, średnia liczba uczestników wynosiła ok. 20 osób.
- Spotkania grupy roboczej – to spotkania grupy krajowych autorytetów, kierunkujących działania podejmowane w ramach projektu INSPIRE. Grupą tą stała się Inicjatywa Poprawy Procesów Programowych IP3, która spotykała się kilka razy przy okazji innych spotkań szkoleniowych.

3. Finansowane przez projekt konsultacje mentorów

W polskiej części budżetu INSPIRE została wydzielona kwota na sfinansowanie współpracy mentorów z czterema polskimi firmami wytwarzającymi oprogramowanie. Zadaniem mentorów współpracujących z tymi firmami jest głównie:

- pomoc w ocenie stanu firmy pod względem jakości i zarządzania procesem produkcji oprogramowania,
- pomoc we wprowadzaniu zmian w firmie i w tym procesie produkcji.

Otrzymaliśmy 9 zgłoszeń od konsultantów i 6 od firm. W wyniku ustalonego w projekcie procesu selekcji zostały wybrane następujące pary Mentor-Firma:

- Janusz Górski - PROFI-DATA, Szczecin, Juliusz Niemotko,
- Chris Debou - ComArch S.A., Kraków, Tomasz Niedojadlo,
- Grażyna Nowakowicz - Biuro Projektowania Systemów Cyfrowych, Katowice, Franciszek Szweda,
- Tomasz Byzia - Simple S.A.,Warszawa, Marian Kossowski.

4. Finansowane przez projekt eksperymenty poprawiające proces wytwarzania oprogramowania

Idea eksperymentu polega na finansowym wsparciu firmy, która ma zidentyfikowane problemy związane z procesem wytwarzania oprogramowania i wie jak takie problemy rozwiązać. Korzyści ma odnieść zarówno ta firma, jak i inne z tej branży, poznając założenia, sposób przeprowadzenia i wyniki eksperymentu („case study").

W budżecie projektu mieliśmy środki finansowe na przeprowadzenie 3 eksperymentów. Otrzymaliśmy 5 zgłoszeń z firm chcących zorganizować w siebie eksperymenty. W wyniku procesu selekcji zostały wybrane następujące firmy:

- Max Elektronik S.A., Zielona Góra, Bogdan Ficner,
- Globema sp. z o.o., Warszawa, Tomasz Gulczyński,
- Hobbit sp. z o.o., Warszawa, Bogusław Pluta.

5. Działalność „propagandowa"

W założenia projektu jest szerokie rozpowszechnianie informacji i wiedzy na temat zagadnień związanych z projektem. Realizowaliśmy to poprzez:

- biuletyny projektu – wydanych zostało 5 numerów (zgodnie z planem),
- informacje prasowe o akcjach projektu,
- informacje przekazywane poprzez internet.

Oprócz przedstawionych powyżej konkretnych prac, działalność projektu **INSPIRE** przyniosła szereg dodatkowych korzyści. Za najważniejsze uważam zwiększenie

zainteresowania problemami jakości oprogramowania w prasie informatycznej oraz integrację polskiego środowiska informatyków specjalizujących się w szeroko rozumianej inżynierii oprogramowania. W lipcu 1997 roku została zawiązana Inicjatywa na rzecz Poprawy Procesów Programowych IP3, która stałą się grupą roboczą projektu. Ta nieformalna organizacja działa pod przewodnictwem profesora Janusza Górskiego i obecnie grupuje ok. 20 osób, przedstawicieli zarówno środowiska akademickiego, jak i firm wytwarzających oprogramowanie. W 1998 roku powstała Sekcja Inżynierii Oprogramowania PTI – tym razem już grupa sformalizowana. W wyniku połączenia omawianych działań w październiku 1999 roku odbędzie się 1. Krajowa Konferencja Inżynierii Oprogramowania w Kazimierzu Dolnym.

Jak już to podkreślałem, podstawowym celem projektu INSPIRE jest wymiana wiedzy i doświadczeń na temat zapewnienia wysokiej jakości wytwarzanego oprogramowania. Służyć temu ma również jak najszersze rozpowszechnianie wyników prac i eksperymentów prowadzonych w firmach. Stąd też specjalna sesja prezentacyjno-dyskusyjna w Szczyrku. W dalszej części opracowania mogą Państwo znaleźć krótkie opisy eksperymentów i prac doradczych realizowanych w ramach projektu INSPIRE.

Bogdan Ficner

Max Elektronik S.A.
Zielona Góra

Realizacja eksperymentu INSPIRE w firmie MAX ELEKTRONIK S.A.

Realizacja eksperymentu mającego na celu poprawę procesu tworzenia oprogramowania w ramach projektu INSPIRE w firmie Max Elektronik S.A. obejmuje prace działu rozwoju systemów bibliotecznych.

Krótko o firmie

Max Elektronik S.A. jest firmą działającą na rynku zastosowań informatyki od 1991 roku. Od 1993 roku firma posiada dedykowany dział rozwoju systemów bibliotecznych. Dział ten rozwija linię produktów składającą się ze zintegrowanego systemu bibliotecznego PROLIB, systemu zarządzania dokumentami PROMAX oraz interfejsu www PROWEB dla systemu PROMAX. Systemy te są instalowane w wielu bibliotekach i ośrodkach naukowych na terenie kraju.

Podstawowe cele eksperymentu

Podstawowe cele eksperymentu obejmują prace związane z rozwojem systemu PROMAX. W ramach eksperymentu przewidziano realizację:

a) wdrożenie systemu SCM w pracy zespołu

b) przejścia oprogramowania z platformy Progress 4GL v.7.x na wersję 8.3

c) odtworzenie modelu logicznego danych.

Zadania te związane są z wprowadzaniem nowych wersji środowiska pracy, zwiększającą się ilością członków zespołu oraz wdrożeń systemu, nowymi implementacjami systemu i związanymi z tym procesem problemami w zachowaniu spójności modelu logicznego i fizycznego danych.

Etapy realizowane/zakończone

W dniu dzisiejszym można już mówić o końcowej fazie etapów a) i b). Nie trzeba nikogo przekonywać o potrzebie wdrożenia SCM w projekcie związanym z zapewnieniem jakości. Problem tkwi jedynie w sposobie realizacji tego zadania. Częstym problemem wdrażania SCM jest zbyt duże skupienie się na efektach poprawiających jakość pracy kierownika projektu pozostawiając efekty dedykowane dla członków zespołu na drugim planie. W swoich pracach, ze względu na środowisko pracy (Progress) i koszty (ograniczenia) wybrano narzędzie firmy Starbase RoundTable Lite. Podstawowe zadanie w procesie wdrażania tego narzędzia postawione przed zespołem to:

a) odtworzenie w repozytorium stanu aktualnego systemu,

b) opracowanie metodyki wykorzystania funkcji narzędzia przez kierownika projektu i członków zespołu,

c) wygenerowanie wersji początkowej produktu,

d) realizacja drugiego etapu projektu tj. przejście z platformy Prgs 7.x na wersję 8.3

e) wygenerowanie kolejnej wersji produktu.

Można powiedzieć, że zadania a), b) i c) zostały zakończone, a firma PSC (głównie Technical Support) znacznie pomogła

wskazując przetarte szlaki procesu wdrożenia narzędzia.

Więcej problemów związanych jest z przeniesieniem aplikacji z platformy Prgs 7.x na 8.3. Nie jest to tylko problem rekompilacji kodu źródłowego. Pytanie jak zrealizować zadanie wykorzystując jednocześnie nowe mechanizmy i funkcje nowej platformy (np. SmartObjects). Zadanie to komplikuje dodatkowo fakt, że już po rozpoczęciu prac firma PSC wprowadziła kolejną wersję środowiska z nowym ADM (Modelem Tworzenia Aplikacji).

Etapy pozostałe

Pozostałe etapy pracy to zakończenie przenoszenia systemu do Prgs v.8.3, wygenerowanie kolejnej wersji systemu oraz odtworzenie modelu logicznego danych.

Tomasz Gulczyński

Globema, Warszawa

Doskonalenie procesu analizy systemowej w przedsięwzięciach GIS-owych

Przystępując do programu INSPIRE chcemy poprawić proces rozwoju i wdrażania oprogramowania. Dotąd zajmowaliśmy się bardziej produkcją oprogramowania – teraz dostarczaniem rozwiązań. Niemniej przy okazji opracowywania rozwiązań „pod klucz" chcemy wypracowywać produkty rynkowe.

Za główne cele biznesowe postawiliśmy sobie:

- Dotrzymywanie czasu i budżetu przedsięwzięcia
- Obniżenie kosztów (bądź ekstra zysk) poprzez powtarzalność
- Satysfakcję klienta z wdrożonego rozwiązania
- Większą produktywność: skupienie się na sprawach merytorycznych a nie formalnych

Realizację programu zaczęliśmy od uświadomienia sobie obecnej sytuacji, czyli zdefiniowania procesu produkcji oprogramowania w takiej postaci, jakiej stosujemy go w naszej firmie (poszczególne podprocesy, czynności). Następnie poddaliśmy każdy z tych podprocesów samoocenie, w wyniku której wyodrębniliśmy te dziedziny, które wymagają poprawy w pierwszym rzędzie. Stwierdziliśmy, że taką dziedziną jest analiza wymagań i sporządzanie specyfikacji systemu.

Doskonaleniu podlegają następujące aspekty w obrębie wybranej dziedziny:

1. Procedura
   - Szybka zamiana wymagań na specyfikację systemu (co system ma robić)
   - Płynne przejście do projektowania (jak system ma to robić)
   - Zaangażowanie klienta, ścisły podział funkcji, uproszczona komunikacja
2. Szablony
   - Ścisła strukturalizacja specyfikacji
   - Podstawa odbioru systemu
   - Koncentracja na sprawach merytorycznych
3. Narzędzia; kryteria wyboru:
   - łatwe przejście do projektowania
   - kompatybilność z naszym systemem (Smallworld GIS)
   - możliwości dokumentacyjne i wizualizacyjne
   - polskie znaki itp.
4. Zarządzanie
   - Ryzyko (specyfikacja nie spełnia wymagań bądź jest nieścisła)

Eksperyment kończymy we wrześniu i do tego czasu spodziewamy się zakończyć ten sam proces poprawy dla dwóch kolejnych priorytetowych dziedzin.

Bogusław Pluta

Hobbit, Warszawa

Poprawa efektywności pracy zespołu programistycznego

Eksperyment przeprowadzany jest w ramach realizacji projektu, składającego się z siedmiu etapów prac. Efektem pierwszego etapu jest wstępna analiza wymagań, umożliwiająca sprecyzowanie granic modułów systemu. Kolejnych sześć etapów obejmuje wykonanie poszczególnych modułów systemu.

Eksperyment polega na przeprowadzeniu pewnych zmian w organizacji prac zespołu programistycznego, i pomiarze efektów tych zmian. W tym celu porównane zostaną ze sobą dwa etapy - trzeci i czwarty, obejmujące wykonanie odpowiednie drugiej i trzeciej „paczki" modułów. Wybór tych etapów podyktowany jest chęcią zmniejszenia wpływu na wyniki eksperymentu czynników innych niż świadomie wprowadzane zmiany, np. zakłóceń spowodowanych zapoznawaniem się z klientem czy zamykaniem projektu.

Lista wszystkich miar, jakie zostaną zastosowane do porównania ze sobą dwóch etapów prac zostanie zaprezentowana w trakcie wystąpienia w Szczyrku. Pośród nich znajdą się na pewno:

- stosunek planowanego nakładu pracy na wytworzenie oprogramowania do faktycznego – porównanie wartości w projektach prowadzonych „starymi" metodami, i po wprowadzeniu zmian;
- awaryjność oprogramowania, mierzona stosunkiem ilości negatywnych wyników poszczególnych kroków algorytmu testowania do ogólnej liczby kroków – porównanie wartości w modułach wykonanych „starymi" metodami, i po wprowadzeniu zmian;
- ilość błędów i awarii systemu w ciągu ustalonego okresu eksploatacji (tu: pierwsze dwa miesiące), w stosunku do złożoności modułów mierzonej łączną liczbą kroków w algorytmach testowych i obciążeniem mierzonym średnią liczbą równocześnie pracujących użytkowników.

Planowane zmiany w sposobie realizacji prac obejmują:

- sformalizowanie podziału zadań i procesu kontroli jakości;
- skrócenie zadań (tak, by termin realizacji nie przekraczał 2-3dni);
- wprowadzenie do zasad przydzielania zadań jeszcze jednego kryterium: minimalizacji kosztów (przede wszystkim czasu pracy) integracji wyników pracy poszczególnych osób;
- opracowanie i wdrożenie metody pielęgnacji wymagań (dbanie o posiadanie stale aktualnej analizy wymagań).

Tomasz Niedojadło

ComArch, Kraków

Założenia procesu poprawy jakości oprogramowania w ComArch s.a.

Wszelkie ustalenia na temat konsultacji z mentorem poprzedzone zostały przez głęboką analizę potrzeb spółki w zakresie wytwarzania oprogramowania. W wyniku przeprowadzonej analizy zidentyfikowane zostały obszary, które wymagały bądź to gruntownego przeprojektowania bądź dodatkowych porad ze strony doświadczonego mentora. Wszelkie problemy zostały zidentyfikowane na podstawie wywiadów przeprowadzonych z pracownikami firmy bezpośrednio zaangażowanymi w prace nad produkcją oprogramowania. Dzięki temu uzyskano obraz rzeczywistych potrzeb a nie pobożnych życzeń. Na spotkaniach z mentorem Chrisem Debou ustalony został wstępny harmonogram działań zmierzających do poprawy procesu wytwarzania oprogramowania.

Podstawowe założenia planu współpracy obejmują dwa rodzaje pomocy:

1. Wykłady dla personelu zarządzającego projektami
2. Poprawę procesu w konkretnym wybranym projekcie

Plan wykładów obejmuje tak ważne tematy jak *project managenet planing, project managenet tracking* oraz *kontrolę jakości* i ma na celu wzrost wydajności pracy pracowników oraz poprawę jakości zarządzania projektami. Natomiast plan współpracy z mentorem w ramach konkretnego projektu obejmie analizę przebiegu procesów w projekcie na podstawie szczegółowych wywiadów, analizy dokumentacji i stosowanych procedur. Na tej podstawie stworzony zostanie program poprawy, który będzie obejmował kolejne etapy prowadzenia projektu począwszy od oferty a skończywszy na procesie odbioru systemu.

Dzięki pomocy mentora zamierzamy udoskonalić zachodzące w przedsiębiorstwie procesy, podnieść efektywność naszych działań oraz zapewnić naszym klientom jeszcze lepszą obsługę.

Janusz Górski[1], *Juliusz Niemotko*[2]

[1]*Politechnika Gdańska*
[2]*Profi-Data, Szczecin*

Inspekcje w procesie wytwarzania oprogramowania

W ramach projektu EU Copernicus INSPIRE podjęto eksperyment w zakresie doskonalenia technologii wytwarzania oprogramowania. Eksperyment ten jest wykonywany w firmie Profi-Data w Szczecinie. Związany jest on z uruchomieniem procesu inspekcji w ramach procesu wytworzenia modułu programowego. Eksperyment otrzymał nazwę IPD (Inspekcje w Profi-Data). Plan projektu IPD obejmuje następujące etapy:

E1 Zdefiniowanie celów projektu.

E2 Szkolenie uczestników eksperymentu.

E3 Modyfikacja procesu wytwarzania modułu programowego.

E4 Zdefiniowanie produktów procesu wytwarzania.

E5 Zdefiniowanie procesu inspekcji.

E6 Zdefiniowanie mechanizmów zbierania danych.

E7 Zaplanowanie eksperymentu.

E8 Realizacja eksperymentu.

E9 Ocena wyników eksperymentu.

Obecny stan zaawansowania projektu wygląda następująco:

- Zdefiniowano cele projektu IPD, które obejmują: uruchomienie procesu inspekcji, pomiar wpływu inspekcji na jakość wytwarzanego oprogramowania oraz na nakłady związane w wytwarzaniem oprogramowania.
- Przeprowadzono szkolenie w zakresie metody GQM (ang. *Goal-Question-Metrics*) oraz w zakresie inspekcji oprogramowania.
- Zdefiniowano zakres projektu w ramach projektów realizowanych w ProfiData na rzecz klientów.
- Dokonano modyfikacji procesu wytwarzania pod kątem jego bardziej precyzyjnego zdefiniowania oraz w celu wbudowania w ten proces inspekcji.
- Podjęto decyzje odnośnie formy realizacji poszczególnych inspekcji oraz zapoczątkowano prace zmierzające do zdefiniowania kryteriów poszczególnych inspekcji
- Zdefiniowano zakres i mechanizmy zbierania danych dotyczących eksperymentu.
- Zaplanowano kolejne etapy realizacji eksperymentu.

W świetle planów, eksperyment powinien się zakończyć we wrześniu, a wyniki w formie udokumentowanej powinny być dostępne w październiku 1999 r.

Grażyna Nowakowicz[1], Franciszek Szweda [2]

[1]InfoViDE, Warszawa

[2]BPSC, Katowice

Poprawa procesów wdrażania i pielęgnacji oprogramowania

1. Okoliczności, które zadecydowały o przystąpieniu BPSC do projektu INSPIRE

W ostatnich kilku latach BPSC intensywnie pracowało nad ukończeniem nowego pakietu klasy ERP (środowisko i narzędzia Oracle) dla średniej wielkości przedsiębiorstw. Pakiet w zasadniczej swej części został ukończony w 1997 roku. Wraz z rozpoczęciem sprzedaży tego pakietu staraliśmy się wypracować metodologię wdrożenia. Była to jedna z najistotniejszych spraw towarzyszących powstaniu pakietu. Powodem takiego podejścia do sprawy była zastraszająco duża ilość nieudanych wdrożeń dużych systemów klasy ERP w polskich przedsiębiorstwach. Dlatego też postanowiliśmy zwrócić szczególną uwagę na poprawę procesów związanych z wdrażaniem systemu. Nasze prace w tym względzie zbiegły się z różnymi inicjatywami na rzecz poprawy jakości oprogramowania i prowadzenia projektów informatycznych. Z projektem INSPIRE zetknęliśmy się na początku ubiegłego roku a od marca tego roku korzystamy z doradztwa w ramach wspomnianego projektu.

2. Zakres prac objętych projektem INSPIRE

- Zdefiniowanie i opisanie istniejących w BPSC procedur realizacji wdrożenia pakietu 4GL
- Zdefiniowanie docelowych procedur poprawiających proces wdrożenia
- Realizacja zdefiniowanego procesu

3. Oczekiwane efekty prowadzonych prac

- Poprawienie istniejącej metodologii wdrożenia.
- Optymalizacja procesów okołowdrożeniowych ze szczególnym uwzględnieniem przepływu informacji wewnątrz firmy, z docelowym wdrożeniem systemu przepływu pracy.
- Obniżenie kosztów związanych z opieką nad oprogramowaniem, jego rozwojem w aspekcie zmian generowanych przez wymagania klientów, zmiany przepisów i własnych planów rozwojowych.

Tomasz Byzia

InfoViDE, Warszawa

Doradztwo w firmie Simple S.A.

Powodzenie projektów poprawy procesów wytwórczych oprogramowania zależy od właściwego rozpoznania potrzeb i sformułowania celów doskonalenia. Dlatego przed podjęciem działań projektowych wiele uwagi poświęcono przygotowaniu odpowiedniego kontekstu dla poprawy, włączenia w prace szerokiego grona pracowników oraz szczegółowego przygotowania planów działania. Pierwsze dwa cele osiągnięto za pomocą rozpowszechnienia wśród pracowników firmy ankiety stworzonej w celu szybkiej diagnostyki procesów wytwórczych. Ostatni cel osiągnięto przygotowując szereg dokumentów opisujących przedsięwzięcie, jego cel, zakres, działania, zagrożenia i wymagania. Dokumenty te zostały dostarczone wszystkim kierownikom i odpowiedzialnym w firmie z przeznaczeniem do opublikowania wśród podległych im pracowników.

Diagnoza jakości procesów

Celem przeprowadzonej wśród pracowników firmy ankiety było zebranie obiektywnych (ilościowych) danych o procesach wytwórczych i organizacyjnych działających w firmie. Ankieta została przygotowana na podstawie standardu ISO/IEC 15504 i dotyczyła podstawowych praktyk inżynierskich i organizacyjnych w obszarach i procesach zdefiniowanych przez standard. Wyniki ankiety opracowane statystycznie zostały zaprezentowane kierownictwu firmy na specjalnej sesji, na której ocena procesów opracowana przez doradcę została zweryfikowana i uzupełniona. Profile jakości poszczególnych procesów potwierdziły, że firma dysponuje wysoką kulturą inżynierską i jest gotowa do rozpoczęcia działań na rzecz stabilizacji i dokumentacji procesów wytwórczych. Kierownictwo firmy potwierdziło także dużą trafność wniosków wynikających z ankiety. W przyszłości okresowe przeglądy za pomocą podobnej ankiety pozwolą na wygodne i obiektywne zebranie danych do pomiaru rzeczywistych postępów w projektach poprawy.

Projekt poprawy

Zidentyfikowane w wyniku diagnozowania procesów wytwórczych obszary, w których możliwa jest poprawa stały się podstawą opracowanej strategii doskonalenia firmy. Strategia koncentruje się na podnoszeniu jakości obsługi klienta i wdrożeń oprogramowania produkowanego przez firmę. W złożonym środowisku partnerów i dystrybutorów oprogramowania firmy, krytycznego znaczenia nabiera system informacyjny i on właśnie został wybrany jako pierwszy i podstawowy cel projektu poprawy. W wyniku kolejnych sesji projektowych uszczegółowiono koncepcję rozwiązania, zdefiniowano wymagania informacyjne poszczególnych działów firmy oraz zamodelowano przepływy informacji między nimi. W toku dalszych ustaleń przyjęto zarys architektury technicznej wewnętrznego systemu wspomagającego zarządzanie informacją oraz sformułowano ramowy plan realizacji projektu.

Obecnie trwa uszczegółowianie wymagań, specyfikowanie poszczególnych zasobów informacyjnych oraz pilotaż architektury technicznej.

Tomasz Byzia

InfoViDE sp. z o.o.,
02-119 Warszawa,
Pruszkowska 17,
tbyzia@infovide.pl

Przychodzi informatyk do lekarza ...

Wstęp

W krajowej i zagranicznej prasie informatycznej często można przeczytać o zagrożeniach współczesnej inżynierii oprogramowania, nieudanych projektach, milionowych stratach, itp. Często spotykamy tam podobne stwierdzenia:

- systemy jakości naszych projektów są chore,
- kanały informacyjne, nerw każdego przedsięwzięcia, są wąskimi gardłami zarządzania,
- działania koncentrują się na leczeniu skutków błędów, zamiast na ich profilaktyce,
- diagnostyka niezawodności systemów pozostawia wiele do życzenia,
- dobro klienta leży nam na sercu.

Zauważmy, że słownictwo stosowane w powyższych stwierdzeniach ma wiele wspólnego z medycyną. Skoro język opisu zjawisk zachodzących w informatyce i medycynie jest tak podobny, to czy nie oznacza to, że medycyna jest nauką pokrewną służącą do rozwiązywania podobnych problemów jak te, przed którymi stoją twórcy i użytkownicy systemów informatycznych? Medycyna w odróżnieniu od informatyki jest jednak nauką bardziej dojrzałą, która ma wypracowane stabilne kanony postępowania w sytuacjach typowych (standardy leczenia) oraz metody postępowania z nowymi problemami.

Niniejsza prezentacja jest propozycją systematycznego podejścia do zagadnień poprawy jakości systemów wytwórczych oprogramowania. Dokonuje przeglądu współczesnych technik i metod oceny i poprawy procesów wytwórczych, pokazując ich znaczenie i miejsce w kontekście strategicznych planów poprawy pozycji konkurencyjnej firm na rynku.

Standard leczenia procesów wytwórczych

Uświadomienie problemów

Podstawowym warunkiem podjęcia skutecznej poprawy procesów wytwórczych jest uświadomienie sobie jej potrzeby. Bez tego aktu ludzkiej woli niemożliwy jest żaden postęp. Częsta jest sytuacja, w której od lat działamy w podobny sposób, borykamy się z tymi samymi problemami i nie wyobrażamy sobie, że można inaczej. Dopiero jakieś silne lub dramatyczne zdarzenie (np. niepowodzenia na rynku, odejście wartościowego pracownika) zmusza nas do refleksji nad celem, zakresem i metodami naszej pracy.

Sformułowanie problemów

Kiedy już uświadomimy sobie, że jest „źle" pozostaje odpowiedź na pytanie: co jest źle? Identyfikacja, sformułowanie i opisanie problemów jest podstawą dla wytyczenia planów poprawy. Rzadko zdarza się, że jesteśmy w tym wystarczająco obiektywni. Najczęściej potrzebna jest niezależna trzecia strona, która jest obiektywna, a jednocześnie ma zasób wiedzy wystarczający nie tylko do skutecznego opisania, ale także znalezienia przyczyn danego stanu rzeczy.

Konsultacje

Konsultacje są najbardziej ogólną formą poszukiwania przyczyn problemów i możliwości ich rozwiązania. Zauważmy, że konsultant nie jest stroną, która ma przeprowadzać poprawę procesów. Może w tym pomóc, ale to sami

zainteresowani muszą dołożyć wszelkich starać, aby proces ten się udał.

Konsultacje są formą subiektywnego (jakościowego) badania procesów wytwórczych, które odbywa się najczęściej w formie wywiadów z pracownikami firmy, kierownictwem oraz ewentualnie klientami.

Diagnostyka procesu

Właściwe rozpoznanie problemów jest krytyczne dla skutecznego przeprowadzenia projektów poprawy. Dlatego zalecane jest zobiektywizowanie wyników wstępnych ustaleń za pomocą systematycznych metod badania procesów:

- ankieta diagnostyczna jakości procesów,
- przeglądy projektów,
- przeglądy produktów projektowych.

Audyt

Wstępna diagnoza stanu procesów wytwórczych pozwala szybko i obiektywnie wyszukać te obszary, które mają istotne znaczenie dla jakości produktów projektowych. Wybrane obszary powinny być następnie poddane szczegółowemu badaniu obejmującemu zarówno stosowane tam technologie, metody, rozwiązania organizacyjne, relacje z klientami zewnętrznymi i wewnętrznymi, a także socjologię możliwych zmian. Celem audytu poza dogłębnym zrozumieniem procesów i występujących tam problemów, jest także wskazanie możliwych kierunków poprawy.

Projekt poprawy

Projekt poprawy jest implementacją przyjętej przez firmę strategii doskonalenia w wybranych obszarach. Projekt poprawy składa się zwykle z małych etapów, które rozwiązują kolejne problemy zgodnie z przyjętymi założeniami i priorytetami wynikającymi z oceny ich wagi. Na projekt poprawy wpływ ma wiele czynników, które decydują o jego powodzeniu oraz niepowodzeniu. Spełnienie warunków powodzenia jest czasem bardzo trudne, dlatego ważne jest dobre planowanie projektu, które uwzględnia nie tylko osiąganie celów doraźnych i strategicznych, ale także widzi poprawę jako złożony proces kulturowy wymagający zaangażowania wszystkich pracowników, a głównie kierownictwa firmy.

Przeglądy okresowe

Przeglądy okresowe są przejawem systematycznej troski firmy o jakość jej produktów i procesów. Mają na celu nie tylko wyszukanie pojawiających się nowych problemów, ale także monitorowanie skuteczności realizowanych projektów poprawy, wprowadzonych zmian oraz stabilności wdrożenia mechanizmów zapewnienia jakości.

Przeglądy są także ważnym elementem kultury jakości, która aby skutecznie oddziaływać na procesy w firmie musi opierać się na profilaktyce, tzn. przewidywaniu i usuwaniu zagrożeń zanim staną się problemami.

Janusz Krzysztof Grabara

Politechnika Częstochowska

Komputerowe wspomaganie zarządzania jakością, ochroną środowiska oraz bezpieczeństwem i higieną pracy

Współczesne systemy informatyczne zarządzania stosują coraz nowsze technologie mające na celu przyśpieszenie procesu podejmowania decyzji i rozwiązywania problemów na różnych szczeblach począwszy od szczebla strategicznego, a na szczeblu operacyjnym kończąc. W swoich założeniach mają one poprawę działań organizacyjnych, ekonomicznych i jakości wytwarzanych przez organizację dóbr i usług. Natomiast zapomina się niejako przy projektowaniu systemów o bezpieczeństwie ludzi zatrudnionych w organizacji i stworzeniu szczególnego podsystemu zarządzającego bezpieczeństwem pracy.

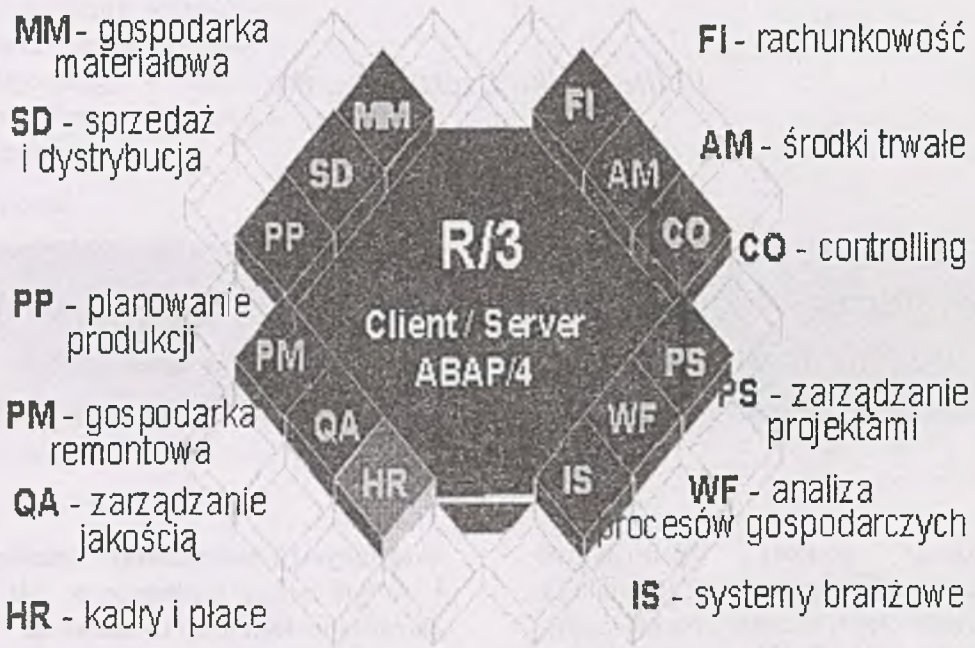
Obecnie działanie w sferze produkcji na rzecz bezpieczeństwa pracy jest rozpatrywane w płaszczyźnie nadzoru i wykonawstwa. W płaszczyźnie wykonawstwa sprowadza się do ścisłego przestrzegania instrukcji stanowiskowych, czynnościowych i bhp przez bezpośrednich wykonawców zadań pracy (produkcji), czyli bez robotników. Udział bezpośrednich wykonawców w tworzeniu bezpiecznych warunków jest zależny od ich kwalifikacji oraz od cech charakterologicznych i stopnia motywacji, czyli osobistego zaangażowania się w pracy. Szeregowy robotnik w kolektywie roboczym rozporządza określonymi narzędziami, mechanizmami, i w tym sensie występuje w roli podmiotu zarządzania techniką i technologią.

W działalności w sferze produkcyjnej istnieją różne formy organizacji, od jej poziomu zależy poziom bezpieczeństwa pracy w zespole, brygadzie, oddziale, wydziale i w przedsiębiorstwie jako całości. Możliwie skuteczne zapewnienie bezpieczeństwa pracy wymaga stworzenia systemu bhp w skali makro i mikro.

Podstawą systemowego podejścia do zagadnień bezpieczeństwa pracy osób zatrudnionych w gospodarce byłoby podejmowanie decyzji dotyczących bezpieczeństwa zatrudnionych z uwzględnieniem skutków tych decyzji w skali całego systemu, a nie jak to się często zdarza w skali przedsiębiorstwa. Systemowe podejście do zagadnień bhp pozwala na ograniczanie poczynań żywiołowych i na kompleksowe działanie programowe w tym zakresie.

Poprawę sytuacji w tym zakresie można osiągnąć idąc jedną z dwóch dróg.

Rozwijać każdą z płaszczyzn oddzielnie i przedstawiać kolejne moduły jako odrębne składniki całości systemu (patrz rysunek 1).



Rys.1 Model integracyjny R/3 firmy SAP Polska

Można też spróbować stworzyć moduł wspólny obejmujący swym zasięgiem obszar jakości, ochrony środowiska i bezpieczeństwa i higieny pracy.

Systemowe ujęcie przedsiębiorstwa w trzech płaszczyznach zarządzania:

- jakością,
- środowiskiem.
- bezpieczeństwem pracy,

jest coraz powszechniej akceptowane. Rangę podkreśla fakt, że dwie z nich zostały uregulowane normami międzynarodowymi ISO 9000 i ISO 14000, a trzecią płaszczyznę regulują normy państwowe z których za wzorcową uważa się normę brytyjską BS 8800, a teraz także istniejące projekty Polskich Norm PPN 18000.

Wszystkie trzy normy razem wzięte określają zespół warunków i postępowania przyczyniających się do podniesienia jakości wyrobów, jakości pracy oraz jakości współdziałania z zewnętrznym otoczeniem (naturalnym i społecznym). Uregulowania te dotyczą tych samych podmiotów, zasobów organizacji i jej zarządzania. W sposób naturalny obszary działalności organizacji gospodarczej regulowane normami w dużym stopniu nakładają się. Taka sytuacja stwarza możliwość łącznego ujmowania spraw jakości, środowiska i bezpieczeństwa pracy w jednym zintegrowanym systemie zarządzania.

Integrację systemów zarządzania jakością środowiskiem i bezpieczeństwem pracy należy rozpatrywać na minimum trzech poziomach:

- polityki,
- podsystemów,
- dokumentacji.

Na poziomie polityki integracja wyraża się sformułowaniem jednej polityki rozwoju przedsiębiorstwa uwzględniającej w równorzędny i jednoczesny sposób aspekty jakości, ochrony środowiska i bezpieczeństwa pracy.

Natomiast integracja z punktu widzenia podsystemów sprowadza się do znalezienia najlepszych rozwiązań w zakresie wszystkich obszarów działalności organizacji gospodarczej uwzględniających wszystkie wymienione wcześniej trzy aspekty. Można to uzyskać jedynie poprzez systemowe ujęcie przedsiębiorstwa określające te obszary, zaangażowane w nich podmioty i zachodzące procesy.

Dokumentacja jest wdzięczną formą integracji bo polega ona na ustaleniu wspólnych elementów norm dotyczących zarządzania jakością, środowiskiem i bezpieczeństwem pracy i wdrożeniu ich do dokumentowania przyjętych rozwiązań.

Elementy zintegrowanego systemu zarządzania jakością BHP i ochroną środowiska

Cele przedsiębiorstwa w zakresie jakości, bezpieczeństwa i ochrony środowiska

Podstawę wszystkich działań powinny. stanowić jasno zdefiniowane, wymierne i odpowiednio udokumentowane cele. Należy jednoznacznie określić zakres odpowiedzialności za realizację przyjętych celów i zapewnić niezbędne do tego środki. Postępy w realizacji celów nadzorowane są w ramach cyklicznych badań prowadzonych w formie controllingu (rachunkowość zarządcza). Stwierdzone ewentualnie odchylenia od wartości założonych stanowią podstawę do zweryfikowania możliwości realizacji zamierzonych celów (z uwzględnieniem zmieniających się okoliczności. krytycznej oceny przydzielonych środków i stosowanych dotychczas metod pracy oraz podjęcia niezbędnych działań korygujących.

Struktura organizacyjna

W przeszłości funkcjonowały najczęściej niezależne systemy zapewnienia jakości i BHP oraz systemy związane z ochroną środowiska. Wzrost zagrożenia naszego środowiska, stopień jego zanieczyszczenia, coraz wyższe wymagania ustawowe i zaostrzające się warunki walki konkurencyjnej na rynku spowodowały konieczność jednoznacznego określenia zakresu kompetencji i obowiązków na najwyższym szczeblu kierowniczym („za sprawy związane ze zintegrowanym systemem zarządzania jakością, BHP i ochrony środowiska odpowiedzialny jest szef - ale nie tylko"). Szczebel ten odpowiada także za właściwą koordynację gwarantującą odpowiednią efektywność działań.

W dużych przedsiębiorstwach zalecane jest utworzenie specjalnej interdyscyplinarnej komórki odpowiedzialnej za realizację tej funkcji w ramach obowiązków kierowniczych W jej skład wchodzą kierownicy działów jakości BHP i ochrony środowiska ("służby ds. zintegrowanego systemu zarządzania jakością, BHP i ochrony środowiska"). W zależności od wielkości i struktury organizacyjnej przedsiębiorstwa oraz prowadzonej w nim działalności zadania te mogą być realizowane przez jedną osobę. Nadrzędnym celem systemów zarządzania jest stworzenie struktur organizacyjnych zapewniających zdolność jednostki do realizacji określonych zadań. Właściwe struktury pozwalają na sprawniejszą realizację bieżących działań w zakresie ochrony środowiska i zarządzania jakością.

Wewnętrzne zależności w ramach zintegrowanego systemu zarządzania jakością, BHP i ochroną środowiska

Osiągnięcie i zapewnienie oczekiwanej jakości przy uwzględnieniu zasad bezpieczeństwa i higieny pracy oraz poszanowania środowiska wymaga planowanego działania. Nowością zintegrowanego systemu zarządzania jakością, BHP i ochroną środowiska jest przekazanie w jedne ręce odpowiedzialności za realizację jego poszczególnych komponentów. Zalety zintegrowanego systemu zarządzania jakością, BHP i ochroną środowiska stają się odczuwalne dla pracownika nawet wówczas, jeżeli nie możemy od razu zmienić procedury i wszystkie dotychczas niezbędne działania muszą być realizowane w starym trybie. Pracownik korzysta jednak tylko z jednego schematu postępowania odpowiednio dostosowanego do wymogów praktycznych.

Struktura operacyjna w ramach zintegrowanego systemu zarządzania

Każdy cel przedsiębiorstwa należy "odpowiednio zaplanować" określić i zrealizować Otrzymane efekty wymagają sprawdzenia i weryfikacji, a następnie uwzględnienia w kolejnych planach. Najważniejsze są główne cele przedsiębiorstwa Dlatego już na etapie ich opracowywania należy mieć na uwadze wszystkie aspekty związane z realizacją życzeń klientów, wymaganiami jakościowymi,

opłacalnością, bezpieczeństwem pracy i ochroną środowiska.

Nadając poszczególnym aspektom odpowiednią rangę tworzymy podstawy realizacji wszystkich procesów w przedsiębiorstwie Poniżej przedstawione zostaną poszczególne etapy cyklu zarządzania w ramach zintegrowanego systemu zarządzania jakością, BHP i ochroną środowiska

Ustalenie celów

Służby do spraw zintegrowanego systemu zarządzania jakością, BHP i ochroną środowiska wspólnie z kierownictwem przedsiębiorstwa, poszczególnymi wydziałami, odbiorcami i dostawcami określają cele i standardy, które należy zrealizować. Ustalany zostaje także zakres kompetencji, sposób realizacji celów, jednostka zobowiązań do sprawdzenia i spełnienia przyjętych zadań oraz sposób prowadzenia zadań. Przedstawione w ten sposób cele stają się wiążące i wymierne. Działają one motywująco i umożliwiają sprawdzenie uzyskanych efektów, dotrzymania terminów i wywiązania się z obowiązków.

Planowanie

Planowanie działań należy do kompetencji poszczególnych wydziałów takich jak: projektowania i konstrukcji, zaopatrzenia, produkcji, jakości, sprzedaży. W zakresie działań realizowanych w ramach zintegrowanego systemu zarządzania jakością, BHP i ochroną środowiska należy zaplanować terminy i sposób prowadzenia nadzoru własnego oraz zorganizowanie i przeprowadzenie badań.

Planowania wymaga ponadto sposób eksploatacji urządzeń technicznych, gospodarka odpadami i ściekami, prowadzenie audytów i realizacja przedsięwzięć szkoleniowych dla pracowników

Decyzja

Plany weryfikujemy pod kątem założonych celów i ich aktualności Na tej podstawie ustala się priorytety i przystępuje do działania. W ramach zintegrowanego systemu zarządzania jakością, BHP i ochroną środowiska należy sprawdzić czy spełniono wszystkie przyjęte założenia.

W przypadku zauważenia niezgodności należy podjąć działania pozwalające na poprawę sytuacji. W zależności od spełnienia wymagań jakościowych wyrób zaliczamy do braków lub stwierdza się konieczność poddania go obróbce dodatkowej. Może zaistnieć potrzeba weryfikacji wybranego materiału pod kątem ekologicznym lub wprowadzenia najnowszych technologii produkcji.

Realizacja

Podjęte decyzje należy realizować zgodnie z ustalonym planem. Służby odpowiedzialne za zintegrowany system zarządzania jakością BHP i ochroną środowiska opracowują odpowiednie instrukcje operacyjne dla pracowników, prowadzą niezbędne rozmowy i konsultacje oraz nadzorują przebieg prac sterując w ten sposób procesem projektowania, konstrukcji, zaopatrzenia. produkcji i nadzoru

Cykliczny charakter zintegrowanego systemu zarządzania jakością, BHP i ochroną środowiska

Dotychczasowe doświadczenia wykazały, że jednostkowe działania w zakresie poprawy jakości oraz zarządzania systemami bezpieczeństwa i ochrony środowiska nie są w stanie przynieść większych i długotrwałych efektów.

Spostrzeżenie to stanowi podstawę systematycznej realizacji procesów w ramach zintegrowanego systemu zarząd jakością, BHP i ochroną środowiska. Proces wdrażania systemu zintegrowanego wykorzystuje jego cykliczny charakter funkcjonowania.

Klasyczna pętla jakości podkreśla znaczenie poszczególnych wydziałów przedsiębiorstwa z punktu widzenia końcowej jakości wyrobu. Uwzględniane są wszystkie działania projakościowe w całym cyklu życia produktu.

Należy zdefiniować wymagania poszczególnych wydziałów w zakresie zintegrowanego systemu zarządzania i urzeczywistniać w ramach realizacji konkretnych działań.

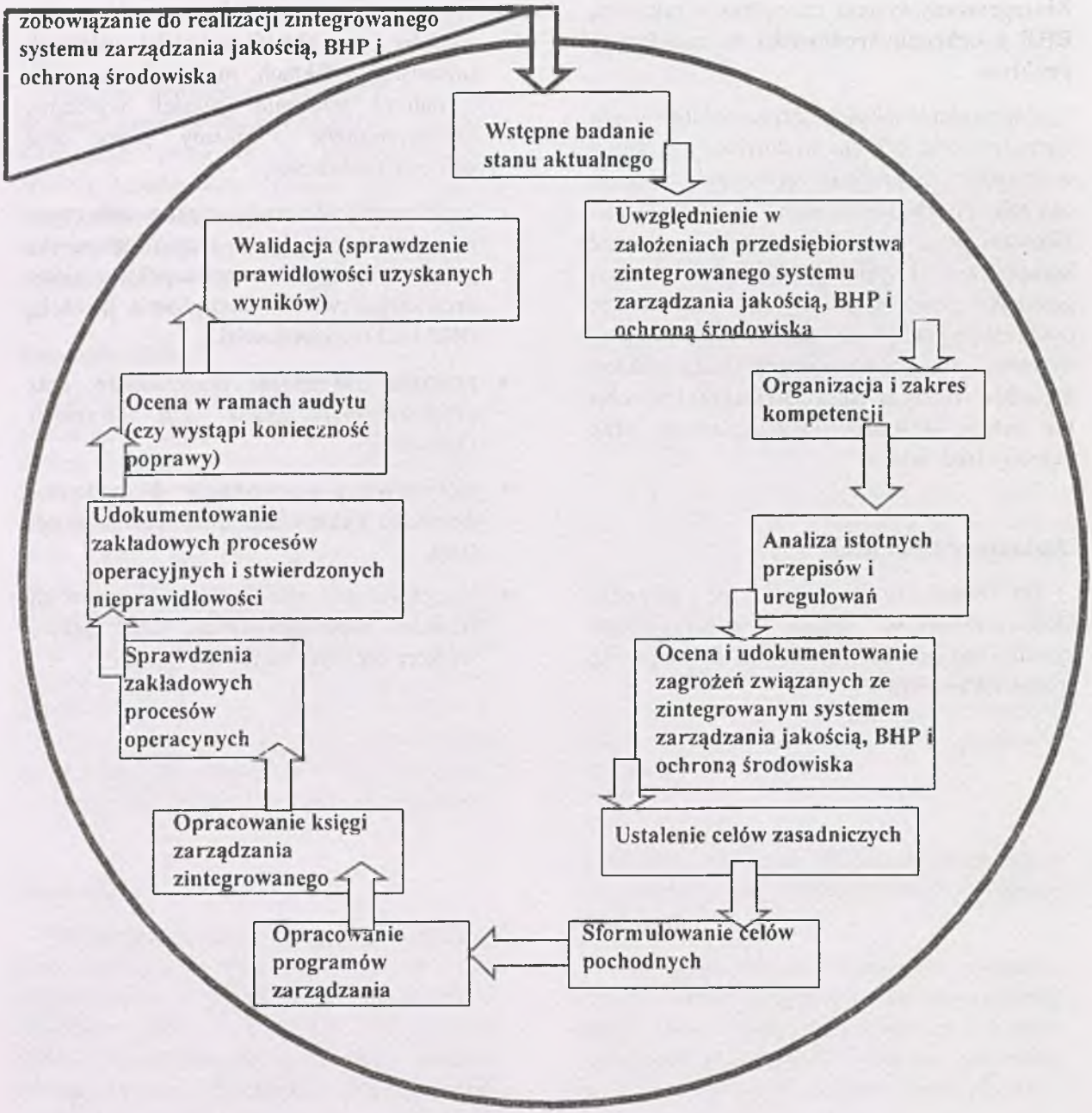
Zintegrowany system zarządzanie jakością, BHP i ochroną środowiska w codziennej praktyce

Zapewnienie jakości, bezpieczeństwo pracy i urządzeń oraz ochrona środowiska odgrywają w świadomym procesie zarządzania zasadniczą rolę. Niezbędni są odpowiednio wykwalifikowani pracownicy, którzy potrafią działać kompleksowo i spełniać daleko idące wymagania. Szkolenia i regularne kursy doskonalące podkreślają praktyczne znaczenie pracy wykonywanej na poszczególnych stanowiskach z punktu widzenia ostatecznej jakości wyrobu lub usługi, warunków bezpieczeństwa oraz ochrony środowiska.

Zadania przełożonych

Do zadań realizowanych na szczeblu kierowniczym w ramach zintegrowanego systemu zarządzania jakością, BHP i ochroną środowiska należą:

- staranne i bieżące informowanie pracowników o obowiązujących ustawach przepisach i faktach, mających znaczenie z punktu widzenia jakości wyrobów, bezpieczeństwa i higieny pracy oraz ochrony środowiska,
- zapewnienie właściwego opanowania czynności wymaganych na danym stanowisku pracy, z uwzględnieniem aspektów zintegrowanego systemu zarządzania jakością, BHP i ochroną środowiska,
- poważne traktowanie pracowników oraz przekazywanych przez nich informacji i propozycji,
- motywowanie pracowników do odpowiedzialnego traktowania spraw przedsiębiorstwa,
- konsekwentne przestrzeganie wszystkich środków zapobiegawczych, dając pracownikom przykład do naśladowania.



Rysunek 2. Cykliczny charakter zintegrowanego systemu zarządzania jakością, BHP i ochroną środowiska.

Elementy zintegrowanego systemu zarządzania jakością, BHP i ochroną środowiska w poszczególnych działach przedsiębiorstwa

Działania związane z zarządzaniem jakością, ochroną środowiska oraz bezpieczeństwem i higieną pracy muszą być bezpośrednią lub pośrednio realizowane we wszystkich wydziałach przedsiębiorstwach. Realizacja tych zadań wymaga stworzenia odpowiednich struktur organizacyjnych. Opierając się na cykliczności procesów realizowanych w ramach zintegrowanego systemu zarządzania jakością, BHP i ochrona środowiska oraz na jego elementach można wymienić kilka zadań realizowanych na poszczególnych stanowiskach (tabele poniżej).

W podobny sposób można wszystkie pozostałe działy zarówno związane z produkcją, gospodarką materiałową, kadrami jak i służbami finasowo-rachunkowymi.

Dokonując szeregowania elementów zarządzania jakością w oparciu o normy ISO serii 9000 oraz zarządzania ochroną środowiska według norm ISO serii 14000, a także

zarządzania bezpieczeństwem i higieną pracy sformułowanych w normach BS 8800 i projekcie polskiej normy PPN serii 18000 można stwierdzić, że jak dotychczas przy obowiązującym stanie prawnym elementy wspólne stanowią około 50% wszystkich wymogów (rysunek 3).

Aby stworzyć zintegrowany system wspomagający zarządzanie jakością, BHP i ochroną środowiska należy odpowiednią pod kątem integracji zweryfikować dokumentację i założenia systemu. Z dotychczasowych doświadczeń wiadomo, że najskuteczniejszym sposobem będzie oparcie się o system zarządzania jakością, gdyż takie systemy informatyczne już są wdrażane. Natomiast jak do tej pory żadna firma wdrażająca zintegrowane systemy nie wdrożyła systemu zarządzania bezpieczeństwem i higieną pracy. Stan taki wynika z faktu, że nikt dotychczas nie zamówił takiego systemu.

Dział	Zadania i problemy dotyczące jakości	zadania i problemy dotyczące BHP	Zadania i problemy dotyczące ochrony środowiska
Kierownictwo	Sformułowanie i określenie celów polityki w ramach zintegrowanego systemu zarządzania jakością, BHP i ochroną środowiska, wdrożenie księgi zarządzania systemem zintegrowanym		
Badanie rynku	Jaki jest popyt rynku na produkty spełniające wymagania jakościowe ?	Jaki jest popyt rynku na wyroby spełniające wysokie wymagania w zakresie bezpieczeństwa ?	Jak silne jest zapotrzebowanie na spełnienie wymagań w zakresie ochrony środowiska ?
Planowanie wyrobu, projektowanie	Jakie szczególne wymagania jakościowe powinien spełniać wyrób lub usługa ?	Jakie są oczekiwania klientów odnośnie formy spełniania wymagań w zakresie bezpieczeństwa wyrobu ?	Jakie są wymagania klienta odnośnie przyjaznych dla środowiska technologii produkcji wyrobu i jego utylizacji ?
Reklama	W jaki sposób firma może wykorzystywać informacje jakościowe do prowadzenia działalności reklamowej?	Które z aspektów bezpieczeństwa mają szczególnie silne oddziaływanie reklamowe?	W jaki sposób można wykorzystać argumenty dotyczące ochrony środowiska przy reklamowaniu określonego wyrobu lub usługi ?
Public relations	Gdzie i w jakiej formie publikowane są informacje na temat koncepcji i działań jakościowych w firmie?	W jaki sposób prezentowane są działania przedsiębiorstwa realizowane na rzecz bezpieczeństwa?	Czy sporządzone są sprawozdania ekologiczne ? Czy publikowane są bilanse ekologiczne i informacje na temat strategii w zakresie ochrony środowiska?
Magazyn surowców i produktów gotowych	Właściwe składowanie nie dopuszczające do utraty jakości.	Zapewnienie bezpiecznych warunków magazynowania.	Zapobieganie zanieczyszczaniu środowiska przez wycieki, emulsje itp.
Dystrybucja/ transport	Zapewnienie jakości wyrobu poprzez właściwe zabezpieczenie, pakowanie i wybór środka transportu.	Przestrzeganie wszystkich przepisów w zakresie BHP	Dobór środków transportu przyjaznych dla środowiska (np. kolej), wyposażenie pojazdów w katalizatory.

Tabela 1 : Kierownictwo przedsiębiorstwa, marketing, dział sprzedaży

Dział	Zadania i problemy dotyczące jakości	Zadania i problemy w zakresie BHP	Zadania i problemy dotyczące ochrony środowiska
Sprzedaż	Doradztwo i wyjaśnienie zapotrzebowań klienta	Doradztwo klienta w zakresie bezpiecznego użytkowania wyrobu, zwrócenie uwagi na zagrożenia związane z nieprawidłowym	Zwrócenie uwagi na zagrożenia dla środowiska związane z wyrobem, informacje na temat ekologicznych preferencji danego wyrobu, właściwa utylizacja itp.

Obsługa klienta, serwis	Doradztwo dla klienta w trakcie użytkowania wyrobu, usuwanie ewentualnych wad, przekazywanie informacji na temat słabych ogniw stwierdzonych podczas użytkowania, usunięcie przyczyn wad, zbieranie doświadczeń praktycznych, informowanie kompetentnych jednostek, wdrażanie działań prowadzących do poprawy.	Usuwanie zagrożenia spowodowanego spowodowanymi wyrobami, wprowadzanie działań mających na celu poprawę bezpieczeństwa, udostępnienie służb awaryjnych, współpraca ze stowarzyszeniami zawodowymi.	Usuwanie zagrożeń dla środowiska spowodowanych wadliwymi wyrobami, wprowadzanie działań zapobiegawczych, oferowanie usług w zakresie utylizacji i zagwarantowanie wykorzystania zużytych materiałów
Badanie podstawowych założeń i projektowanie	Opracowanie i zaprojektowanie nowych, ulepszonych koncepcji jakościowych pozwalających na zapewnienie pożądanych cech jakościowych oraz poprawę realizowanych procedur	Opracowanie i zaprojektowanie nowych koncepcji i technologii związanych z bezpieczeństwem	Opracowanie i zaprojektowanie wyrobów i procedur szczególnie przyjaznych dla środowiska, np. stosowanie regeneracyjnych źródeł energii i surowców odnawialnych oraz opracowanie technologii utylizacji

Tabela 2 : Działalność badawczo rozwojowa i projektowanie

Janusz Górski

Katedra Zastosowań Informatyki
Politechnika Gdańska

Zarządzanie ryzykiem w projekcie informatycznym

1. Pojęcie ryzyka

Intuicyjnie ryzyko oznacza możliwość obniżenia poziomu sukcesu przedsięwzięcia (do kompletnego braku sukcesu włącznie). Tak więc operowanie pojęciem ryzyka wiąże się z odwołaniem się do skali wartości określającej sukces (a właściwie brak sukcesu) przedsięwzięcia oraz do określonej sytuacji z tym związanej (a wiec niechcianej). W skali projektu informatycznego ta *niechciana sytuacja* (nazywana też *elementem ryzyka*) jest definiowana w kontekście udziałowców projektu (klienci, wykonawcy, wdrożeniowcy, konserwatorzy, użytkownicy, itd), np.:

- dla klienta i wykonawcy może to być przekroczenie budżetu lub przekroczenie czasu realizacji,
- dla użytkownika - zła funkcjonalność, „trudny" interfejs użytkowy, nieefektywność czasowa lub wysoka zawodność systemu,
- dla wdrożeniowca - niska jakość oprogramowania, trudność w dopasowaniu go do środowiska docelowego.

Zwykle dąży się do opisania elementu ryzyka wartością punktową lub przedziałem wartości na jednej skali liczbowej (np. w terminach kosztów) lub, jeżeli jest to niemożliwe, na jakościowej skali wpływu na przedsięwzięcie (np. katastrofalny, duży, średni, mały). Oznacza to, że element ryzyka zawsze (jawnie lub nie) będziemy interpretować w kategoriach jego skutków.

Możliwość wystąpienia elementu ryzyka oznacza, że potencjalnie istnieje przejście od sytuacji wyjściowej (obecnej) do sytuacji reprezentowanej przez rozpatrywany element. Dalsza ocena i analiza tej możliwości nie będzie skuteczna jeżeli nie sprecyzujemy informacji na temat sytuacji wyjściowej. Opis tej sytuacji może w szczególności odwoływać się do czasu. Sytuację odniesienia, w stosunku do której opisujemy element ryzyka, a w szczególności oceniamy szansę jego materializacji, nazwiemy *czynnikiem ryzyka.*

Przedsięwzięcie jest pojęciem, które może być stosowane rekursywnie, tzn. dane przedsięwzięcie możemy traktować jako złożone z innych, bardziej szczegółowych przedsięwzięć. W tym sensie możemy mówić o ryzyku w sposób hierarchiczny, rozpatrując je na różnych poziomach szczegółowości i w odniesieniu do różnych zakresów celów. Podobnie można analizować przedsięwzięcia uszeregowane w czasie, gdzie element ryzyka w przedsięwzięciu poprzedzającym może być rozpatrywany jako sytuacja wyjściowa (czynnik ryzyka) w przedsięwzięciu następnym. Dla przykładu nadmierna złożoność programu (element ryzyka przed przystąpieniem do projektowania programu) staje się czynnikiem ryzyka dla nadmiernych kosztów pielęgnacji (od momentu przystąpienia do wdrażania tego programu).

W pojęciu ryzyka zawieramy naszą niepewność odnośnie przyszłości. Należy rozgraniczyć dwa obszary tej niepewności, które najłatwiej zilustrować następującym przykładem.

Załóżmy, że bierzemy udział w grze w ruletkę. Przed przystąpieniem do kolejnej

gry możemy (przy pewnych idealizujących założeniach) skwantyfikować niepewność związaną z sytuacją „wypadnie pole czerwone". Wystarczy w tym celu zastosować prawa rachunku prawdopodobieństwa. Jesteśmy więc w stanie określić dokładnie niepewność dotyczącą rezultatu gry, a szacowanie można uznać za pewne.

Weźmy teraz pod uwagę sytuację gry w pokera. Nawet jeżeli założymy pełną losowość przy rozdawaniu kart i zastosujemy odpowiednie reguły rachunku prawdopodobieństwa, nasze obliczenia dotyczące rozkładów kart nie dają nam w pełni wiarygodnego oszacowania niepewności związanej z końcowym rezultatem. Jest tak ponieważ rezultat ten zależy od wielu czynników, których nie potrafimy uwzględnić w obliczeniach (odporność psychiczna graczy, realizowane przez nich strategie, możliwość bluffowania, itp.). Tak więc, nawet jeżeli oszacujemy niepewność związaną z końcowym wynikiem, samo szacowanie również będzie obciążone niepewnością.

Warto tu również zauważyć, że nawet w stosunku do sytuacji, które poddają się wiarygodnej ocenie niepewności, zachowania ludzi są bardzo różne i zależne od subiektywnej oceny sytuacji. Dla przykładu przywołajmy pojęcie „dobrej passy" w grze w ruletkę, gdzie po pierwszej wygranej, gracz zwykle znacznie wyżej ocenia możliwość kolejnej wygranej pomimo tego, że ma do czynienia z losowymi zdarzeniami niezależnymi. Wskazuje to na subiektywizm ocen oraz potencjalną możliwość ich znacznej rozbieżności w zależności od stanu psychicznego, oczekiwań, doświadczenia i wiedzy oceniającego.

Kryterium sukcesu jest stosowane do konkretnej sytuacji danego przedsięwzięcia, mającej swoje umiejscowienie w czasie, w sensie punktowym (np. przewidziana kontraktem data oddania systemu, data zakończenia etapu budowy systemu, data zakończenia zadania, punkt kontrolny w ramach budowy systemu itp.) lub przedziałowym (np. pierwsze 2 miesiące eksploatacji systemu po jego zainstalowaniu). Rozpatrując ryzyko możemy się również odwołać do czasu w sposób niejawny, np. gdy mówimy o „ryzyku włamania do systemu" mamy na myśli sytuację, dla której moment wystąpienia nie jest możliwy do określenia zanim ta sytuacja nie nastąpi.

Tak więc, mówiąc o ryzyku bierzemy pod uwagę element ryzyka który jawnie lub niejawnie lokalizujemy w czasie przyszłym. Odległość w czasie (od momentu odniesienia do momentu wystąpienia elementu ryzyka) jest ważnym atrybutem ponieważ wyznacza „okno czasowe" na podjęcie ewentualnych działań naprawczych.

Rozważmy przypadek jednokrotnego rzutu monetą. Jako element ryzyka przyjmijmy nie wyrzucenie orła. Możliwość wystąpienia tej sytuacji w przyszłości, przy założeniu jednokrotnego rzutu, da się oszacować dokładnie i wynosi 1/2. Nie zależy to również od tego czy rzut wykonam za chwilę czy za tydzień. Załóżmy teraz, że interesuje mnie ten sam element ryzyka, ale wiem, że rzuty wykonuję często, np. co 10 sek. Wtedy oczywiście możliwość wystąpienia rozpatrywanego elementu ryzyka zależy od ustalenia przedziału czasu (*okresu ekspozycji na ryzyko*). Na przykład, gdy obserwację będę prowadził przez 40 sek., szansa nie wyrzucenia orła wynosi 15/16.

Zależność ryzyka od czasu nie wiąże się wyłącznie z okresem ekspozycji, tak jak to miało miejsce w opisanym wyżej przykładzie rzutów monetą. W okresie ekspozycji na ryzyko mogą zaistnieć fakty, które zmienią szansę wystąpienia rozpatrywanego elementu ryzyka. Dla przykładu, jeżeli wiem że do ważnego dla mnie rzutu monetą mam wystarczająco dużo czasu aby „zamówić" monetę z dwoma orłami, to od momentu podjęcia takiej decyzji (przy założeniu sukcesu w jej realizacji) nie ponoszę ryzyka nie wyrzucenia orła. A więc wpłynąłem na zmianę (obniżenie) szansy niechcianej przeze mnie sytuacji.

Ryzyko może się również zmienić, jeżeli interesujący nas element ryzyka stanie się „mniej niechciany". W tym wypadku odnosimy się do jego skutków. Dla przykładu rozważmy ryzyko związane z zakażeniem AIDS. Cały świat odetchnie z ulgą w momencie gdy zostanie wykryta skuteczna

szczepionka przeciw tej chorobie, ponieważ w sposób radykalny zmienią się skutki związane z zakażeniem, a co za tym idzie, zmieni się również odczucie ryzyka związane z tą chorobą.

Powyższe przykłady wskazują na możliwość aktywnego wpływania na ryzyko poprzez podejmowanie celowych działań.

2. Reprezentowanie ryzyka

Dyskusja przeprowadzona w powyższym rozdziale wskazuje na strukturę informacji niezbędnej do pełnego opisania ryzyka. W ramach takiego opisu powinny znaleźć się następujące elementy.

Zakres	określenie przedsięwzięcia, w kontekście którego definiujemy element ryzyka (np. eksploatacja systemu, projekt wytworzenia systemu, projekt wytworzenia podsystemu, integracja modułu z innymi modułami itp.).
Element ryzyka	opis niechcianej sytuacji w kontekście zakresu.
Skutki	opis skutków wystąpienia elementu ryzyka uzasadniający dlaczego jest on niechciany. Wskazanie na nieuchronność tych skutków (jeżeli związek ten nie jest pewny, sugeruje to konieczność poddania go oddzielnej analizie). Opis skutków może odwoływać się poza zakres związany z danym elementem ryzyka (np. rozważać jego wpływ na inne przedsięwzięcia).
Sytuacja wyjściowa	opis (w ramach przyjętego zakresu) sytuacji wyjściowej, względem której oceniana jest szansa wystąpienia elementu ryzyka.
Szansa wystąpienia	oszacowanie szansy wystąpienia elementu ryzyka przy założeniu sytuacji wyjściowej i jej kontynuacji (w szczególności może to być wyrażone w terminach prawdopodobieństwa).
Ekspozycja w czasie	określenie okna czasowego związanego z ekspozycją na ryzyko.
Powiązania	określenie efektu jaki będzie miało wystąpienie danego elementu ryzyka na inne elementy ryzyka. W szczególności może to być wzrost prawdopodobieństwa związanego z innymi elementami ryzyka, wzmocnienie ich skutków, ograniczenie wyborów mających na nie wpływ lub skrócenie okna czasowego na dokonywanie takich wyborów.
Niepewność	określenie stopnia pewności odnośnie informacji przedstawionej w poprzednich punktach.

W praktyce często dąży się do scharakteryzowania elementu ryzyka za pomocą jednej wartości na skali liczbowej. Tworzy to warunki do porównywania różnych elementów ryzyka i daje podstawę do wyboru spośród alternatywnych decyzji. Często przyjmowana jest następująca definicja:

$$\text{RYZYKO} = P(ER) * S(ER)$$

gdzie

ER - element ryzyka,

P(ER) - prawdopodobieństwo wystąpienia ER,

(ER) - miara skutku wystąpienia ER, najczęściej wyrażana kosztem.

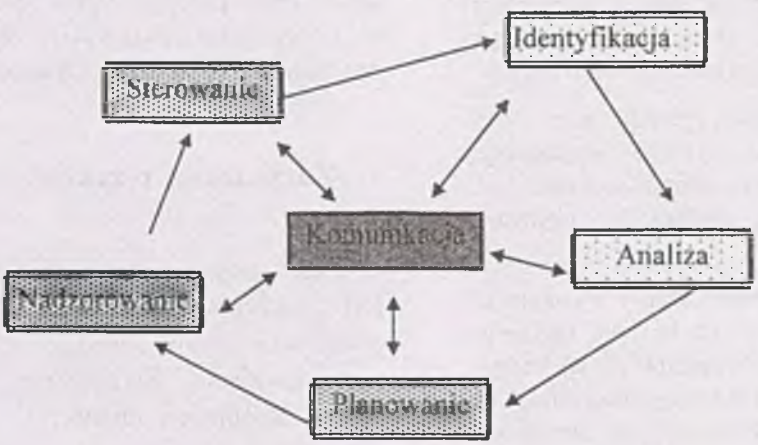
Definicja ta, chociaż przydatna podczas analiz, nie powinna w sposób mechaniczny zastępować pełnego opisu ryzyka podanego w poprzednim paragrafie, gdyż może to prowadzić do nadmiernych uproszczeń.

3. Zarządzanie ryzykiem

Nadrzędnym celem zarządzania ryzykiem jest utrzymanie odpowiedniego stopnia pewności odnośnie sukcesu interesującego nas przedsięwzięcia. Zarządzanie ryzykiem obejmuje następujące elementy:

Ocena ryzyka	*Identyfikacja*:	Identyfikacja niebezpieczeństw zanim zaczną one oddziaływać na przedsięwzięcie. W efekcie powstaje lista specyficznych dla projektu elementów ryzyka, które mogą mieć wpływ na jego sukces.
	Analiza:	Przekształcenie danych o zidentyfikowanym ryzyku w informację decyzyjną. Dla każdego zidentyfikowanego elementu ryzyka oceniane są prawdopodobieństwo i rozmiar strat oraz oceniane są również straty związane z łącznym wystąpieniem elementów ryzyka. Tworzona jest uporządkowana lista elementów ryzyka z przypisanymi im priorytetami.
Obniżanie ryzyka	*Planowanie*:	Wykorzystanie informacji o ryzyku w decyzjach i działaniach określających czynności łagodzące skutki, ustalanie priorytetu planowanych działań i ich integracja w ramach planu zarządzania ryzykiem. W stosunku do każdego elementu ryzyka określany jest sposób postępowania, sposób koordynowania poszczególnych planów i sposób ich integracji w planie ogólnym.
	Nadzorowanie:	Obejmuje śledzenie postępu projektu pod kątem oceny znaczenia poszczególnych elementów ryzyka oraz w miarę konieczności inicjowanie akcji naprawczych.
	Sterowanie:	Korygowanie odchyleń od planowanych działań eliminujących/obniżających ryzyko przy użyciu procedur zarządzania projektem. Dążenie do sytuacji, w której ryzyko jest zredukowane, poprzez obniżanie prawdopodobieństwa wystąpienia elementów ryzyka lub zmniejszanie znaczenia wynikających z nich skutków.
Komunikacja		Wymiana informacji o ryzyku na różnych poziomach organizacji w celu efektywnej realizacji przedstawionych wyżej działań.

Współzależność tych działań ilustruje rysunek.



Rys.1. Zarządzanie ryzykiem

Zarządzanie ryzykiem nie oznacza unikania ryzyka za wszelką cenę. Z ryzykiem mogą wiązać się również większe szanse i możliwości, tak więc (świadome) podejmowanie ryzyka może być w pełni uzasadnione w danej sytuacji biznesowej.

4. Strategie postępowania

W stosunku do zidentyfikowanego elementu ryzyka można rozpatrywać następujące strategie postępowania:

Obniżenie ryzyka	polega na zmniejszeniu prawdopodobieństwa i/lub skutków ryzyka.
Uniknięcie ryzyka	wyeliminowanie możliwości wystąpienia tego elementu ryzyka poprzez wybór drogi alternatywnej. Oznacza to często zamianę danego elementu ryzyka innym, który jest akceptowany lub którym można łatwiej zarządzać.
Transfer ryzyka	spowodowanie, że ktoś inny współuczestniczy lub przejmuje ryzyko na siebie. Przykładem jest skorzystanie z firmy ubezpieczeniowej.
Zaakceptowanie ryzyka	zaplanowanie działań awaryjnych oraz śledzenie ryzyka i wdrożenie planu awaryjnego w sytuacji gdy ryzyko przerodzi się w problem.

Przed dokonaniem wyboru, każdą strategię należy przeanalizować pod kątem jej wpływu na czynniki i atrybuty charakteryzujące rozpatrywany element ryzyka. Wybór strategii wynika z przeprowadzonych analiz uwzględniających stosunek nakładów do uzyskanych efektów. Dla przykładu, przy obniżaniu ryzyka oceniany jest stopień zmniejszenia prawdopodobieństwa i skutków związanych z danym elementem ryzyka w stosunku do kosztu podejmowanych akcji.

Ryzyko staje się problemem gdy jego wartość przekroczy pewną zdefiniowaną wartość graniczną. Dlatego zarządzanie ryzykiem wymaga ustanawiania wartości granicznych, po przekroczeniu których należy podjąć akcje naprawcze. Akcje te muszą być zdefiniowane odpowiednio wcześniej. Wiąże się to z zastosowaniem dwóch typów strategii:

Planowanie akcji. Dotyczy tych ryzyk, które mogą być obniżone przez natychmiastowe działania podejmowane w stosunku do związanych z nimi czynników. Dla przykładu, ryzyko związane z niedostateczną ekspertyzą w zakresie architektury sprzętu może być obniżone przez opracowanie i wdrożenie planu szkoleń zespołu wykonawców, zatrudnienie personelu o potrzebnych kwalifikacjach lub włączenie do zespołu odpowiednich konsultantów. Oczywiście wynikający stąd koszt nie powinien przekraczać spodziewanego uzysku wynikającego z podjęcia tych akcji.

Planowanie postępowań awaryjnych. Dotyczy to tych ryzyk, które wymagają śledzenia i podjęcia akcji naprawczych, jeżeli zaistnieje taka potrzeba. Dla przykładu, w celu zmniejszenia ryzyka opóźnionej dostawy sprzętu przez dostawcę, plan postępowania awaryjnego może przewidywać środki mające na celu obserwację postępów prac zmierzających do dostawy sprzętu po stronie dostawcy oraz niezależnie od tego budowę emulatora docelowego środowiska systemu. Każda z tych akcji będzie uruchomiona gdy przeprowadzona analiza wskaże, że celowe jest poniesienie kosztów na jej realizację. Z drugiej strony, akcja budowy emulatora nie może być podjęta zbyt późno gdyż przydatność emulatora jest uwarunkowana jego dostępnością w określonym czasie. Zwraca to uwagę na konieczność oszacowania okna czasowego związanego z danym elementem ryzyka.

5. Zarządzanie projektem poprzez zarządzanie ryzykiem

Zarządzanie ryzykiem może być włączone w zakres zarządzania projektem. Podejście takie bazuje na *śledzeniu głównych czynników ryzyka* (chodzi tu o ograniczenie się do niewielu, uznanych za najważniejsze czynników ryzyka). Stosowanie tej techniki umożliwia

koncentrację uwagi na kilku zasadniczych czynnikach związanych z dużym ryzykiem, mających znaczny potencjał poprawy oraz krytycznych dla sukcesu przedsięwzięcia. Obejmuje to następujące działania:

- Wyselekcjonowanie głównych elementów ryzyka w projekcie;
- Identyfikacja najważniejszych czynników ryzyka mających wpływ na sukces projektu;
- Ustanowienie planu regularnych przeglądów postępu projektu. Taki oficjalny przegląd powinien być dokonywany pod kierunkiem przełożonego kierownika projektu (aby zmniejszyć subiektywizm ocen). Wewnątrz projektu mogą być również dokonywane przeglądy, z większą częstotliwością;
- Przegląd rozpoczyna się od sprawozdania na temat głównych czynników ryzyka. Należy podać miejsce czynnika na liście, miejsce na liście podczas poprzedniego przeglądu, częstość pojawiania się na liście oraz informacje dotyczące postępów w obniżaniu wpływu tego czynnika ryzyka od ostatniego przeglądu;
- W trakcie przeglądu uwaga koncentruje się na rozwiązywaniu problemów i usuwaniu trudności w działaniach na rzecz usuwania lub obniżania wpływu poszczególnych czynników ryzyka.

Selekcja czynników ryzyka oraz zakres działań ukierunkowanych na obniżanie ich wpływu mogą być wspomagane listami kontrolnymi, które reprezentują doświadczenia danej organizacji zgromadzone z innych, wcześniejszych projektów. Można również wykorzystać w tym celu dane publikowane w literaturze prezentujące wyniki analiz dotyczących projektów w innych organizacjach. Poniższa tabelka przedstawia takie wyniki opublikowane przez B.W.Boehma (*Software Risk Management: Principles and Practices*, W: *Software Management, Fourth Edition, D.J.Reifer Ed.*)

Tabela 1: Dziesięć głównych czynników ryzyka związanych z produkcją oprogramowania.

Czynnik ryzyka	**Działania związane z zarządzaniem ryzykiem**
niedostatki personelu	obsadzanie ról najlepszymi ludźmi; dobieranie ludzi do zadań; budowanie zespołu; motywowanie kluczowego personelu; szkolenia.
nierealny budżet i harmonogram	szczegółowe, oparte na różnych źródłach szacowanie kosztu i budżetu; projektowanie w granicach kosztu; przyrostowa budowa systemu; wielokrotne wykorzystanie; „odchudzanie" wymagań.
wytworzenie niechcianych funkcji lub cech systemu	analiza celów systemu; analiza organizacji, dla której system jest przeznaczony; sformułowanie koncepcji operacyjnej; szersze kontakty i zaangażowanie użytkowników; prototypowanie; wczesne wytworzenie podręcznika użytkownika; analiza efektywności; analiza czynników jakości.
wytworzenie nieadekwatnego interfejsu użytkownika	prototypowanie; specyfikowanie wymagań na bazie przypadków użycia (*use ceses)*; analiza zadań użytkowników; włączenie użytkowników w proces specyfikowania i projektowania.
„pozłacanie" systemu	„odchudzanie" wymagań; prototypowanie; analiza kosztu-do-uzysku; projektowanie w granicach kosztu; kontrola zmian.
ciągły strumień zmian w wymaganiach	podniesienie progu akceptacji zmian; ukrywanie informacji; budowa przyrostowa („zamrażanie" wymagań na poziomie przyrostu).
niedostatki w komponentach dostarczanych z zewnątrz	benchmarking; inspekcje; sprawdzanie powiązań; analiza kompatybilności
niedostatki w zadaniach wypełnianych na zewnątrz	sprawdzenie powiązań; audyt przed podpisaniem kontraktu; kontrakty z możliwością premii; zamawianie konkurencyjnych produktów; budowanie zespołu

niedostatki wydajności	symulacja; benchmarking; modelowanie; prototypowanie; instrumentowanie wspomagające pomiary; strojenie
przecenienie możliwości technologii informatycznej	analiza techniczna; analiza kosztu-do-zysku; prototypowanie; sprawdzanie powiązań i zależności

Każdy z elementów na powyższej liście może być dalej rozwijany do bardziej szczegółowych list kontrolnych.

Implementacja zarządzania ryzykiem wymaga wprowadzenia praktyk i zasad zarządzania ryzykiem do działań objętych zarządzaniem projektem. Zalecany sposób implementacji to droga przyrostowa (ewolucja a nie rewolucja). Pierwszy krok to wprowadzenie śledzenia głównych czynników ryzyka. Kluczowe znaczenie ma tu koncentracja uwagi na głównych czynnikach mających wpływ na sukces (w odróżnieniu od podejścia tradycyjnego, opartego na z góry zdefiniowanych schematach postępowania, które często powodują koncentrację uwagi na działaniach nie mających decydującego znaczenia dla powodzenia projektu). W ten sposób zarządzanie ryzykiem nie jest po prostu jeszcze jednym procesem dodanym do już praktykowanych. Jest to zasadnicza zmiana w podejściu do podejmowania decyzji w warunkach niepewności i wiodące kryterium przy podejmowaniu tych decyzji.

Efektywne zarządzanie ryzykiem wymaga szerokiego propagowania i wdrożenia w ramach organizacji zachowań ukierunkowanych na identyfikowanie i komunikowanie informacji związanej z ryzykiem oraz czynne zaangażowanie się w działania zmierzające do utrzymania ryzyka w akceptowalnych granicach. W efekcie następuje w organizacji stopniowa zmiana kulturowa, bez której zarządzanie ryzykiem przybierze jedynie formę zbiurokratyzowaną, a jego skuteczność będzie niewielka.

Tomasz Łopaciński

IBM Polska,
ul. 1 Sierpnia 8,
02-134 Warszawa

WSDDM – pakiet narzędzi do wspomagania zarządzania przedsięwzięciami w firmie IBM

1. Wprowadzenie

W drugiej połowie naszego wieku, wraz z rozwojem technologii informatycznej na świecie i wzrastającą liczbą prowadzonych coraz bardziej złożonych przedsięwzięć, pojawiła się potrzeba usystematyzowania wiedzy dotyczącej zarządzania tego typu projektami.

Przodujące w świecie korporacje rozpoczęły pracę nad stworzeniem modeli zarządzania opierając się na danych empirycznych pochodzących z aktualnie prowadzonych oraz zakończonych przedsięwzięć. Miało to na celu ograniczenie ryzyka projektowego, ukierunkowanie projektu na zapewnienie oczekiwanych rezultatów projektu oraz uzyskanie przez inwestora pełnej kontroli nad przebiegiem projektu.

Prace trwały całe lata. Mimo zwiększającej się świadomości i coraz bardziej zaawansowanych technologii statystyki wykazują, że w ostatnich latach tylko kilkanaście procent wszystkich projektów informatycznych w Stanach Zjednoczonych skończyło się sukcesem czyli w zamierzonym czasie, w ramach założonego budżetu i wypełnia określony na początku zakres. Wynika to z faktu, że wciąż wiele projektów prowadzonych jest z pominięciem stosowania jakiejkolwiek zweryfikowanej metodyki zarządzania projektami.

The Standish Group w swym raporcie „CHAOS", dotyczącym przyczyn porażek w projektach informatycznych, jako główne czynniki sukcesu przedsięwzięcia podał zaangażowanie użytkownika, wsparcie ze strony kierownictwa, precyzyjnie zdefiniowane wymagania oraz właściwe planowanie.

W swych pracach nad narzędziami wspomagającymi pracę Kierownika Projektu IBM skoncentrował się nad wsparciem procesów planowania i realizacji projektu. Narzędzia realizujące te zadania wchodzą w skład pakietu WSDDM (Worldwide Solution Design and Delivery Methods).

2. Metodyka Zarządzania Projektami firmy IBM

Podczas 75-ciu lat istnienia firma IBM blisko współpracowała z Klientami wypracowując wspólnie zasady zarządzania przedsięwzięciami aby zminimalizować prawdopodobieństwo niepowodzenia prowadzonych wspólnie projektów.

Liczba, różnorodność i zasięg przedsięwzięć prowadzonych przez IBM oraz liczba osób i firm zaangażowanych do ich realizacji wygenerowała potrzebę stworzenia przejrzystego i uniwersalnego modelu procesu zarządzania przedsięwzięciami. Kilka grup ekspertów z IBM pracowało równolegle nad stworzeniem metodyki zarządzania projektami dostosowanej do działalności firmy. Największym uznaniem spotkała się metodyka opracowana w IBM UK – Management the Implementation of the Total Project (MITP). Po uznaniu przez korporację tej metodyki jako wiodącej wprowadzono do niej kilka modyfi-

kacji i teraz pod nazwą Project Management Methodology (PMM) jest powszechnie stosowana jako standard korporacyjny.

Metodyka PMM opiera się na czterech procesach:

- *Zarządzanie Planem Projektu:*

  śledzenie postępu prac, raportowanie statusu realizacji zobowiązań.

- *Zarządzanie Kontraktem*:

  dokumentacja kontraktowa, wypełnianie zobowiązań kontraktowych, fakturowanie, gromadzenie funduszy.

- *Zarządzanie Wyjątkami*

  zmiany, problemy i ryzyka.

- *Zapewnienie Jakości:*

  przeglądy jakości prowadzenia projektów pod kątem zgodności z metodyką. Przeglądy nie koncentrują się na jakości dostarczanych w ramach projektu produktów.

Poniżej przedstawiono trzy fazy składające się na metodykę PMM:

- Rozpoczęcie Projektu
- Realizacja Projektu
- Zakończenie Projektu

Zadania powiązane z fazami Rozpoczęcia i Zakończenia Projektu są realizowane raz. Natomiast zadania z fazy Realizacji Projektu są wykonywane w sposób iteracyjny i przebiegają równolegle do zadań z faz technicznych.

2.1. Rozpoczęcie Projektu

Rozpoczęcie projektu odbywa się zgodnie ze zdefiniowanymi w korporacji standardami i zapewnienia, iż wymagana infrastruktura techniczna oraz zasoby są dostępne. Plan Projektu podlega przeglądowi na właściwym poziomie szczegółowości i zatwierdzeniu.

2.2. Realizacja Projektu

Cykliczna realizacja grup zadań, które zapewniają iż projekt jest pod kontrolą oraz zaproponowane rozwiązanie zostanie dostarczone na czas, w budżecie i będzie zgodne ze wszystkimi założonymi standardami jakości oraz spełni ustalone wraz z klientem Kryteria Akceptacji.

2.3. Zamknięcie Projektu

Formalne zamknięcie projektu i wszystkich dokumentów kontraktowych oraz zwolnienie zasobów. Przeprowadzenie Spotkania Zamykającego Projekt w celu zebrania i uaktualnienia technik estymacji, udokumentowania wszystkich nowych doświadczeń zdobytych podczas realizacji przedsięwzięcia oraz oceny efektywności zastosowanych procedur i metod. W trakcie tej fazy oceniane jest również Zadowolenie Klienta z dostarczonego rozwiązania i podjęcie ewentualnych czynności, by skorygować zidentyfikowane niedociągnięcia.

3. WSDDM - Narzędzia firmy IBM wspomagające zarządzanie przedsięwzięciami informatycznymi

Powodzenie projektu zależy w dużej mierze od zastosowanej właściwej metodyki. Poprawna metodyka umożliwia poprawną definicję projektu, analizę ryzyka, systematyczne planowanie i wdrażanie rozwiązań, zarządzanie zmianami w przedsięwzięciu oraz ocenę postępu prac projektowych.

W ramach stosowanego przez IBM pakietu narzędzi do zarządzania projektami znajdują się programy:

- Planner 2
- Project Control Book 2 (PCB 2)

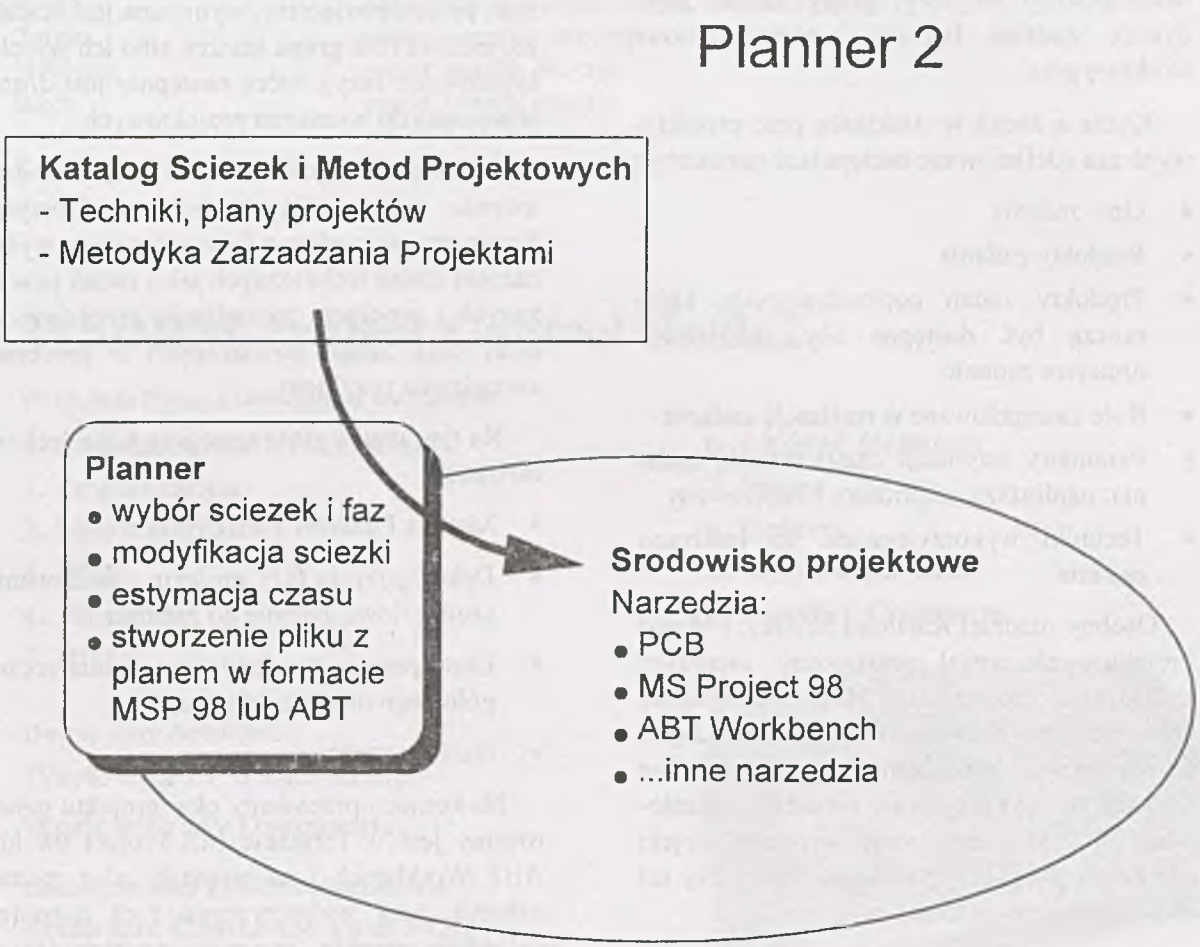
Planner znajduje zastosowanie w procesie przygotowywania planu projektu, natomiast Książka Kontrolna Projektu (PCB) jest wykorzystywana podczas codziennej pracy zespołu projektowego. Wspomaga organizację zespołu projektowego, zarządzanie spotkaniami oraz procesy zarządzania kontraktem, zmianami, ryzykiem i problemami.

3.1. Planner 2

Planner 2 jest aplikacją, która wspomaga Kierownika Projektu w opracowaniu planu projektu. Jej podstawową funkcją jest dostosowanie jednego lub więcej zdefiniowanych, w Katalogu Ścieżek i Metod Projektowych, planów projektów do aktualnych wymagań.

Aplikacja składa się z trzech podstawowych części:

- Katalogu Ścieżek i Metod Projektowych
- Przeglądarki dokonującej wizualizacji danych zawartych w katalogu
- Właściwej aplikacji Planner pozwalającej na modyfikację ścieżek projektowych i dostosowanie ich do potrzeb opracowywanych projektów, ocenę ryzyk, estymację czasu trwania przedsięwzięcia oraz wygenerowanie pliku z planem projektu w formacie MS Project lub ABT Workbench.



3.1.1. Katalog Ścieżek i Metod Projektowych

Katalog zawiera dokładny opis 38 ścieżek projektów informatycznych. Ścieżki te dzielą się na dwie podstawowe grupy: Przygotowanie Oferty (Solution Design) i Wdrożenie Rozwiązania (Solution Delivery).

Wykaz wybranych ścieżek projektowych:

- P01 Client/Server Application Development
- P02 Infrastructure Design
- P03 Rapid Solutions
- P04 Information Engineering
- P05 Host-Based Application Development
- P06 Structured Development 1.0
- P07 Package Selection and Implementation
- P08 Package Implementation - SAP
- P09 Systems Integration and Rollout
- P10 Enhancement
- P11 Corrective Maintenance
- P13 Workflow Manager
- P14 One of a Kind
- P15 Object Technology - FastPath
- P16 Object Technology - Extended
- P22 Transformation 2000 Detailed Analysis Planning
- P23 Transformation 2000 Implementation

- P32 Product and Rollout Services
- P33 Data Center Traditional Design and Build
- P34 Logical and Physical Connectivity
- P35 Data Center Fast Track Design Build
- P37 Europath

Każda ze ścieżek projektowych podlega dekompozycji na fazy, grupy zadań, pojedyncze zadania tworząc czteropoziomową strukturę prac.

Każde z zadań w strukturze prac projektowych ma zdefiniowane następujące parametry:

- Opis zadania
- Produkty zadania
- Produkty zadań poprzedzających, które muszą być dostępne aby zrealizować niniejsze zadanie
- Role zaangażowane w realizację zadania
- Parametry estymacji czasu trwania zadania: najdłuższy, najkrótszy i sugerowany
- Techniki wykorzystywane do realizacji zadania

Osobny rozdział Katalogu Ścieżek i Metod Projektowych został poświęcony metodyce zarządzania projektami PMM. Szczegółowa dekompozycja wszystkich zadań niezbędnych do poprawnego zarządzania przedsięwzięciem pozwala na wykorzystanie metodyki niezależnie od faktu czy opracowywany projekt należy go grupy Przygotowanie Oferty czy też Wdrożenie Rozwiązania.

3.1.2. Przeglądarka

Narzędzie to służy do wizualizacji danych zawartych w katalogu.

3.1.3. Planner

Właściwe narzędzie stosowane do procesu planowania projektu. Posiada ono cztery podstawowe moduły:

1. Definicja projektu
2. Planowanie
3. Estymacja czasu trwania projektu
4. Generowanie pliku w formacie MS Project 98 lub ABT Workbench.

W definicji projektu zawarte są wszystkie podstawowe informacje o projekcie czyli: nazwa projektu, klient, typ projektu, zastosowana metodyka zarządzania projektami oraz poziom zabezpieczeń.

W module planowania identyfikowane i analizowane są ryzyka związane z planowanym przedsięwzięciem, wybierana jest ścieżka projektowa (lub grupa ścieżek albo ich wyselekcjonowane fazy), która następnie jest dostosowywana do wymagań projektowych.

W module estymacji szacowany jest czas trwania poszczególnych faz projektowych. Szacowane są zarówno fazy związane z wykonaniem zadań technicznych jak i zadań powiązanych z procesem zarządzania przedsięwzięciem oraz zadań powiązanych z procesem zarządzania ryzykiem.

Na tym etapie stosowane jest kilka technik estymacji:

- Metoda Punktów Funkcyjnych
- Dekompozycja fazy projektu i szacowanie szczegółowe zadanie po zadaniu
- Ekstrapolacja wyników szacowania szczegółowego na pozostałe fazy
- Narzuty czasowe

Na koniec opracowany plan projektu generowany jest w formacie MS Project 98 lub ABT Workbench i na powstały plan można nałożyć, przy wykorzystaniu tych narzędzi, kalendarz projektu, przypisać poszczególnym rolom dostępne zasoby i opracować harmonogram całego przedsięwzięcia.

3.2. Project Control Book 2 (PCB 2)

Książka Kontrolna Projektu (PCB 2) została opracowana na platformie Lotus Notes z pełnym wykorzystaniem funkcjonalności tego systemu w zakresie zarządzania dokumentami i pracy grupowej.

Głównym celem prowadzenia Książki Kontrolnej Projektu jest dokumentowanie wszelkich istotnych z punktu widzenia projektu decyzji i aktywności. Wewnętrzne powiązania pomiędzy dokumentami ułatwiają wypełnianie formularzy zgodnie ze zdefiniowanymi procesami.

Książka posiada 11 rozdziałów:

1. Standardy i procedury - standardowe i nowe procedury
2. Organizacja - role i przypisane im osoby, zakres obowiązków, odpowiedzialność
3. Punkty węzłowe - opisy i terminy
4. Plany - plan bazowy i kolejne modyfikacje
5. Spotkania - agenda, raport, decyzje
6. Ryzyka - identyfikacja, analiza, decyzja, monitorowanie
7. Zmiany - żądanie, analiza, decyzja
8. Problemy - raport, analiza, decyzja
9. Błędy - raport, analiza, decyzja
10. Kontrakty z klientem - kontrakt, aneksy, ustalenia
11. Kontrakty z podwykonawcami - kontrakt, aneksy, ustalenia

WSDDM Project Control Book 2

Work with Project Documents by Section:

0. Standards and Procedures
1. Organization
2. Milestones
3. Plans
4. Meetings
5. Risks
6. Change Requests
7. Issues
8. Defects
9. Client Contracts
10. Supplier Contracts

Day-to-Day Activities:

Work with PCB Documents

Work with My Documents

Getting Started \ PCB2 Maintenance

Setup and Customize Your PCB

PCB Overview

4. Literatura

1. „CHAOS", The Standish Group, 1995
2. Exception Management Guide, IBM Corporation
3. Financial Management Guide, IBM Corporation
4. Organization and People Management Guide, IBM Corporation
5. Project Completion Guide, IBM Corporation
6. Project Control Book Guide, IBM Corporation
7. Project Definition Guide, IBM Corporation
8. Quality Management Guide, IBM Corporation
9. Risk Management Guide, IBM Corporation
10. Work Breakdown Structure Guide, IBM Corporation

Danuta Kajrunajtys

Mali też mają swoje potrzeby

Streszczenie: Prowadzenie drobnej działalności gospodarczej wymaga (poza coraz mniej liczną grupą) ewidencjonowania przychodów i rozchodów na cele rozliczeń fiskalnych. Różnego rodzaju czynniki motywują właścicieli firm do zakupu zestawu komputerowego i oprogramowania, które zgodnie z intencją autorów powinno wspomagać procesy ewidencyjno-rozliczeniowe. W referacie podjęto próbę analizy sytuacji zarówno wytwórców oprogramowania jak i klientów korzystających z niego do swych potrzeb.

Słowa kluczowe: SI, podatkowa księga przychodów i rozchodów, rejestry VAT,

1. Wstęp

Twórcy i odpowiedzialni za rozwój metod informatyki (metodyk badania potrzeb informacyjnych, analizy SI, projektowania SI itp.) odnoszą swe koncepcje do takiego modelu organizacji, w którym możliwym jest wyodrębnienie funkcji i procesów realizowanych przez co najmniej kilkuosobowe zespoły pracowników.

Autorce nie są znane przypadki zastosowania tychże metodyk wobec drobnej działalności gospodarczej w celu określenia strategii działania, dobrania narzędzia wspomagającego SI itp.

Przedmiotem omówienia w niniejszym raporcie są trzy obiekty: końcowy użytkownik, produkt oraz jego wytwórca. Poza typowymi pojawiają się w połączeniu tych trzech elementów problemy wynikające ze skali i uwarunkowań socjologicznych.

2. Klient

Analiza narzędzi wspomagania i warunków tworzenia oprogramowania zostanie przeprowadzona w odniesieniu do cech charakterystycznych klientów zainteresowanych jego zakupem. Jest to niezbędne do tego, abyśmy stworzyli sobie model wyobrażeniowy sytuacji, w której zakupione oprogramowanie jest eksploatowane.

Obiektem naszych zainteresowań jest drobna działalność gospodarcza (jednoosobowa lub spółka cywilna) zarejestrowana przez bezpośredniego wykonawcę (wytwórcę, usługodawcę). Podmiot taki swoją działalność wykonuje samodzielnie, często wykorzystując zaangażowanie członków najbliższej rodziny, okresowo korzystając z pomocy osób współpracujących na zasadach np. umowy zlecenia lub rozwijając skalę działalności zatrudnia kilku do kilkudziesięciu pracowników. Referat dotyczy tych firm, które nie są zobowiązane do prowadzenia pełnej ewidencji księgowej (z mocy Ustawy o Rachunkowości)

Rozpoczęcie działalności (czynności rejestracyjne) związane jest z dokonaniem kilku wyborów rzutujących w dalszym ciągu na sposób i zakres rozliczania się z należności podatkowych. Większość podmiotów zobowiązana jest do prowadzenia podatkowej księgi przychodów i rozchodów oraz rejestrów VAT. Kandydat na „małego biznesmena" staje przed koniecznością rozpoznania uwarunkowań, dokonania oszacowania podstawowych przewidywanych charakterystyk gospodarczych swojego biznesu i wyboru form opodatkowania. Wszystko to można by zamknąć pojęciem „zaplanowanie organizacji" przy czym skala zjawiska jest zupełnie inna niż to tradycyjnie rozumiemy pod tym pojęciem.

W początkowym okresie działalności w większości przypadków osobiście pełni takie role jak zaopatrzeniowca swej działalności, bezpośredniego wykonawcy, sprzedawcy czy księgowego.

2.1. Działalność podstawowa

Dla ustalenia uwagi, rozpatrzmy przykładowo działalność warsztatu samochodowego założonego przez mechanika samochodowego (czeladnika, mistrza czy absolwenta szkoły samochodowej), warsztatu fryzjerskiego, sklepu, hurtowni, wytwórcy siatki ogrodzeniowej (produkowanej w garażu własnego domku jednorodzinnego), fermy drobiarskiej, firmy świadczącej usługi w dziedzinie zakładania i utrzymania zieleni, zakładu dentystycznego, cukierni itp.

Działalność ta ma charakter jednorodny, wąsko specjalizowany. Prowadzący działalność (właściciel firmy) jest zobowiązany zgodnie z wymaganiami formalnymi rejestrować zaszłości jakie miały miejsce w minionym dniu (lub z kilkudniowym poślizgiem).

2.2. System informacyjny

Zwykle pod pojęciem systemu informacyjnego rozumie się węzły przetwarzania i kanały przesyłania informacji w organizacji. Poszczególne węzły odpowiadają ze realizację funkcji niezbędnych do osiągania przez organizację jej celów. Hipotetyczną i obwarowaną wieloma zastrzeżeniami strukturę *drzewa celów* na nasze potrzeby możemy sobie wykreować. Znajdą się w niej problemy związane z podstawową sferą działalności (technologia, jej logistyka, ewidencja i rozliczanie), wynikające z zatrudniania pracowników, rozwój organizacji i planowanie działań. Jeśli porównamy transakcje powstające w związku z procesami mającymi miejsce w wymienionych przykładowo obszarach działalności (funkcjach) to można postawić tezę, że różnią się one od tych analizowanych przez wieloosobowe zespoły konsultantów w bardziej złożonych organizacjach jedynie liczbą transakcji w okresie czasu oraz skoncentrowaniem wiedzy i umiejętności (i potrzeby poszukiwania tej wiedzy) w rękach jednej osoby (właściciela).

Jak wynika z pobieżnej oceny (bez prowadzenia dogłębnych badań ankietowych), w większości przypadków motywów do zakupu sprzętu i oprogramowania dostarczyły czynniki drugo – trzeciorzędne (w tym naciski dorastających dzieci). Brak rzetelnej analizy (zastawienia choć podstawowych oczekiwań) kończy się zwykle zakupem sprzętu być może doskonale nadającego się do uruchamiania gier multimedialnych i zupełnie przypadkowego oprogramowania do prowadzenia ewidencji działalności gospodarczej (często dodawanego gratisowo w ramach promocji przez producenta komputera)

2.3. Obowiązki podatkowe

Pod pojęciem *obowiązki podatkowe* rozumie się rygory dokonywania wpłat stosownych podatków i przekazywania do urzędów skarbowych dokumentów zawierających dane, z których wynika wysokość naliczonych wpłat. Harmonogram wpłat podatków i przekazania dokumentów unormowany jest stosownymi dokumentami prawnymi. Obowiązek podatkowy pojawia się z chwilą poinformowania właściwego urzędu skarbowego o rozpoczęciu działalności.

Powyższe oznacza iż, wymagane jest od prowadzącego działalność dysponowanie wiedzą o aktualnym stanie wymagań formalno – prawnych, śledzenie i rejestrowanie zmian parametrów podatkowych (stawki, progi, klasyfikacje itp.) bądź też zmuszony jest do skorzystania z usług biura podatkowego.

Właścicielem (właścicielami w przypadku spółki cywilnej) jest zwykle specjalista w branży, w której podejmuje działalność, (bardzo niewiele małych firm jest inicjowanych przez inwestorów posiadających kapitał a zatrudniających specjalistów z branży, w której planowana jest aktywność).

Jak łatwo zorientować się z podanych wcześniej przykładów, istotny odsetek biznesmenów stanowią osoby legitymujące się średnim bądź zasadniczym zawodowym, a nawet podstawowym wykształceniem. Znając realia szkolnictwa na tym poziomie, można stwierdzić, że poziom ogólnej znajomości prawa i przepisów wykonawczych, poziom wiedzy o możliwościach informatyki i umiejętności skorzystania z komputera oraz co najważniejsze świadomość konieczności podnoszenia własnych kwalifikacji takich osób należy oceniać średnio lub nisko.

3. Produkt

Przedmiotem analizy jest oprogramowanie wspomagające prowadzenie księgi przychodów i rozchodów oraz ewidencji VAT, pozwalające na tej podstawie dokonywać stosownych rozliczeń z budżetem.

3.1. Proces technologiczny

Wytwarzanie oprogramowania o powyżej nazwanych funkcjach ma w wielu firmach dość długą historię. Oprogramowanie to powstawało wówczas w wielu przypadkach jako entuzjastyczna reakcja na możliwość (stworzoną przez dostępny sprzęt) eksploatowania oprogramowania użytkowego na stanowisku pracy. Cóż, kiedy zmianie możliwości eksploatacji nie towarzyszyły zmiany w procesach produkcyjnych. Obserwowaliśmy np. „interakcyjne przetwarzanie wsadowe", któremu z czasem próbowano dodać elementy dialogowe.

Genezą dostępnego dziś na półkach sklepowych dealerów bądź w działach sprzedaży wytwórców oprogramowania jest zazwyczaj jedna z dwóch poniższych sytuacji:

1. programiści (nie analitycy, lecz specjaliści od programowania w Pascalu, dBase, itp.) po zgromadzeniu stosownych aktów prawnych dokonali ich analizy i zakodowania tworząc oprogramowanie „użytkowe" ale niekoniecznie użyteczne,
2. przygotowane w przeszłości na potrzeby prowadzenia własnej ewidencji wybranego modułu / funkcji (głównie ewidencji sprzedaży) oprogramowanie zostało „przyjaźnie" obudowane i ewentualnie wzbogacone w kilka funkcji, stając się na pozór spójnym merytorycznie pakietem.

Krótkie sprawdzenie, jak oferowane oprogramowanie wspomaga niektóre sytuacje (i czy w ogóle je uwzględnia) pozwala nabywcy ocenić, czy oprogramowanie to *może* być dla niego użyteczne, czy też *na pewno* nie przyniesie pożytku. Nie ma on jednak żadnych szans ocenić że będzie ono *na pewno* użyteczne. Jego sytuację pogarsza często fakt, że wyboru dokonuje przed podjęciem działalności, mając tylko mgliste wyobrażenie swoich potrzeb. Głównym kryterium wyboru staje się wtedy cena nabycia oprogramowania i *prezencja* ekranów.

3.2. Dokumentacja

Obecnie już jej dostarczanie wraz z pakietem oprogramowania nie stanowi problemu. Problemem są natomiast jej jakość – w szerokim tego słowa znaczeniu. Dokumentacja jest zazwyczaj pisana przez samych twórców oprogramowania, dzięki czemu znajdują się w niej niezamierzone skróty, pominięcia elementów *oczywistych*, ponadto zmuszeni do pisania programiści widzą w tym bardziej element dyskryminacji niż czynnik podnoszący użyteczność ich własnego produktu. Aktualizacja, dostarczanie poprawek i uzupełnień w związku ze zmianami w pakiecie nie nadąża za zmianami w oprogramowaniu. Skala działalności wytwórców oprogramowania w bardzo wielu przypadkach utrudnia (lub całkowicie wyklucza) wydatkowanie nakładów na (po opracowaniu merytorycznym) dopracowanie dokumentacji pod względem wygody w częstym z niej korzystaniu.

3.3. Sprzedaż i marketing

Na półkach sprzedawców oprogramowania stoją obok siebie gry komputerowe i oprogramowanie użytkowe. Nie mogą one być jednak przez sprzedawców traktowane jednakowo. Zakupić grę przyjdzie nastolatek, korzystający z komputera, który nawet jeśli popełni z niewiedzy (braku doświadczenia) jakieś błędy będzie to skutkować co najwyżej opóźnionym rozpoczęciem zabawy z zakupioną grą. Od sprzedającego produkt z grupy „oprogramowanie wspomagające prowadzenie ..." oczekuje się nie tylko biernego odpowiadania na jednoznacznie postawione pytania klienta, ale także zachowań skutecznego sprzedawcy. Cóż kiedy, w wielu sklepach obsługa miała bardzo niewiele kontaktu z informatyką.

4. Producent

4.1. Geneza rozwoju analizowanych firm

Poszukiwanie tematyki do tworzenia oprogramowania (zwrócę uwagę, że to nie to samo, co rozwiązywanie problemów przy pomocy narzędzi informatyki) rozpoczęło się wraz z dostępnością komputerów osobistych. Grupy entuzjastów (raczej programistów, gdyż im było łatwiej rozwiązywać problemy natury technologicznej) rozpoczęły wprowadzanie na rynek swe

produkty w chwili, kiedy rynek ten nie był jeszcze przygotowany do równoprawnej (partnerskiej) współpracy z dostawcą. Nielicznymi klientami były osoby posiadające elementarną wiedzę z podstaw informatyki, potrafiące samodzielnie rozwiązywać ujawniające się w toku eksploatacji problemy. Skutkiem tego, wytworzyła się wygodna sytuacja: dość samodzielny użytkownik nie wymuszał jakości produktu. Upływ czasu z jednej strony radykalnie zwiększył potencjalną liczbę klientów ale też zmieniły się oczekiwania klientów – ukształtował się rynek odbiorcy. Nieświadomego odbiorcy.

4.2. Problemy małych firm software'owych

Wśród istotnych problemów należy wymienić:

- zatrudniani są pracownicy, często studenci lub osoby nie wiążące rozwoju swej kariery z firmą, raczej traktujący ją jako chwilowy etap w swym rozwoju; analitycy z doświadczeniem poszukują pracy atrakcyjnej tak merytorycznie jak i finansowo, dominacja programistów stwarza atmosferę wyższości tej grupy zawodowej, która równocześnie dzięki doskonałej znajomości zagadnień technologicznych czuje się kompetentna w interpretowaniu nieskomplikowanych na pierwszy rzut oka zagadnień podatkowych,
- dobór kadr jest intuicyjny, nie stosuje się sformalizowanych metod oceny kompetencji i kwalifikacji,
- firmą zarządza bardzo często właściciel – kiedyś wykonujący zawód programisty i osiągający na tym polu satysfakcję zawodową, któremu dziś trudno jest się pogodzić z koniecznością pozostawienia (z wielu powodów) swego „konika" i rozwinięcia aktywności jako organizator i weryfikator pracy zatrudnionych pracowników.

Małym firmom zwykle brak jest środków a przede wszystkim motywacji do podjęcia rozmów z zewnętrznymi analitykami i doradcami, podczas gdy same nie znają rzeczywistych potrzeb swojego klienta ani nie nadążają za zmianami wprowadzanymi przez MF i inne organa centralne wydające decyzje co do sposobu postępowania podatników. Problem stanowi dla nich także wprowadzanie zmian i dystrybucja bieżących wersji (upgrade'ów).

5. Sugestie końcowe

„Klientów przyciągają obietnice, a zadowolenie sprawia, ze pozostają klientami długo. Obietnice można stworzyć działaniami marketingowymi, ale satysfakcję może zagwarantować jedynie cała firma" (Philip Kotler)

Usługa oferowana właścicielom prowadzącym działalność gospodarczą powinna mieć charakter kompleksowy: począwszy od doradztwa na etapie podejmowania decyzji o rozpoczęciu działalności (być może nawet uświadomienie korzyści płynących z opracowania nawet pobieżnego biznesplanu), pomocy w doborze i dostosowaniu do charakteru prowadzonej działalności oprogramowania wspomagającego ewidencję i rozliczanie z innymi podmiotami oraz bieżący serwis oprogramowania.

Być może na wzór brokerów ubezpieczeniowych szanujących się firm, każdy wyrażający zainteresowanie „biznesmen małej skali" powinien pozostawać pod stałą opieką doradcy odpowiedzialnego za bieżącą pomoc w prowadzeniu działalności. Wśród głównych obszarów tej pomocy widziałabym:

- doradztwo na etapie planowania działalności gospodarczej (biznesplan, wybór formy prowadzenia ewidencji i opodatkowania);
- systematyczne przekazywanie informacji o zmianach w przepisach dotyczących podmiotu – w zakresie jego działalności: zmiany stawek, klasyfikacji, obligatoryjnych terminów, prawa podatkowego itp.;
- dystrybuowanie, instalowanie, w razie potrzeby szkolenie z zakresu użytkowania oprogramowania (lub zmodyfikowanych jego modułów);
- doradca z natury byłby zainteresowany śledzeniem pojawiających się wykładni, interpretacji i przepisów wykonawczych i wybieraniem z nich elementów oddziaływujących na będące pod jego opieką podmioty gospodarcze;
- w sporach urzędu i podmiotu mógłby służyć pomocą w wyszukiwaniu podstaw i przesłanek prawnych, pod które podpada sprawa sporna.

Działalność tego rodzaju niezależnych doradców związanych lub nie z wytwórcą oprogramowania powinna być odczuwana w skali gospodarki, nie tylko z punktu widzenia wspomaganych podmiotów. Obserwujemy spory brak inicjatywy z obawy przed koniecznością samodzielnego przejścia i zrozumienia przepisów podatkowych, umiejętności rozsądnego i w granicach prawa zaplanowania działalności, popełnianie wręcz elementarnych błędów będących w prosty sposób do uniknięcia. Świadomość możliwości skorzystania w każdej chwili z porady pozwoliłaby przedsiębiorcom bardziej koncentrować się na prowadzeniu biznesu.

„Jakość systemów informatycznych – problemy i rozwiązania"

pod redakcją ***Piotra Fuglewicza***

Wydawca:

Polskie Towarzystwo Informatyczne
Oddział Górnośląski
40-012 Katowice
ul. Św. Jana 10

Druk i oprawa:

ZM „BUMAR-ŁABĘDY" S.A.
Drukarnia Informatyki
44-109 Gliwice 9
ul. Mechaników 9

Projekt okładki:

Bogusław Jackowski
BOP s.c.
Gdańsk

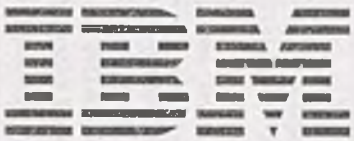
Przygotowanie materiałów do druku:

Mirosława Minich

SPONSORZY

JEDENASTEJ GÓRSKIEJ SZKOŁY PTI

Szczyrk 21-25 czerwca 1999



Jedenasta Górska Szkoła PTI, Szczyrk, 21–25 czerwca 1999

Jakość systemów informatycznych – problemy i rozwiązania

PROGRAM

TERMIN	8:45–9:30	9:45–10:30	11:00–11:45	12:00–13:30	15:00–15:45	16:00–16:45	17:15–18:45	20:15
Poniedziałek 21.06.99	**Robert Adamski** (Andersen Consulting, Warszawa): *Kompleksowe sterowanie jakością w procesach Integracji Biznesu* **Prof. Piotr Jędrzejowicz** (Wyższa Szkoła Morska, Gdynia): *Komputerowe wspomaganie procesów podejmowania decyzji* **Bogdan Machowski** (Motorola, Kraków): *Wprowadzanie systemu jakości według modelu CMM*				**Dr Barbara Begier** (Politechnika Poznańska): *Ogólne zasady inżynierii a poziom dojrzałości procesów tworzenia oprogramowania*	**Prof. Henryk Krawczyk** (Politechnika Gdańska): *Narzędzia i eksperymenty pomiarowe oprogramowania*	**Dr Marek Miłosz** (Politechnika Lubelska): *Jakość systemów informatycznych: między teorią a praktyką*	**Coctail**
Wtorek 22.06.99	**Dr Marian Kuraś, Agnieszka Zając** (Akademia Ekonomiczna, Kraków): *Technologia Informatyczna – konieczny wydatek czy korzystna inwestycja?*	**Michał Hałas** (ODiTK, Gdańsk): *Filozofia „Łańcucha Krytycznego" czyli jak gwałtownie przyśpieszyć projekt*	**Christophe Debou** (Q-labs, Kraków): *Statistical Usage Testing – opis procesu i przykład narzędzia*	**Dr Marcin Sikorski** (Politechnika Gdańska): *Techniki pomiarów i oceny jakości użytkowej produktu*	**Prof. Piotr Jędrzejowicz** (Wyższa Szkoła Morska, Gdynia): *Rozwiązywanie problemów za pomocą algorytmów opartych na ewolucji populacji*	**Konkurs Prac Dyplomowych PTI** Prezentacje laureatów	Dyskusja panelowa *Czy opłaca się stosować komputery?* Moderator: **Dr Marian Kuraś**	**Wieczór autorski Dr Andrzej Jaszkiewicz** (Politechnika Poznańska): *Inżynieria oprogramowania*
Środa 23.06.99	**Dr Jerzy Nawrocki** (Politechnika Poznańska): *Szacowanie rozmiaru oprogramowania*	**Rafał Juraszek** (Computerland, Gliwice): *Wdrażanie systemu zapewnienia jakości w projekcie ALSO-PULS*	**Tomasz Koniecko** (QAD Polska): *Q-Advantage jako narzędzie zapewnienia jakości wdrożenia MFG/PRO*	**Borys Stokalski** (InfoViDE, Warszawa): *Zliczyć do trzech – analiza opłacalności inwestycji informatycznych*	**Dr Marek Maniecki**, *Podsumowanie projektu INSPIRE* **Bogdan Ficner**, Max Elektronik SA, Zielona Góra **Tomasz Gulczynski**, Globema sp. z o.o., Warszawa **Anna Ostaszewska, Bogusław Pluta**, Hobbit sp. z o.o., Warszawa **Tomasz Byzia – Marian Kossowski**, Simple SA, Warszawa **Christophe Debou – Tomasz Niedojadło**, ComArch SA, Kraków **Janusz Górski – Juliusz Niemotko**, Profi-Data sp. z o.o., Szczecin **Grażyna Nowakowicz – Franciszek Szweda**, BPSC sp. z o.o., Katowice			**Koncert** Ela Adamiak i Andrzej Poniedzielski
Czwartek 24.06.99	**Tomasz Byzia** (InfoViDE, Warszawa): *Przychodzi informatyk do lekarza...*		**Janusz Grabara** (Politechnika Częstochowska): *Komputerowe wspomaganie zarządzania jakością, środowiskiem i bezpieczeństwem pracy*	**Prof. Janusz Górski** (Politechnika Gdańska): *Zarządzanie ryzykiem w projekcie informatycznym*	**Tomasz Łopaciński** (IBM): *WSDDM – pakiet do wspomagania zarządzania przedsięwzięciami w firmie IBM*	**Danuta Kajrunajtys** *Mali też mają swoje potrzeby*	Dyskusja podsumowująca Moderator: **Piotr Fuglewicz**	**Ognisko**
Piątek 25.06.99	**Tomasz Byzia** (InfoViDE, Warszawa): *Zarządzanie jakością w projekcie informatycznym* **Grzegorz Prochowski** (Pro-Cons, Warszawa): *Piąta dyscyplina w projektach informatycznych* **Prof. Kazimierz Subieta** (IPI PAN, Warszawa): *Wprowadzenie do obiektowych metodyk projektowania i notacji UML*							

PTI *Zarząd Główny*
ul. Żelazna 87
00-879 Warszawa
tel./fax (0-48-22) 652-32-59
pti_zg@orgmasz.waw.pl

Oddział Górnośląski
ul. Św. Jana 10
40-012 Katowice
tel./fax (0-48-32) 253-96-78
ptiog@netpol.pl

CZŁONKOWIE WSPIERAJĄCY POLSKIE TOWARZYSTWO INFORMATYCZNE

IBM Polska Sp. z o.o.
02-134 Warszawa, Wiśniowy Business Park, ul. 1 Sierpnia 8

„Océ-Poland Ltd." Sp. z o.o.
02-232 Warszawa, ul. Łopuszańska 53

SAP Polska Sp. z o.o.
02-672 Warszawa, Mokotów Bussiness Park, ul. Domaniewska 41

oraz

Centralny Ośrodek Informatyki Górnictwa S.A., 40-065 Katowice, ul Mikołowska 100

ICL Poland Sp. z o.o., 01-199 Warszawa ul. Leszno 21

InfoViDE Sp. z o.o., 02-119 Warszawa, ul. Pruszkowska 17

Ster-Projekt Sp. z o.o., 02-652 Warszawa, ul. Magazynowa 1

Zakłady Przemysłu Barwników „BORUTA" S.A., 95-100 Zgierz, ul. Struga 30