

dr inż. Katarzyna Skroban
Wydział Zarządzania
Politechnika Warszawska

Wdrażanie metod zarządzania jakością przez dostawców oprogramowania i odbiorców

Implementing methods of quality management by suppliers of the software and recipients

Streszczenie: *W referacie podjęto tematykę związaną z wykorzystaniem narzędzi doskonalenia jakości do poprawy funkcjonowania przedsiębiorstw informatycznych oraz narzędzi, które mogą być wykorzystane przez użytkowników - odbiorców oprogramowania. Krótko scharakteryzowano narzędzia, które w ocenie autora, mogą być wykorzystane w branży informatycznej. Dokonano analizy możliwości zastosowań narzędzi w poszczególnych fazach cyklu życia oprogramowania, specyficznych z punktu widzenia dostawcy i odbiorcy. Wymieniono także specyficzne metody stosowane do zapewniania jakości w przedsiębiorstwach o charakterze informatycznym.*

Słowa kluczowe: oprogramowanie, cykl życia oprogramowania, narzędzia zarządzania jakością, model CMM, model RUP, proces dostarczania usług

Abstract: *The area in this paper is associated with using tools of improving the quality for the improvement in functioning of IT enterprises and tools which can be used by users-recipients of the software. Briefly characterized were tools which in the assessment of the author, can be used in the IT industry. An analysis was made on the possibility of the application of tools in individual phases of the life cycle of the software, particularly from the point of view of the supplier and the recipient. Also mentioned were specific methods applied for assuring qualities in computer enterprises.*

Key words: software, life cycle of the software, tools of the quality management, CMM model, RUP model, process of delivering services

Wstęp

W XXI wieku ludzkość staje się społeczeństwem informacyjnym. Wdrażanie systemów informatycznych wspomagających procesy zarządzania w przedsiębiorstwach staje się koniecznością. Stąd z jednej strony istnieje wiele firm oferujących systemy informatyczne oraz znacznie większa grupa odbiorców. Zarówno grupa dostawców jak i odbiorców jest w znacznym stopniu zróżnicowana. Zróżnicowanie odbiorców wynika z różnych aspektów, m.in. z wielkości przedsiębiorstwa oraz profilu i zakresu działania. Na-

tomiast zróżnicowanie dostawców wynika z potrzeb oraz różnic, charakteryzujących odbiorców oprogramowanie.

Oprogramowanie

Oprogramowanie stosowane przez użytkowników i projektantów jest różne. Jedną z definicji przedstawia oprogramowanie jako zbiór zróżnicowanych środków programowych. Zaś program jako specyficzny opis sposobu rozwiązywania pewnego zadania¹.

Oprogramowanie stanowi jeden z podstawowych elementów systemu informatycznego. System informatyczny składa się z 3 podstawowych składników takich jak: oprogramowanie, sprzęt oraz użytkownik.

Według innej definicji oprogramowanie są to programy komputerowe. Program komputerowy jest zbiorem instrukcji dla sprzętu. Programy przetwarzają uniwersalną maszynę zawartą w sprzęcie w maszynę wyspecjalizowaną do wykonania pewnego zadania. Można wyróżnić rodzaje oprogramowania takie jak: systemy operacyjne, języki programowania, aplikacje. Można także podzielić oprogramowanie na tworzone masowo na rynek i oprogramowanie tworzone specjalnie dla określonej organizacji.²

Według normy PN-ISO/IEC 25000:2008 oprogramowanie zostało zdefiniowane w odniesieniu do klienta-odbiorcy. Oprogramowanie klienta stanowi specyficzny produkt programowy, który został opracowany tak aby uwzględniać wymagania użytkownika³. Takie podejście wskazuje na zróżnicowany podział podejścia do oprogramowania inny z punktu widzenia dostawcy oraz inny z punktu widzenia odbiorcy.

Oprogramowanie według normy PN-ISO/IEC 27005: 2010 jest traktowane jako jedno z aktywów wspierających, elementów istotnych dla przedsiębiorstwa z punktu widzenia bezpieczeństwa informacji. W normie został dokonany podział oprogramowania na 4 grupy:

- system operacyjny, czyli wszystkie programy, pozwalające na uruchamianie wszystkich innych programów (usług i aplikacji), stanowi podstawę współpracy sprzętu z oprogramowaniem,
- oprogramowanie usługowe, utrzymania lub administracyjne, czyli programy stanowiące uzupełnienie systemu operacyjnego,
- pakiety oprogramowania lub oprogramowanie standardowe, stanowiące ofertę kompletną, niededykowaną dla konkretnego klienta,
- aplikacje biznesowe, udostępniające użytkownikowi konkretne usługi i funkcje, niezbędne do realizacji procesów biznesowych⁴.

Oprogramowanie jest specyficznym produktem. Jest produkowane w sposób jednostkowy. Natomiast niektóre typy oprogramowania podlegają sprzedaży

¹ *Informatyka dla ekonomistów. Studium teoretyczne i praktyczne*, praca zbiorowa pod red. A. Nowicki, PWN, Warszawa-Wrocław 1998, s. 151.

² P. Beynon-Davies, *Inżynieria systemów informacyjnych*, WNT, Warszawa 1999, s. 51.

³ *PN-ISO/IEC 25000:2008: Inżynieria oprogramowania. Wymagania jakości i ocena produktów programowych (SQuaRE). Przewodnik po SQuaRE*, s. 11.

⁴ *PN-ISO/IEC 27005:2010: Technika informatyczna. Techniki bezpieczeństwa. Zarządzanie ryzykiem w bezpieczeństwie informacji*, s. 36.

tw. „masowej”, zaś niektóre sprzedaży jednostkowej. Aby rozważyć możliwości zastosowania metod zarządzania jakością przez dostawców i odbiorców oprogramowania należy wobec tego rozważyć potencjalne cykle życia tego specyficznego produktu.

Cykle życia oprogramowania

Cykl życia oprogramowania jest istotny z punktu widzenia działań dostawcy, a równocześnie ma wpływ na możliwe działania zapewniające jakość oprogramowania.

Cykl życia (niekiedy nazywany cyklem trwania) oprogramowania nazywamy całość działań z nim związanych, od kierowania projektem systemu poprzez wdrażanie, modyfikację aż do likwidacji. Cykl życia jest to zestaw wszystkich etapów przez jakie przechodzi oprogramowanie, traktowane jako wyrób, począwszy od etapu przed projektowaniem a skończywszy na konieczności zaprzestania jego użytkowania.

Nie ma jednego określonego cyklu życia oprogramowania. Istnieje wiele typów cyklu. Najpopularniejsze z nich to^{5, 6}:

- kaskadowy, zwany cyklem wodospadu lub liniowym,
- realizacja kierowania dokumentami,
- prototypowanie,
- programowanie odkrywcze,
- realizacja przyrostowa,
- montaż z gotowych elementów,
- cykl spiralny,
- formalne transformacje,
- wielki wybuch.

Żaden z przedstawionych cykli nie jest optymalny. Zastosowanie poszczególnych cykli zależy od potencjalnych potrzeb przyszłych użytkowników. Cykle jak również ich elementy można łączyć ze sobą.

Cykl liniowy zwany inaczej klasycznym znajduje najlepsze zastosowanie dla firm, produkujących systemy o dobrze zdefiniowanych wymaganiach, z właściwie zdefiniowanymi wymaganiami przyszłych użytkowników. Przykładowy liniowy cykl życia systemu przedstawia rysunek 1. Polega na realizacji kolejno po sobie następujących faz.

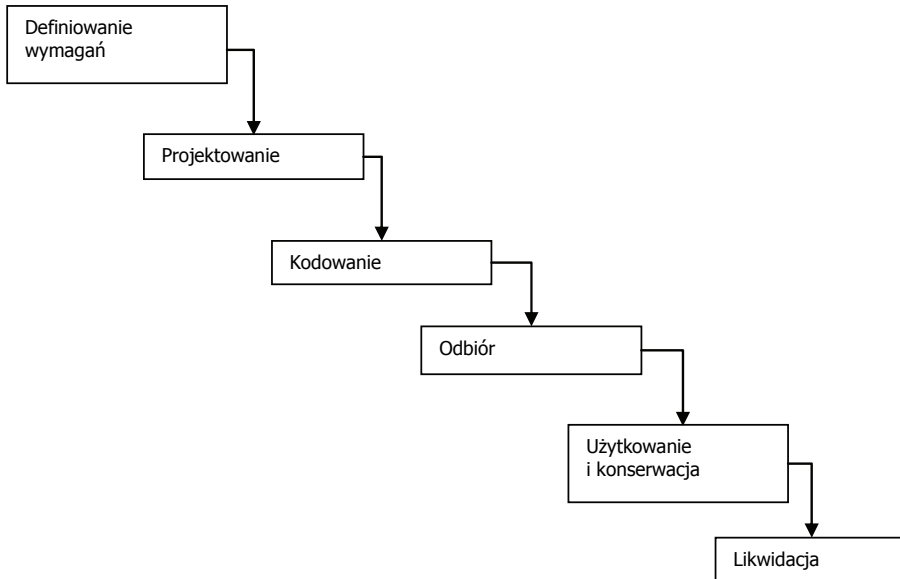
Klasyczny cykl liniowy (wodospadu) składa się z określonych etapów:

- definiowanie wymagań – wynikiem są określone funkcje, cele i wymagania wobec systemu,
- projektowanie – wynikiem jest szczegółowy projekt analizowanego systemu,
- kodowanie, niekiedy zwane implementacją – oznaczana, że projekt zostaje wykonany w określonym środowisku programistycznym,
- odbiór – oznacza sprawdzanie poszczególnych modułów, połączeń pomiędzy nimi oraz sprawdzanie całości systemu,

⁵ A. Jaszkiwicz, *Inżynieria oprogramowania*, Wydawnictwo Helion, Gliwice 1997, s. 15-27.

⁶ R. Patton, *Testowanie oprogramowania*, Wydawnictwo Mikom, Warszawa 2002, s. 44-48.

- użytkowanie i konserwacja – etap, w którym oprogramowanie jest wykorzystywane przez użytkownika a producent dokonuje konserwacji, wprowadza zmiany, poprawki i aktualizacje,
- likwidacja – jest to ostatni etap, w którym działanie oprogramowania jest „wygaszane”; dane oprogramowanie nie będzie użytkowane z różnych powodów np. zmiany na inny rodzaj oprogramowania.



Rys. 1. Liniowy cykl życia oprogramowania

Źródło: opracowanie własne na podstawie: A. Jaszkievicz, *Inżynieria oprogramowania*, Wydawnictwo Helion, Gliwice 1997, s. 23.

Cykl liniowy był pierwszym cyklem, który został opracowany na potrzeby informatyki. Obecnie jest stosowany bardzo rzadko. Przydatny jest do opracowywania oprogramowania na indywidualne zamówienia. Z tym cyklem związane są takie cechy realizacji projektu jak krótki czas realizacji oraz duża indywidualizacja oprogramowania. Z tych względów obecnie cykl jest rzadko stosowany.

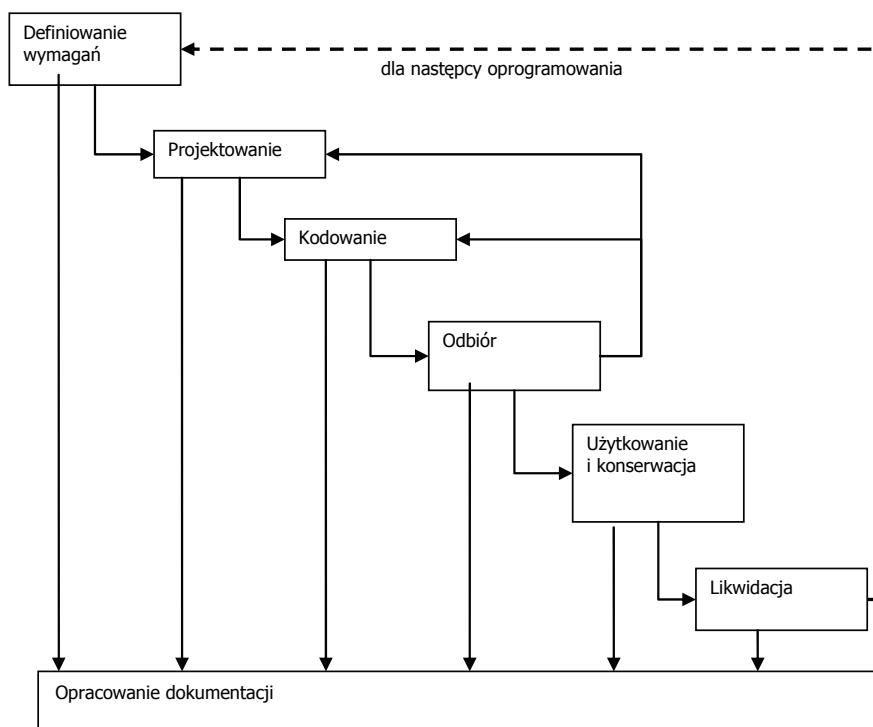
Jego przeciwieństwo stanowi cykl spiralny. Cykl spiralny składa się z iteracyjnie realizowanych czterech etapów⁷: planowanie, analiza ryzyka, konstruowanie kolejnej wersji systemu, weryfikacja. Planowanie jest działaniem realizującym dwa rodzaje planów, w zależności od numeru wersji: planowanie wstępnych wymagań i planowanie kolejnej wersji systemu. Analiza ryzyka może być także przeprowadzona z dwóch różnych punktów widzenia, w zależności od numeru wersji. Dla pierwszej wersji jest oparta na wstępnych wymaganiach i przeprowadzona z punktu widzenia autora oprogramowania. Kolejna wersja uwzględnia reakcję użytkownika i służy jako podstawa do

⁷ A. Jaszkievicz, *Inżynieria oprogramowania*, Wydawnictwo Helion, Warszawa 1997, s. 15-27.

zmian w następnych etapach. Konstruowanie kolejnej wersji systemu wykonywane jest zgodnie z cyklem liniowym. Dla kolejnych wersji powstaje następna oprogramowania (upgrade). Weryfikacja polega na ocenie kolejnej wersji oprogramowania przeprowadzanej przez użytkownika. Według cyklu spiralnego powstaje oprogramowanie tzw. „z półki”, niededykowane dla konkretnego klienta, tylko skierowane na potrzeby klienta masowego.

Możliwych jest wiele modyfikacji cyklu życia. Dla oprogramowania kustomizowanego stosowane są modyfikacje zarówno cyklu spiralnego jak i liniowego. Jedną z możliwych modyfikacji została pokazana na rysunku 2. Na rysunku zostały pokazane logiczne następstwa związane z działalnością wymiany oprogramowania na inny typ. Są możliwe dwa podejścia do pierwszej fazy cyklu, czyli definiowania wymagań:

- jeśli po raz pierwszy jest wdrażanie w przedsiębiorstwie oprogramowanie wówczas wymagania powstają w oparciu o doświadczenie dostawcy i wiedzę użytkownika czyli odbiorcy oprogramowania,
- jeśli oprogramowanie jest wymieniane na kolejną wersję lub na inny rodzaj wówczas można korzystać z doświadczeń wyniesionych z poprzednio użytkowanego oprogramowania.



Rys. 2. Modyfikacja cyklu liniowego
Źródło: opracowanie własne.

Fazy wymienione na rysunku 2 są poszerzone dodatkowo o równoległą fazę zwaną opracowaniem dokumentacji. Dokumentacja ma istotne znaczenie niezależnie od rodzaju cyklu życia oprogramowania; powinna być opracowywana w każdej fazie, nawet w fazie likwidacji

Jakość oprogramowania dla dostawcy i odbiorcy

Dostawca oprogramowania jest to jednostka organizacyjna, która dostarcza użytkownikowi oprogramowanie oraz świadczy usługi związane z oprogramowaniem w dowolnej fazie cyklu życia.

Odbiorca oprogramowania jest to użytkownik pośredni lub bezpośredni oprogramowania.

Udział dostawcy i odbiorcy oprogramowania w kolejnych fazach cyklu życia jest różny. Największy aktywny udział odbiorcy jest w fazach: definiowanie wymagań, odbiór oprogramowania, użytkowanie i konserwacja. Najmniejszy a nawet na poziomie zerowym udział jest w fazie projektowania i kodowania.

Najistotniejszym dla odbiorcy oprogramowania jest to aby ten specyficzny wyrób spełniał wszystkie wymagania przez cały czas użytkowania. Złej jakości oprogramowanie może wygenerować wiele strat o różnym charakterze, począwszy od strat czasowych związanych z niemożnością wykorzystania funkcji wyrobu a skończywszy na potencjalnej odpowiedzialności cywilno-karnej. Do najczęstszych negatywnych skutków należą:

- znaczny wzrost kosztów dodatkowego przetwarzania danych,
- znaczny wzrost czasu potrzebnego do analizy informacji, który może prowadzić do tzw. kosztów utraconych korzyści (czyli brak możliwości wykorzystania sprzyjającej sytuacji z powodu braku informacji lub złego jej przetwarzania,
- nadmierny koszt zakupu oprogramowania nie wykorzystywanego lub wykorzystanego tylko w pewnych obszarach,
- utrata informacji i kosztów jej odzyskania,
- wystąpienie błędów w przetwarzaniu informacji, kosztów kontroli i ich usuwania oraz ewentualnych kar z tego powodu,
- utrata zaufania klientów z powodu błędów oraz długiego oczekiwania na produkt lub realizację usługi,
- odpowiedzialność karno-skarbowa,
- a nawet w ostateczności likwidacja firmy.

Odbiorca pozyskując oprogramowanie oczekuje, że nabycie tego wyrobu nie pociągnie za sobą niepożądanych konsekwencji. Dla wyrobów o charakterze materialnym istnieją okresy gwarancji, które pozwalają na pewne poczucie bezpieczeństwa dla odbiorcy. Gwarancja na oprogramowanie w zależności od jego rodzaju ma różną długość i różne rodzaje odpowiedzialności. Prawodawstwo polskie nie ma uregulowanych przepisów dotyczących odpowiedzialności za oprogramowanie. Stąd brak poczucia bezpieczeństwa dla odbiorcy. Dodatkowym problemem w przypadku oprogramowania są zmiany dokonywane na nim w fazie konserwacji i użytkowania. Jest to kolejna różni-

ca pomiędzy środkiem trwałym niematerialnym, jakim jest oprogramowanie a środkami trwałymi materialnymi, które mają zapewnioną stałą funkcjonalność podczas użytkowania. Instalowanie kolejnych wersji może spowodować awarie prawidłowo dotychczas działających funkcjonalności. Stąd istotne dla odbiorcy jest wiarygodność dostawcy, który zagwarantuje odpowiednią jakość kolejnych wersji.

Z kolei dostawca powinien zapewnić, że dostarczany produkt, czyli oprogramowanie oraz jego kolejne wersje spełniają wymagania odbiorcy. Oprogramowanie jest produkowane w sposób jednostkowy, ale każda wersja w przeciwieństwie do wyrobów materialnych może być „poprawiona”. Jednakże dostawca nie jest w stanie zweryfikować wszystkich funkcjonalności oprogramowania. Wobec tego istotne jest odpowiednie działanie systemu zarządzania organizacji oraz zastosowanie metod zarządzania jakością.

Metody i narzędzia zarządzania jakością

Wszystkie idee doskonalenia jakości: TQM czy normy serii 9000:2000 wskazują na zasadność stosowania narzędzi doskonalenia jakości, służących zarówno do zbierania danych, ich analizowania lub też rozwiązywania problemów. Narzędzia z jednej strony podnoszą skuteczność zarządzania i przyspieszają proces wprowadzania zmian a z drugiej strony wspomagają działania jakościowe w każdym z obszarów zarządzania w przedsiębiorstwie.

W literaturze dotyczącej zarządzania jakością stosowane są różne nazwy, takie jak: metody, techniki, narzędzia, zasady, sposoby doskonalenia jakości. Jedną z propozycji podziału instrumentów oddziaływania na jakość zawarli w swojej pracy Hamrol i Mantura⁸. Dokonali oni podziału na zasady, metody i narzędzia zarządzania jakością. W tabeli 1 zestawiono te instrumenty, które najczęściej znajdują zastosowanie w doskonaleniu jakości wyrobów i usług.

Tabela 1. Podstawowe zasady, metody i narzędzia zarządzania jakością

	Cechy i sposób oddziaływania na jakość	Przykłady
Zasady Zarządzania Jakością	Oddziaływanie długotrwałe kształtujące strategię i kulturę przedsiębiorstwa Wykraczają poza ramy przedsiębiorstwa Nie dają wytycznych operacyjnych Rezultaty stosowania są trudne do oceny bieżącej	Zasada Deminga Zasada „ciągłego doskonalenia procesów” (Kaizen) Zasada „zero defektów” Zasada pracy zespołowej
Metody Zarządzania Jakością	Oddziaływanie „średnioterminowe” Pozwalają kształtować przede wszystkim jakość projektową i jakość wykonania Opierają się na ogólnie przyjętych algorytmach postępowania.	QFD FMEA Projektowanie eksperymentów Metody Taguchi i Shainina

⁸ A. Hamrol, W. Mantura. *Zarządzanie jakością. Teoria i praktyka*, Wydawnictwo Naukowe PWN, Warszawa, 2002, s. 201.

Narzędzia Zarządzania Jakością, z grupy narzędzi tradycyjnych	Podają proste, efektywne sposoby do bezpośredniego wykorzystania i oddziaływania na jakość w różnych fazach procesu. Mogą być stosowane samodzielnie lub w połączeniu z metodami. Wykorzystywane są głównie do analizy już istniejących problemów	Diagram Ishikawy Diagram Pareto-Lorenza Schematy blokowe Diagram korelacji Histogramy Karty kontrolne
Narzędzia Zarządzania Jakością, z grupy narzędzi nowych	Stanowią „wsparcie” dla metod zarządzania jakością. Wymagają pracy zespołowej.	Diagram relacji Diagram pokrewieństwa Diagram drzewa Diagram macierzowy Macierzowa analiza danych Diagram PDPC Diagram strzałkowy
SPC - Statystyczne sterowanie procesem	Kształtowanie jakości wyrobów i procesów w fazie produkcyjnej	Statystyczne sterowanie procesem produkcyjnym lub usługowym.

Źródło: A. Hamrol, W. Mantura. *Zarządzanie jakością. Teoria i praktyka*, Wydawnictwo Naukowe PWN, Warszawa, 2002, s. 201.

W literaturze można spotkać inną klasyfikację⁹:

- Siedem starych narzędzi, czyli: lista zbiorcza, histogram, diagram korelacji, karty kontrolne, diagram przyczynowo-skutkowy, analiza Pareto-Lorenza, schemat blokowy,
- Siedem nowych narzędzi, czyli: diagram pokrewieństwa, diagram zależności, diagram drzewa, diagram tablicowy, tablicowa analiza danych, diagram planowania procesu decyzyjnego,
- Siedem narzędzi wspomagających, czyli: analiza pola sił, burza mózgów, technika kwestionariuszowa, QFD - dom jakości, benchmarking, analiza SWOT, modele organizacyjne,
- Inne narzędzia.

Bez względu na rodzaj klasyfikacji zasady, metody i narzędzia doskonalenia jakości mogą być zastosowane w każdym przedsiębiorstwie, bez względu na wielkość i rodzaj działalności. Kryteria przydatności danej metody czy narzędzia to:

- efekt, który można osiągnąć dzięki ich zastosowaniu,
- sytuacja, w której dane narzędzia lub metoda będą użyte.

W przedsiębiorstwach działających w branży informatycznej szczególną uwagę powinno zwracać faza definiowania wymagań. Produkt - oprogramowanie, który powstaje, opiera się głównie na wymaganiach klienta. Od fazy definiowania wymagań zależy w dużym stopniu efekt końcowy czyli zadowolenie i satysfakcja klienta.

Dla odbiorców oprogramowania najbardziej istotne są metody, które można zastosować w fazie odbiorów.

⁹ K. Szczepańska, *Techniki menadżerskie w TQM*, Wydawnictwo Normalizacyjne ALFA-WERO, Warszawa 1999, s. 58.

W tabeli 2 zostały przyporządkowane wybrane narzędzia i metody zarządzania jakością do zastosowania w poszczególnych fazach cyklu życia oprogramowania.

Tabela 2. Przyporządkowanie metod zarządzania jakością do faz cyklu życia oprogramowania

Nazwa fazy cyklu życia oprogramowania	Metody zarządzania jakością stosowane przez dostawców oprogramowania	Metody zarządzania jakością stosowane przez odbiorców oprogramowania
Definiowanie wymagań	QFD FMEA projektowanie eksperymentów burza mózgów technika kwestionariuszowa benchmarking modele organizacyjne	
Projektowanie	schemat blokowy diagram sieciowy diagram planowania procesu decyzyjnego QFD FMEA	-----
Kodowanie	lista zbiorcza histogram karty kontrolne diagram przyczynowo-skutkowy testowanie	-----
Odbiór	lista zbiorcza histogram diagram przyczynowo-skutkowy analiza Pareto-Lorenza testowanie	lista zbiorcza histogram karty kontrolne analiza Pareto-Lorenza diagram przyczynowo-skutkowy testowanie
Użytkowanie i konserwacja	lista zbiorcza histogram karty kontrolne analiza Pareto-Lorenza diagram przyczynowo-skutkowy testowanie	
Likwidacja	QFD FMEA projektowanie eksperymentów burza mózgów technika kwestionariuszowa benchmarking modele organizacyjne	

Źródło: opracowanie własne.

Największy wpływ na jakość wyrobu – oprogramowania mają metody, które występują w fazie definiowania wymagań i projektowania. Metody te służą głównie doskonaleniu jakości na etapie przedprojektowym. Celem tych

metod jest uzyskanie wymaganej jakości projektowej wyrobu. Ze względu na sposób i miejsce stosowania metod projektowania dla jakości wyróżnia się w cyklu istnienia wyrobu:

- metody projektowania parametrów wyrobu lub procesu,
- metody zapobiegania wadom wyrobu oraz procesu (metody prewencyjne).

Metody projektowania parametrów służą do identyfikacji czynników, które najsilniej oddziałują na jakość wyrobu lub procesu. Wśród metod projektowania parametrów na szczególną uwagę zasługuje metoda QFD oraz metody związane z projektowaniem eksperymentów (np. metoda Taguchi).

QFD

QFD – dom jakości jest metodą planowania jakości. Uwzględnia wpływ wymagań klienta-odbiorcy, dotyczących produktu począwszy od fazy badań i rozwoju poprzez wytwarzanie, aż do sprzedaży i serwisu. QFD w pewien sposób „tłumaczy” wymagania klienta – na język inżynierów. Stanowi źródło informacji o produkcji oraz procesie. Charakterystycznym narzędziem wykorzystywanym w QFD jest Dom Jakości (House of Quality).

Metoda QFD posiada swoją odmianę, która może być wykorzystywana w projektach informatycznych - SQFD (Software Quality Function Deployment). Metoda ta kładzie nacisk na wymagania klienta, które stanowią podstawę każdego oprogramowania. W przypadku oprogramowania wyjaśnienie wszystkich elementów w fazach definiowania wymagań i projektowania jest znacznie tańsze niż w kolejnych fazach.

Minusem tej metody jest to, że jest kosztowna i czasochłonna, wymagająca od zespołu wiedzy, kompetencji i doświadczenia.

FMEA

Metody prewencyjne umożliwiają wykrycie i usunięcie już na etapie projektowania procesu lub wyrobu, ewentualnych wad i błędów, których skutki mogłyby być rozpoznawane dopiero podczas produkcji lub eksploatacji wyrobu. Najbardziej rozpowszechnioną metodą prewencyjną jest metoda FMEA.

Metodę FMEA wykorzystuje się w celu eliminowania wad wyrobu (konstrukcji wyrobu) lub procesu poprzez rozpoznawanie przyczyn ich powstawania i stosowania odpowiednich środków zapobiegawczych. FMEA umożliwia unikanie wystąpienia rozpoznanych, a także jeszcze nieznanymi wad w nowych wyrobach i procesach. Polega na wykonaniu analizy ryzyka przed procesem projektowania. Taka analiza ryzyka pozwala uniknąć potencjalnych przyszłych problemów, które są analizowane przez odbiorcę i dostawcę. Jest wykorzystywana na etapie projektowania, gdzie powstaje większość błędów, które ujawniają się dopiero w kolejnych fazach.

W procesach informatycznych jest stosowana odmiana FMEA - metoda FMECA (Failure Modes, Effects and Criticality Analysis).

Testowanie

Metodą, która jest identyfikowana wyłącznie z informatyką jest testowanie. Niekiedy w literaturze testowanie może oznaczać:

- metodę stosowaną do zapewniania jakości dla produktów informatycznych
- fazę odbioru; w tej fazie też jest wykorzystywana metoda testowania.

Testowanie jako metoda jakościowa może mieć na celu:

- weryfikację wymagań odbiorcy oprogramowania,
- sprawdzenie poprawności działania elementu lub gotowego oprogramowania, w stosunku do projektu.

Testowaniu mogą być poddawane dowolne obiekty: projekt oprogramowania, oprogramowanie, jego część lub prototyp. Programy posiadają swoje unikalne funkcje oraz wiele kombinacji funkcji. Każda kombinacja ma inny przebieg, który powinien być przetestowany aby mieć pewność, że stworzony program wykonuje to co jest od niego wymagane. Duża ilość funkcji oprogramowania powoduje, że ilość tych kombinacji rośnie, osiągając niekiedy liczbę wielu tysięcy. Stąd wszystkie te przebiegi powinny być testowane. Testowanie może okazać się bardzo poważnym i obszernym zadaniem, zwłaszcza kiedy ilość możliwych kombinacji nie jest dokładnie określona. Standardowo tylko niewielka część programu komputerowego jest naprawdę testowana.

Testowanie jako metoda może być przeprowadzane za pomocą różnorodnych technik. Techniki testów oprogramowania są dzielone na wiele sposobów ze względu na: przyjętą strategię prowadzonego testowania, metodę, cel, obiekt, jaki jest poddany procesowi testowania, oraz osoby, który przeprowadzają i uczestniczą w procesie testowania.

Modele i normy zarządzania jakością w procesach informatycznych

Metody i narzędzia mogą wspomagać działania jakościowe lokalnie, czyli w danej fazie. Dużo większe znaczenie ma systemowe rozwiązanie problemów, związanych z jakością. W tym celu były opracowane modele i normy. Wśród takich modeli i norm można wyróżnić:

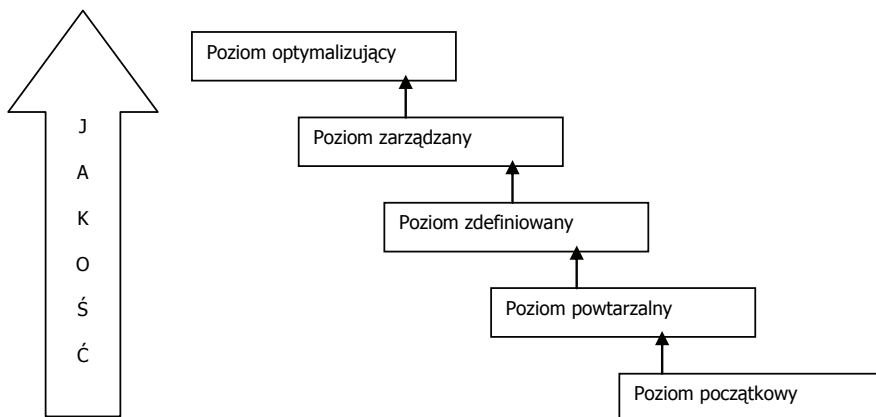
- Model CMM (Capability Maturity Model – model dojrzewania zdolności),
- Model RUP (Rational Unified Process),
- Seria norm 20000, dotyczących procesów zarządzania usługami, dedykowana dla przedsiębiorstw o charakterze informatycznym,
- Norma PN-ISO/IEC 90003:2007, zawierająca wytyczne stosowania 9001:2000 do oprogramowania komputerowego.

CMM

Capability Maturity Model jest to metoda opracowana przez Software Engineering Institute w Carnegie-Mellon University. CMM opisuje podział przedsiębiorstw prowadzących działalność informatyczną na pięć typów ze względu

du zarządzanie jakością (rys. 3). Podstawą CMM jest przyporządkowanie producentów oprogramowania do jednego z pięciu głównych poziomów zaawansowania¹⁰:

- Początkowego – przedsiębiorstwo nie ma procedur zarządzania lub nie wykorzystuje ich w odpowiedni sposób, dostawca oprogramowania nie jest w stanie przewidzieć czasu i budżetu projektu,
- Powtarzalnego – brak formalnego modelu procesu zarządzania jakością, są opracowane wybrane procedury np. zarządzanie konfiguracją, dostawca nie musi stosować jednolitego zarządzania wszystkimi projektami,
- Zdefiniowanego – proces zarządzania został zdefiniowany, ale może się odnosić indywidualnie do każdego projektu,
- Zarządzanego – dodatkowo przedsiębiorstwo uruchamia program zbierania danych ilościowych,
- Optymalizującego – przedsiębiorstwo ciągle doskonali swoje procesy.



Rys. 3 Model CMM

Źródło: opracowanie własne na podstawie I. Sommerville, *Inżynieria oprogramowania*.

Model CMM pozwala nie tylko na ciągłe doskonalenie procesów w przedsiębiorstwie informatycznym ale poprzez ocenę procesów produkcji oprogramowania, na ocenę zdolności potencjalnych dostawców oprogramowania. Czyli w pewien sposób pozwala zwiększyć bezpieczeństwo odbiorcy.

W literaturze model CMM porównywany jest do serii norm ISO 9000.

RUP

RUP (Rational Unified Proces) – jest to sposób projektowania oprogramowania (metodyka obiektowa), opracowany w postaci szablonu pozwalającego na dostosowanie do charakteru konkretnego przedsiębiorstwa, konkret-

¹⁰ I. Sommerville, *Inżynieria oprogramowania*, WNT, Warszawa 2003, s. 568.

nego zespołu projektowego lub nawet charakteru konkretnego projektu. Ma postać zestawu najlepszych praktyk tworzenia oprogramowania¹¹. RUP wymaga dwuwymiarowego cyklu życia oprogramowania. Zaletą modelu jest iteracyjny sposób dostosowania projektu do wymagań funkcjonalnych oraz do zmian w trakcie trwania rozwoju oprogramowania.

Norma PN-ISO/IEC 90003:2007

Norma PN-ISO/IEC 90003:2007, zawiera wytyczne stosowania 9001:2000 do nabywania, dostaw, wytwarzania, obsługi, i utrzymywania oprogramowania komputerowego¹². Podaje i definiuje szereg działań, które powinno podjąć każde przedsiębiorstwo, niezależnie od stosowanych przez siebie procesów, cykli życia oprogramowania czy posiadanej struktury organizacyjnej. W pewnym uproszczeniu norma jest skierowana do dostawców oprogramowania komputerowego.

Według normy przedsiębiorstwa ma zaplanować i wytworzyć procesy potrzebne do realizacji oprogramowania. A więc ma określić m.in.:

- cele dotyczące jakości i wymagania dotyczące oprogramowania,
- procesy, dokumenty i zasoby,
- wymagania dotyczące weryfikacji, walidacji, monitorowania, kontroli i testowania,
- kryteria akceptacji oprogramowania.

Zwraca uwagę na znaczenie modeli cykli życia oprogramowania dla projektowania oprogramowania.

W wymaganiach związanych z oprogramowaniem dostawca musi oprócz wymagań podanych przez klienta, zdefiniować także istotne wymagania których nie podał klient, a które mogą mieć znaczenie dla produktu.

Zostało także poruszone problemy ryzyk związanych z niemożnością dostarczenia oprogramowania odbiorcy, niską jakością, bezpieczeństwem i zabezpieczeniami.

Norma wymienia także możliwość zastosowania testowania w procesie walidacji.

Normy serii 20000

Normy serii 20000 (PN-ISO/IEC 20000-1:2007, PN-ISO/IEC 20000-2) są ukierunkowane nie na produkt ale realizację usługi. Standard ISO 20000 kładzie nacisk na stosowanie zintegrowanych procesów w celu efektywnego świadczenia usług informatycznych. Zawiera wytyczne dotyczące jakości zarządzania usługami informatycznymi.

Seria norm składa się z 2 norm: specyfikacji i reguł postępowania w zarządzaniu usługami informatycznymi.

Norma PN-ISO/IEC ISO 20000-1 Zarządzanie usługami określa wymagania stawiane dostawcom usług w celu zapewnienia określonego, ustalonego po-

¹¹ B. Jayaswal, P. Patton, *Oprogramowanie godne zaufania. Metodologia, techniki i narzędzia projektowania*, Helion, Gliwice 2009, s. 54.

¹² PN-ISO/IEC 90003:2007, *Inżynieria oprogramowania. Wytyczne stosowania 9001:2000 do oprogramowania komputerów*, s. 7.

ziomu jakości ich usług. Wg tego niej, na zarządzanie usługą, składa się z procesów, takich jak m.in.:

- zarządzanie poziomem usługi (SLA),
- sporządzanie raportu usługi,
- zarządzanie ciągłością i dostępnością usługi,
- zarządzanie potencjałem wykonawczym,
- zarządzanie bezpieczeństwem informacji,
- zarządzanie zmianą i inne.

Wdrażanie przez dostawcę usług tej normy ma zagwarantować odbiorcy usługi, że dostawca usługi będzie świadczył ją na ustalonym poziomie, w ustalonym czasie, zapewniając jej dostępność oraz bezpieczeństwo. Norma ma znaczenie też dla dostawcy oprogramowania z uwagi na takie procesy jak dostarczanie kolejnych wersji oprogramowania czy świadczenie usług konserwacji.

Podsumowanie

Oprogramowanie jest środkiem trwałym niematerialnym. Jest znacznie bardziej złożone od środków trwałych materialnych, ponieważ zawiera znacznie większą liczbę funkcji, które mogą być stosowane w wielu kombinacjach. To powoduje, że oprogramowanie zawiera znacznie więcej stanów niż środki materialne. Oprogramowanie aby funkcjonować jest zależne od innych składników systemu informatycznego m.in. od sprzętu i ludzi.

Ponadto większość defektów występujących w środkach materialnych można naprawić przez dokonanie zmian w projekcie czy procesie produkcji¹³. Natomiast błędów związanych z oprogramowaniem nie można naprawić poprzez zmianę procesu produkcji. Naprawić można oprogramowanie poprzez kolejną produkcję następnej wersji oprogramowania. Produkcja oprogramowania jest ewidentnie procesem jednostkowym.

Defekty występujące w oprogramowaniu mogą być generowane we wszystkich fazach cyklu życia oprogramowania; także w fazie użytkowania. Oprogramowanie ma działać zgodnie z wymaganiami użytkownika - odbiorcy. Tymczasem bardzo często odbiorca nie jest jedną osobą. To samo oprogramowanie w jednym przedsiębiorstwie często jest użytkowane przez kilka-kilkaset a nawet tysiące użytkowników.

Stąd tak istotne znaczenie dla jakości oprogramowania ma zastosowanie szeregu zasad, wytycznych, metod i narzędzi zarządzania jakością. Odbiorca oprogramowania musi mieć gwarancję jego działania w odpowiedni sposób przez cały okres użytkowania.

¹³ B. Jayaswal, P. Patton, *Oprogramowanie godne zaufania. Metodologia, techniki i narzędzia projektowania*, Helion, Gliwice 2009, s. 41.

Bibliografia

- Beynon-Davies P., *Inżynieria systemów informacyjnych*, WNT, Warszawa 1999.
- Hamrol A., Mantura W., *Zarządzanie jakością. Teoria i praktyka*, Wydawnictwo Naukowe PWN, Warszawa 2002.
- Informatyka dla ekonomistów. Studium teoretyczne i praktyczne*, praca zbiorowa pod red. A. Nowicki, PWN, Warszawa-Wrocław 1998.
- Jaszkiewicz A., *Inżynieria oprogramowania*, Wydawnictwo Helion, Warszawa 1997.
- Jayaswal B., Patton P., *Oprogramowanie godne zaufania. Metodologia, techniki i narzędzia projektowania*, Helion, Gliwice 2009.
- Patton R., *Testowanie oprogramowania*, Wydawnictwo Mikom, Warszawa 2002.
- PN-ISO/IEC 20000-2:2007 *Technika informatyczna, Zarządzanie usługami*.
- PN-ISO/IEC 27005:2010: *Technika informatyczna. Techniki bezpieczeństwa. Zarządzanie ryzykiem w bezpieczeństwie informacji*.
- PN-ISO/IEC 90003:2007, *Inżynieria oprogramowania. Wytoczne stosowania 9001:2000 do oprogramowania komputerów*.
- PN-ISO/IEC 25000:2008: *Inżynieria oprogramowania. Wymagania jakości i ocena produktów programowych (SQuaRE). Przewodnik po SquaRE*.
- Sommerville I., *Inżynieria oprogramowania*, WNT, Warszawa 2003.
- Szczepańska K., *Techniki menadżerskie w TQM*, Wydawnictwo Normalizacyjne ALFA-WERO Sp. z o.o., Warszawa 1999.