

Karolina Zmitrowicz



JAKOŚĆ PROJEKTÓW INFORMATYCZNYCH

ROZWÓJ I TESTOWANIE OPROGRAMOWANIA

Helion

one
p r e s s

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Opieka redakcyjna: Ewelina Burska

Projekt okładki: Studio Gravite/Olsztyn
Obarek, Pokoński, Pazdrijowski, Zaprucki

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/zapeja>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-0156-6

Copyright © Helion 2015

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

Przedmowa	7
Rozdział 1. Wprowadzenie	9
Rozdział 2. Koncepcja jakości	11
Definicja jakości	11
Normalizacja	15
Znaczenie jakości w projektach informatycznych	16
Koszty jakości	21
Rozdział 3. Zarządzanie jakością	35
Zarządzanie procesowe	35
Zarządzanie jakością	40
Zarządzanie przez jakość	56
Koncepcje zarządzania jakością	63
Zasady Deminga	63
Koła jakości	75
Inne koncepcje, narzędzia i techniki zarządzania jakością	79
Zarządzanie jakością oprogramowania	80
Jakość oprogramowania	80
Kodeks postępowania	85
Manifest jakości	87
Standardy	88
ISO 9000 Quality Management	89
ISO 19011: 2011 — Guidelines for auditing management systems	91
ISO/TS 16949: 2009 — Quality management systems — Particular requirements for the application of ISO 9001: 2008 for automotive production and relevant service part organizations	91
TickIT i TickIT plus	91
ISO Technical Report 19759 (SWEBOK®)	93
Rozdział 4. Zapewnienie jakości	95
Wprowadzenie	95
Planowanie procesu zapewnienia jakości	99
Plan zapewnienia jakości	99
Czynniki wpływu	105
Charakterystyki jakościowe dla procesu i produktu	106

Weryfikacja i walidacja	118
Przeglądy	124
Listy kontrolne	137
Metryki	143
Anomalie — charakterystyka i sposób obsługi	148
Standardy	159
ISO/IEC 25000: 2005 — Software Engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE	159
ISO 9241 — Ergonomics of Human System Interaction	159
ISO 31000: 2009 — Risk Management — Principles and guidelines	160
IEEE 610.12: 1990 — Standard Glossary of Software Engineering Terminology ...	160
IEEE 828: 2012 — Standard for Configuration Management in Systems and Software Engineering	160
IEEE 830: 1998 — Recommended Practice for Software Requirements Specifications	161
IEEE 1233: 1996 — Guide for Developing of System Requirements Specifications	161
IEEE 1362: 1998 — Guide for Information Technology — System Definition — Concept of Operations (ConOps) Document	161
IEEE 29148: 2011 — Systems and software engineering — Life cycle processes — Requirements engineering	161
IEEE 730: 2002 — Standard for Software Quality Assurance Plans	162
IEEE 1012: 1986 — Standard for Software Verification and Validation Plans	162
IEEE 1028: 2008 — Standard for Software Reviews and Audits	162
IEEE 1044: 2009 — Standard Classification for Software Anomalies	162
IEEE 1061: 1998 — Standard for a Software Quality Metrics Methodology	162
Rozdział 5. Testowanie	163
Podstawy testowania	167
Organizacja testów	170
Niezależność testowania	170
Kontekst testowania	173
Strategia testów	174
Poziomy testów	179
Cele testowania	183
Techniki testowe	184
Techniki oparte na intuicji i doświadczeniu	186
Techniki oparte na specyfikacji	187
Techniki oparte na kodzie	195
Techniki oparte na usterkach	201
Techniki oparte na użyciu	202
Techniki oparte na charakterze systemu	202
Proces testowy	203
Podstawowy proces testowy	203
Metryki związane z testowaniem	214
Ocena produktu poddawanego testom	215
Ocena wykonywanych testów	216
Dokumentacja testów	217
Dokumentacja zarządcza	220
Dokumentacja specyfikacji testów	226
Dokumentacja wykonania testów	232
Dokumentacja raportów z testów	233

Wsparcie narzędziowe	236
Standardy w testowaniu	237
BS 7925-1: 1998 — Software testing — Vocabulary	237
BS 7925-2: 1998 — Software testing — Software component testing	237
IEEE 1008: 1987 — Standard for Software Unit Testing	237
IEEE 829: 1998 — Standard for Test Documentation	237
ISO/IEC/IEEE 29119 — Software Testing	238
Normy procesowe	244
Inne standardy	244
Rozdział 6. Doskonalenie jakości	247
Doskonalenie procesów organizacyjnych	248
CMMI®	248
TickITplus	254
ISO/IEC 15504 — Software Process Improvement and Capability Determination (SPICE)	254
Doskonalenie procesu testowego	256
IDEAL	257
TMMi®	258
TPI® Next	269
CTP	271
Inne modele doskonalenia procesu testowego	274
Lean Software Development (LSD)	275
Zasady LSD	276
Rozdział 7. Podsumowanie	281
Literatura	283
Skorowidz	291

Rozdział 2.

Koncepcja jakości

Jakość to pewien stopień doskonałości

— Platon

Definicja jakości

Tematu zapewnienia jakości oraz testowania nie sposób rozpocząć bez wprowadzenia pojęcia jakości. Termin ten przewija się w niemalże wszystkich znanych człowiekowi obszarach — mówimy o jakości usług, produktów, procesów, ale i o jakości samego życia. Można powiedzieć, że w ostatnich czasach zapanowała swoista moda na jakość, ponieważ wraz z rozwojem technologicznym, gospodarczym i społecznym konsumenci oczekują lepszego spełniania ich oczekiwań i zaspokajania potrzeb. Klienci wymagają wysokiej jakości, producenci i usługodawcy dążą do spełnienia oczekiwań (lub przynajmniej sprawiania takiego wrażenia) przez dostarczanie dóbr o nowej czy innowacyjnej jakości. Jakość jest pojęciem modnym. Czym jednak jest i jak ją zdefiniować?

Z definicją jakości wiążą się pewne mity. Wielu ludzi stoi na stanowisku, że jakość jest pojęciem wysoce zależnym od percepcji poszczególnych jednostek, co przekłada się na niemierzalność i nieokreśloność. Według dość popularnego twierdzenia, ponieważ definicja jakości zależy od postrzegania danej osoby (co akurat jest prawdą), nie może być jednoznacznie określona, a co za tym idzie — zmierzona. Jest to niewątpliwie ciekawe podejście do tematu, jednak niezbyt zgodne ze stanem faktycznym.

Koncepcja jakości znana jest już od czasów starożytnych. Po raz pierwszy pojęcie znane dziś jako jakość zostało wprowadzone w *Kodeksie Hammurabiego* z 1750 r. p.n.e. W kodeksie tym istniał przepis, który nakazywał karać śmiercią murarza, jeśli zbudowany przez niego dom nie był odpowiedniej jakości — niska jakość w tym kontekście oznaczała dom, który zawala się i zabija mieszkańców.

W kolejnych wiekach pojęcie jakości było rozwijane w Grecji, kolebce filozofii i nauk przyrodniczych.

Pierwszą pisaną definicję jakości wprowadził grecki filozof Platon (427 – 347 p.n.e.), określając ją jako „pewien stopień doskonałości”. W tym ujęciu jakość była pojęciem filozoficznym, silnie związanym z koncepcją piękna i wartościowania danego przedmiotu przez użytkownika (Kiliński 1979).

Nieco inne podejście reprezentował Arystoteles (384 – 322 p.n.e.), definiując jakość jako „to, co sprawia, że rzecz jest rzeczą, którą jest” (Skrzypek 2000). Definicja Arystotelesa jest o tyle istotna, że kładzie nacisk na powiązanie jakości z przedmiotem, którego ta jakość dotyczy. Można tu znaleźć źródła nowoczesnych definicji określających jakość jako zgodność wyrobu z określonymi charakterystykami.

Filozoficzne postrzeganie jakości kontynuowano i rozszerzano w bliższych nam czasach. W XVII w. René Descartes (Kartezjusz) i John Locke opracowali i przyjęli koncepcję dualistycznego ujęcia jakości, według którego jakość objawia się w dwu płaszczyznach: jakość pierwotna tkwiąca w przedmiocie (np. waga, wymiary) oraz jakość wtórna wynikająca z postrzegania zmysłowego człowieka (np. kolor, smak).

Warto zwrócić uwagę na to, że wymienione dotychczas koncepcje jakości opierają się na podstawach filozoficznych, dość dalekich od obecnego postrzegania jakości — silnie związanego z technologią, rynkiem i aspektami biznesowymi. Ten stan rzeczy uległ zmianie w trakcie rewolucji przemysłowej.

Okolo XVIII w., wraz z rozwojem techniki i wprowadzeniem rozwiązań mających na celu zwiększenie produkcji, nastąpiła wyraźna zmiana postrzegania jakości. Jakość zaczęto kojarzyć z wartością. Ocenę jakości pozostawiono w gestii rynku — w nowoczesnym ujęciu to rynek (konsumenci) określał, co jest dobrej jakości, a co nie. Warto zwrócić też uwagę na to, że ocenie podlegał głównie stosunek jakości do ceny. W odróżnieniu od czasów wcześniejszych zainteresowano się ceną produktów i ich wartością postrzeganą przez rynek. Zmieniły się też definicje jakości.

Na przełomie XIX i XX w. wprowadzono seryjną produkcję i zaczęto wykorzystywać taśmy montażowe. Po raz kolejny zmieniło się postrzeganie jakości — tym razem jako zgodność ze specyfikacją. Dodatkowo pojawiły się aspekty kontroli jakości. Walter Andrew Shewhart (1891 – 1967), „ojciec” statystycznej kontroli jakości, postulował, aby definiować jakość produktu w taki sposób, by można ją było porównywać w różnych okresach.

Współcześnie dominujące definicje jakości określają ją najczęściej jako spełnienie lub przekroczenie wymagań klienta. Nacisk na spełnianie wymagań klienta wynika bezpośrednio ze zwiększenia produkcji i usług we wszystkich obszarach życia człowieka, co przekłada się na wyższy poziom życia społeczeństw i idące za tym wyższe oczekiwania względem konsumowanych produktów i usług. Dodatkowym, bardzo ważnym czynnikiem jest ogromna konkurencja nierzadko wymuszająca na przedsiębiorcach walkę o klienta.

Można się spodziewać, że współcześnie przyjęte definicje jakości będą ewoluować w celu jak najpełniejszego oddania znaczenia jakości w realiach globalnego społeczeństwa konsumpcyjnego.

Obecnie stosowane definicje jakości można sklasyfikować zgodnie z podejściem Davida A. Garvina (1984). Zaproponował on podział definicji jakości na siedem kategorii: ogólne, związane z produkcją, związane z produktem, związane z użytkownikiem, związane z tworzeniem wartości, wielowymiarowe, strategiczne (tabela 2.1).

Tabela 2.1. Klasyfikacja definicji jakości (Garvin 1984)

Jakość — definicje ogólne		
1931	W.A. Shewhart	Dobroć produktu, przy czym dobroć ta może być zastosowana do wszystkich rodzajów produktów i usług.
1951	J. Juran	Podział definicji jakości dla obszaru projektowania oraz zgodności.
1962	J. Juran	Spełnienie żądań (jakość rynkowa), jakość projektowania, jakość zgodności, preferencje klientów (przewaga nad konkurencją), jakość charakterystyk (funkcjonalność), generalna doskonałość (niesklasyfikowana gdzie indziej), funkcja lub odpowiedzialność związana z osiągnięciem jakości produktu, część organizacji odpowiedzialna za produkt (wydział).
1974	J. Juran	Dopasowanie do użytkownika — w 1988 r. nastąpiło rozszerzenie definicji o pojęcie klienta zewnętrznego i wewnętrznego.
Jakość — definicje związane z produkcją		
1979	P. Crosby	Zgodność z wymaganiami wewnętrznymi i zewnętrznymi.
Jakość — definicje związane z produktem		
	R. Shmalensee, J.H. Swan	Wytrzymałość, długie życie produktu; podnoszenie parametrów produktu jest równoznaczne z jakością.
1983	J. Feigenbaum	Zdolność do wykonywania zadań, działania, przydatność.
1984	D.A. Garvin	Właściwe wykonanie oraz dodatkowe wyposażenie.
1990	G. Taguchi, D. Clausing	Jakość jest związana z właściwym projektowaniem.
Jakość — definicje związane z użytkownikiem		
1991	L. Dobyns, C. Crawford- Manson	Spełnienie wymagań klienta.
1951	J. Juran	Przydatność dla użytkownika.
1987	J. Feigenbaum	Kompozycja charakterystyk, marketingu, produkcji, projektowania produktu lub usługi, która w użytkowaniu zaspokoi potrzeby klienta.
Jakość — definicje związane z tworzeniem wartości		
1982	I. Broh	Doskonałość lub przydatność do użytku po akceptowalnej cenie.
Jakość — definicje wielowymiarowe		
1984	D.A. Garvin	Jakość produktu — wykonanie, dodatkowe wyposażenie, zgodność, wytrzymałość, zdolność do działania, estetyka, postrzegana jakość.

Tabela 2.1. *Klasyfikacja definicji jakości (Garvin 1984) — ciąg dalszy*

Jakość — definicje wielowymiarowe (cd.)		
1991	A. Parasurman, L.L. Berry, A. Zeithami	Jakość w odniesieniu do usług — wykonanie części materialnej, niezawodność, reakcja na problemy, kompetencje pracowników, empatia.
1993	Nagroda Baldrige'a	Przywództwo, przepływ informacji i analizy, strategiczne planowanie jakości, rozwój zasobów ludzkich, zarządzanie procesami, wyniki, orientacja na klienta, satysfakcja klienta i pracowników.
Jakość — definicje strategiczne		
1980	M. Porter	Jedna z dróg do odróżnienia produktu od konkurencji — konieczna w obszarach istotnych dla klienta.
1981	R.D. Buzzell, F.D. Wiersma	Produkt, który przekracza jakością konkurentów, może zwiększyć swój udział w rynku.
1986	W.E. Deming	Produkt wyższej jakości może poprawić postrzeganie firmy przez klientów.

Ciekawą definicję jakości zaproponował Leszek Wasilewski (Blikle 2014): „Jakość produktu to miara braku wad w tym produkcie (im mniej wad, tym wyższa jakość), a wadą produktu jest każda taka negatywna cecha produktu — negatywna z punktu widzenia klienta — której klient ma się prawo nie spodziewać”. Definicja ta jest o tyle interesująca, że wyraźnie wskazuje na związek jakości z istnieniem wad w produkcie, co będzie szczególnie istotne w dalszych rozważaniach na temat jakości w IT.

Na zakończenie warto przytoczyć rozważania Garvina, który zdefiniował jakość za pomocą ośmiu wymiarów, pokazując tym samym złożoność problematyki jakości i wielość elementów składających się na nią (Hiam 1999):

- ◆ działanie, czyli cechy podstawowe produktu lub usługi;
- ◆ cechy uzupełniające;
- ◆ solidność wykonania, niezawodność działania;
- ◆ zgodność z normami;
- ◆ trwałość, wyrażana długością życia produktu;
- ◆ łatwość i szybkość obsługi;
- ◆ estetyka, a więc wygląd, smak czy zapach;
- ◆ postrzeganie jakości przez klienta.

Niezależnie od tego, którą z wyżej wspomnianych definicji jakości obierzemy dla naszych celów, należy mieć świadomość, że jakość trzeba rozpatrywać z punktu widzenia użytkownika czy odbiorcy produktu, co oznacza, że należy znać kontekst i przeznaczenie danego produktu. W innym przypadku nie ma możliwości oceny jakości, nie ma też możliwości zapewnienia tejże jakości w produkcji.

Normalizacja

Jakość to nie tylko pewna ocena produktu przez jego odbiorcę. To również sposób komunikacji pomiędzy dostawcą produktu a klientem, gdzie to klient decyduje o „wartości” dobra. Zauważyły to organizacje normalizujące, akcentując fakt, że stopień spełnienia oczekiwań klienta stanowi rzeczywistą miarę oceny skuteczności działania przedsiębiorstwa dostawcy dóbr i jego pozycji rynkowej. Organizacje te dostrzegły również, że na jakość składa się coś więcej niż tylko wyprodukowanie „dobrego” towaru — na satysfakcję klienta przekłada się też zagwarantowanie dostaw danego produktu na określonym, stabilnym poziomie jakości, po konkurencyjnej cenie i w ustalonych terminach. Aby to osiągnąć, potrzebne są odpowiednie procesy, jako że tylko dobrze zdefiniowane procesy pozwalają zapewnić powtarzalność produkcji przy jednoczesnej eliminacji podstawowych błędów.

Początków normalizacji w obszarze jakości należy szukać w latach 50. ubiegłego wieku. Wtedy to firmy funkcjonujące w krajach wolnego rynku zauważyły konieczność zapobiegania błędom, nie tylko ich wykrywania. Za „ojca” norm w dziedzinie jakości można uznać Stany Zjednoczone, gdzie w 1959 r. wydano normę MIL-Q-9858, opisującą system zapewnienia jakości. Norma ta została ochrzczone mianem Wymagania programu jakości i jako pierwsza definiowała wymagania stawiane dostawcom oferującym swe produkty armii USA. Kilka lat później, w 1963 r., zaktualizowaną normę opublikowano jako MIL-Q-9858A i stała się ona podstawą do opracowania regulacji dotyczących dostawców dla NATO wydanych w roku 1969 jako AQAP-1.

Interesujące w powyższych normach jest to, że nałożyły one na dostawców i poddostawców określone wymagania jakościowe, dotyczące wszystkich etapów powstawania produktu, nie tylko wyniku końcowego. Od tej pory producenci chcący dostarczać swoje wyroby armii USA musieli zapewnić spełnienie określonych wymagań jakościowych od fazy przedprodukcyjnej do poprodukcyjnej. Ponieważ obowiązek spełnienia wymagań jakościowych dotyczył też podwykonawców, grono zainteresowanych normami jakościowymi poszerzyło się na inne obszary przemysłu.

W kolejnym dziesięcioleciu (lata 70.) normalizacja objęła przemysł maszynowy i energię jądrową. Powstały też pierwsze normy narodowe, np. BS 5750 w Wielkiej Brytanii, która jest uznawana za pierwszy światowy standard zarządzania systemem jakości. W 1987 r. standard ten stał się normą ISO 9000, której poszczególne elementy stanowią podstawę budowania systemów zarządzania jakością (SZJ) w wielu organizacjach. Normy te zawierają terminologie, wymagania i wytyczne dotyczące wprowadzania, doskonalenia i kontrolowania systemu zarządzania jakością.

Pierwsza nowelizacja norm serii ISO 9000 miała miejsce w roku 1994 i koncentrowała się na ujęciu pełnego cyklu życia wyrobu — od momentu zdefiniowania potrzeb klienta po użytkowanie produktu. Druga nowelizacja, przeprowadzona w roku 2000, jest o tyle interesująca, że polegała na dokonaniu gruntownych zmian mających na celu przybliżenie koncepcji SZJ do TQM¹ oraz m.in. uwzględnienie doświadczeń zebranych

¹ *Total Quality Management* — zarządzanie przez jakość. Koncepcja ta zostanie omówiona w następnym rozdziale.

podczas stosowania wcześniejszych wersji normy i zaakcentowanie znaczenia podejścia procesowego oraz zaangażowania najwyższego kierownictwa jako czynników o kluczowym znaczeniu dla powodzenia SZJ.

Obecnie podstawą do budowania systemów zarządzania jakością są dwie normy:

- ◆ ISO 9001: 2008 — System zarządzania jakością — Wymagania;
- ◆ ISO 9004: 2009 — Zarządzanie mające na celu osiągnięcie trwałego sukcesu organizacji — Podejście poprzez zarządzanie jakością.

Seria ISO 9000 opiera się na dwóch podstawowych założeniach, które należy mieć na uwadze jako fundamentalne zasady jakiegokolwiek systemu zarządzania jakością:

- ◆ Lepiej zapobiegać niż leczyć — przeciwdziałanie występowaniu wad i awarii jest skuteczniejsze (i w ogólnym rozrachunku tańsze) niż ich wykrywanie i usuwanie. Z pomocą przychodzą tu różne narzędzia, wśród których prym wiodzie FMEA (ang. *Failure Model and Effects Analysis* — analiza przyczyn i skutków awarii), którą można postrzegać jako swoiste rozwinięcie diagramu Ishikawy. Wymogiem normy ISO 9001 jest ciągle doskonalenie, a FMEA dostarcza ku temu rozwiązania.
- ◆ Jakość wymaga systemowego podejścia — organizacja rozumiana jako system zarządzania przedsiębiorstwem jest obszarem o największym potencjale oddziaływania na poziom jakości. Innymi słowy: jakość zaczyna się na najwyższych szczeblach zarządzania organizacją.

Warto zwrócić uwagę na to, że normy ISO serii 9000 nie są normami technicznymi — nie definiują parametrów, jakie powinny spełniać wyroby procesu produkcyjnego, jedynie opisują pewne zasady, których przestrzeganie może zapewnić odpowiednią jakość. Do tematu norm wróć w rozdziale dotyczącym kosztów jakości.

Znaczenie jakości w projektach informatycznych

*We współczesnym świecie cena zależy od jakości, a nie od kosztów,
które nie interesują nabywcy*

— Wacław Wilczyński

Wszystkim czytelnikom zapewne doskonale znany jest żartobliwy obrazek przedstawiający poszczególne etapy projektowe: od pozyskania wymagań, poprzez ich analizę i zaprojektowanie docelowego rozwiązania po dostarczenie gotowego produktu do klienta. Etapy te i ich produkty wizualizowane są za pomocą kolejnych projektów tworzonego wyrobu, który w końcowej fazie znacznie odbiega (niedopowiedzenie stulecia...) od wyobrażeń i potrzeb klienta. Grafiki te są zabawne i poprawiają humor, warto jednak poważniej pochylić się nad znaczeniem, które ze sobą noszą.

Czy rzeczywiście tak często w wyniku realizacji projektów powstaje coś zgoła odmiennego od oczekiwań klienta? Czy naprawdę tak często klient płaci za coś, czego nie chciał, lub, co gorsza, czego nie potrzebuje do realizacji swoich celów biznesowych, ponieważ nie spełnia to takiej funkcji? Odpowiedź nie napawa optymizmem. Niestety — tak. Z punktu widzenia biznesowego projekty bardzo często — o wiele za często — kończą się mniejszym lub większym niepowodzeniem. W sytuacjach ekstremalnych w ogóle się nie kończą, gdyż sfrustrowany klient, widząc mizerne efekty prac dostawcy, rezygnuje z kontynuacji prac projektowych. W mniej ekstremalnych i niestety bardzo częstych sytuacjach rezygnacja z ukończenia projektu nie jest możliwa z różnych powodów: od względów finansowych (zainwestowano przecież wiele środków w zaplanowanie i realizację prac projektowych) po formalne (kwestie umów i wymagań formalnych), a z powodu różnego rodzaju błędów procesowych klient otrzymuje niepełny czy/i nieodpowiedni do swoich potrzeb produkt. Przykłady można podawać niemalże w nieskończoność: od drobnych usterek powodujących rozczarowanie klienta i stwierdzenie: „Myślałem, że to będzie inaczej działało/wyglądało” po wady w praktyce uniemożliwiające wdrożenie produktu i jego sprawne użytkowanie.

Skąd takie problemy? I dlaczego są one tak często (wręcz notorycznie) spotykane w branży IT?

Powodów jest wiele. Należy zacząć od przyczyny fundamentalnej, powtarzającej się w przerażającym odsetku przedsięwzięć: nieprawidłowe określenie przedmiotu zamówienia. Czyli w prostych słowach: zamawiający albo nie wie dokładnie, czego potrzebuje, albo nie jest w stanie tego precyzyjnie określić. Można by rzec — to nie problem! Od tego jest dostawca rozwiązania, by doradzić, podpowiedzieć, zasugerować... Nic bardziej mylnego. Nikt, poza klientem, nie jest w stanie określić tego, czego klient potrzebuje do realizacji swojej działalności biznesowej. To klient określa cele i wyniki realizacji projektu, ponieważ to klient będzie korzystał z wytworzonego rozwiązania. To klient musi zdefiniować, co chce osiągnąć za pomocą systemu informatycznego czy innego produktu powstałego w wyniku realizacji prac projektowych. Problem leży w tym, że bardzo często po stronie klienta brakuje etapu przedprojektowego, który to etap powinien poddać analizie funkcjonowanie organizacji, ustalić cele biznesowe i procesy realizujące te cele, określić obszary niezbędne do doskonalenia. Dopiero na tej podstawie wyłaniają się cele, które należy zrealizować, aby usprawnić działanie danej organizacji, co w następnej kolejności powoduje przygotowanie propozycji projektów mających dostarczyć środków do realizacji owych celów. Omawiany etap nosi nazwę analizy przedsiębiorstwa (IIBA 2005) i niestety w wielu (zbyt wielu!) przypadkach jest zupełnie pomijany. Zamiast tego klient radośnie tworzy kilka „wymagań” dla konkretnego systemu informatycznego, licząc na to, że ten właśnie system w cudowny sposób uzdrowi funkcjonowanie firmy... To zupełnie jak podawanie losowego leku choremu, u którego nie wykonano żadnych badań i diagnostyki, z nadzieją że ten właśnie lek zadziała. Trudno się dziwić, jeśli będzie inaczej, prawda? A jednak w przypadku projektów IT za każdym razem klient nie może wyjść ze zdumienia, że produkt projektu, którego koszty są horrendalne, nie tylko nie pomaga — czasami nawet jest gwoździem do trumny.

Innym typowym problemem w projektach IT jest brak właściwych procesów — nie tylko tych bezpośrednio związanych z jakością, ale i innych, kluczowych dla wytworzenia samego produktu. Zastanówmy się: jakie są realia wielu projektów? Na co najczęściej skarżą się członkowie projektów? Na podstawie obserwacji branży i rozmów z wieloma osobami, od „szeregowych” pracowników po menedżerów, można wysnuć

następujący wniosek — czas. Brakuje czasu na niemalże wszystko — na kompletne zebranie wymagań, na ich analizę, na projekt rozwiązania, na implementację, na testy... Znowu — skąd ten problem? Przecież po to jest planowanie projektu, by ustalić realistyczny czas trwania poszczególnych czynności, prawda? Teoretycznie — tak. W praktyce bywa różnie. W praktyce bardzo często trudno jest oszacować potrzebny nakład pracy na podstawie niezbyt dokładnych wymagań klienta, a ponieważ nie ma czasu (*sic!*) bądź możliwości (lub, co gorsza, chęci) na doprecyzowanie wymagań i ustalenie, czego właściwie klient potrzebuje i co dany projekt ma dostarczyć, dostawca zwykle wycenia dany projekt, zakładając pewne rzeczy. Założenia te bardzo często dotyczą wymagań i zakresu rozwiązania — dostawcy wydaje się, że do zrobienia będzie mniej niż to się okazuje w rzeczywistości. Taka metoda szacowania rzadko się sprawdza — śmiem twierdzić, że niewiele zespołów projektowych ma na stanie kompetentną wróżkę umiejącą przewidywać przyszłość i czytać w myślach. Jakie są tego efekty? Już podczas pierwszych etapów projektowych wychodzi na jaw, że w zakładanych parametrach czasowych lub finansowych po prostu nie ma możliwości wykonania pewnych prac. Skutek — aby dotrzymać niemożliwych do spełnienia terminów, dostawca „optymalizuje” plany, dokonując cięć. Zwykle cięcia dotyczą analizy wymagań i testów. Implementacji raczej nie da się skrócić, ponieważ w jakiś sposób należy dostarczyć klientowi produkt (co nie znaczy, że można powiedzieć, że nie są podejmowane próby „przyspieszenia” deweloperów... O efekty tych prób warto zapytać testerów). Ale wymagania i testy to idealni kandydaci do skrócenia czasu trwania projektu. Piszę to stwierdzenie z przekąsem i lekką frustracją, ponieważ mimo tylu przykładów popełniania tego błędu i jego efektów są ludzie, którzy nadal wierzą, że na podstawie wymagań klienta i własnego wyczucia tematu można wyprodukować rozwiązanie zgodne z oczekiwaniami odbiorcy. Niezwykle zaskakujące jest to, że wiele osób nie widzi potrzeby dokładnej analizy wymagań i opracowania starannego projektu rozwiązania, po czym nie może wyjść ze zdziwienia, że klient nie jest zachwycony przedstawionym wytworem prac projektowych. Czasem nie trzeba czekać aż do demonstracji produktu klientowi, ponieważ braki zostają ujawnione już w testach (oczywiście zakładając, że i testów nie „ucięto”), co skutkuje koniecznością rozległych, pracochłonnych i kosztownych przeróbek. Co dziwne, wielu dostawców nie widzi problemu w konieczności ciągłych poprawek i wielokrotnie powtarzanych testów mających na celu sprawdzenie poprawności wprowadzania tych poprawek, zupełnie jakby prace te były wykonywane w ramach wolontariatu i nic nie kosztowały. A przecież znaczny odsetek błędów wynika z nieprecyzyjnych, sprzecznych czy niejednoznacznych wymagań. Logicznym wnioskiem byłoby raczej zwiększenie nakładu prac związanych z analizą wymagań i projektowaniem rozwiązania w celu uniknięcia problemów na późniejszych etapach projektu. Nie od dziś wiadomo, że defekty znajdowane później kosztują więcej (tabela 2.2). Ale nie — z uporem godnym lepszej sprawy wiele organizacji IT popełnia ciągle ten sam błąd, licząc na uzyskanie innych wyników.

Tabela 2.2. Względny koszt naprawy defektu w różnych fazach wytwarzania (McConnell 2004)

Faza wprowadzenia defektu	Faza wykrycia defektu				
	Wymagania	Architektura	Implementacja	Testy systemowe	Wdrożenie
Wymagania	1	3	5 – 10	10	10 – 100
Architektura	–	1	10	15	25 – 100
Implementacja	–	–	1	10	10 – 25

Istnieje jeszcze wiele innych czynników przekładających się na zwiększone ryzyko niepowodzenia projektu IT.

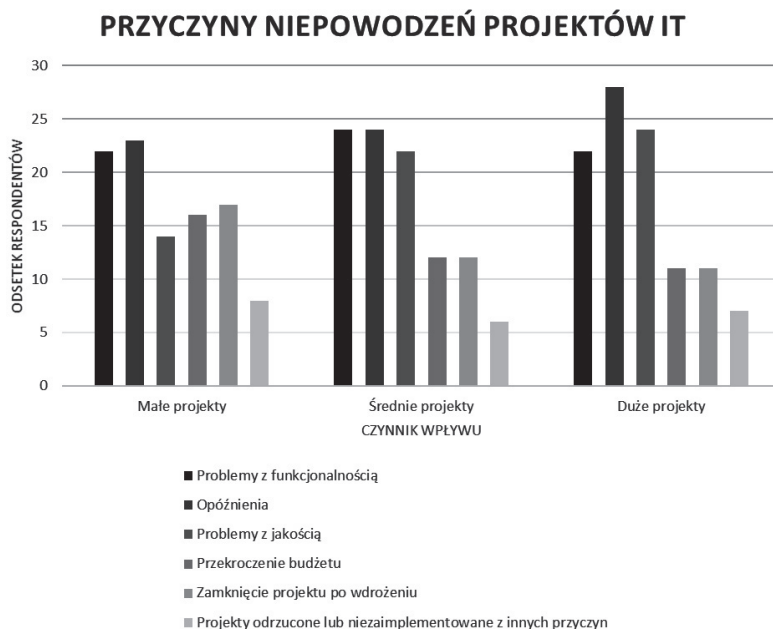
Gartner (2014), organizacja zajmująca się m.in. badaniami rynku, przeprowadziła analizę przyczyn niepowodzeń projektów. Zapytano respondentów o określenie powodów, z których projekty realizowane w firmach respondentów nie odniosły sukcesu. W odpowiedziach powtarzało się sześć przyczyn:

- ♦ problemy z funkcjonalnością,
- ♦ opóźnienia,
- ♦ problemy z jakością,
- ♦ przekroczenie budżetu,
- ♦ zamknięcie projektu po wdrożeniu,
- ♦ projekty odrzucone lub niezaimplementowane z innych przyczyn.

Procentowy udział powyższych czynników w ogólnej porażce projektu przedstawia wykres (rysunek 2.1).

Rysunek 2.1.

Przyczyny niepowodzeń projektów IT (Gartner 2012)



Warto zauważyć, że wskazane przez firmę Gartner przyczyny niepowodzeń są ogólnymi czynnikami podanymi przez respondentów — nie pokazują one rzeczywistych źródeł problemów, a jedynie ich skutki, efekty końcowe zaobserwowane przez uczestników badań. Trudno pokusić się o wyeliminowanie czynników ryzyka, nie znając realnych przyczyn problemu — fakt ten potwierdzałyby statystyki od lat wskazujące niemal dokładnie takie same czynniki wpływające na porażki projektowe. Nic w tym dziwnego

— skoro nie wiadomo, co jest prawdziwym źródłem problemu, nie dojdziemy do rozwiązania eliminującego problem. Popelniamy ciągle te same błędy skutkujące tymi samymi efektami na przestrzeni lat. A przecież, cytując Alberta Einsteina, „szaleństwem jest robić wciąż to samo i oczekiwać różnych rezultatów”.

Czy jest szansa na poprawę tej sytuacji i wyciągnięcie wniosków, które umożliwiłyby zmniejszenie ryzyka porażki projektu? Oczywiście. Z pomocą przychodzą tu narzędzia zarządzania jakością, takie jak analiza przyczyn podstawowych czy diagramy Ishikawy, pozwalające na identyfikację rzeczywistego źródła problemu. Próbę zdiagnozowania sytuacji i syntetycznego podsumowania źródeł problemów w projektach IT podjęto w publikacji *Co robimy źle i jak tego uniknąć?* (Zmitrowicz, Chrabski 2014a). Autorzy oparli swoje wnioski na wynikach badań statystycznych prowadzonych przez międzynarodowe organizacje, w tym Gartner (2012) i Standish Group (1995), wyróżniając następujące wyzwania związane z przedsięwzięciami informatycznymi:

- ◆ cele i wizja,
- ◆ niewłaściwe planowanie projektu,
- ◆ słaba komunikacja,
- ◆ niewłaściwe zarządzanie oczekiwaniami interesariuszy,
- ◆ problemy z wymaganiami i ich zakresem,
- ◆ brak umiejętności miękkich,
- ◆ nierealistyczne oczekiwania,
- ◆ brak zasobów ludzkich,
- ◆ brak odpowiedniego wsparcia narzędziowego i metodycznego.

Co najmniej kilka z opisywanych we wspomnianej publikacji czynników bezpośrednio przekłada się na niską skuteczność procesów wytwórczych, a co za tym idzie, na zwiększone ryzyko dostarczenia nieodpowiedniego produktu (lub niedostarczenia produktu w ogóle).

Wszystko to jest bardzo ciekawe, lecz dociekliwy czytelnik zapyta: „Ale co to ma wspólnego z jakością?”. A czym jest jakość, jeśli nie spełnieniem wymagań odbiorcy? Jeśli nie znamy dokładnych wymagań co do zamawianego produktu i stale, procesowo, nie monitorujemy poziomu zgodności produktu z tymi wymaganiami, to jak mamy zapewnić jakość? Wbrew obiegowym opiniom jakość to nie tylko testowanie — wręcz przeciwnie, testy znajdują się na końcu procesu prowadzącego do osiągnięcia wymaganego stanu jakości.

Spójrzmy na temat z innej strony. Jak zapewniana jest jakość samochodów czy zegarków z wysokiej półki, cieszących się niesłabnącym zainteresowaniem i zaufaniem nabywców? W te produkty jakość się wbudowuje od samego początku: od szczegółowych wymagań, poprzez precyzyjny, zwalidowany projekt aż po dokładnie określony, powtarzalny proces produkcji. W tym przypadku o „cięciu” na etapie projektowania nie ma mowy, gdyż producenci znakomicie zdają sobie sprawę z wagi projektu dla jakości całego produktu i z konsekwencji wszelkich zaniedbań. „Ale to co innego” — mógłby oburzyć się

czytelnik znający realia projektów IT, w których trudno mówić o produkcji seryjnej czy stabilnych wymaganiach. Jednak czy to naprawdę aż taka różnica? Budujemy produkt na konkretne zamówienie konkretnego klienta. Dostarczane na początku projektu wymagania dotyczące oprogramowania bywają nieprecyzyjne i mogą się w pewnym stopniu zmienić. A więc tym bardziej, drogi Czytelniku, powinieneś dbać o jakość i ustalić takie zapewniające ją procesy, by maksymalnie zmniejszyć ryzyko dostarczenia produktu niezgodnego z wymaganiami klienta. Środków do tego nie brakuje — dostarczają ich procesy zarządzania jakością — brakuje najczęściej świadomości i chęci.

Koszty jakości

Jakość jest za darmo

— Philip Crosby

Częstą obawą przed wdrożeniem systemu zarządzania jakością, czy też podjęciem jakichkolwiek innych zorganizowanych działań zmierzających ku zapewnieniu jakości produktów, są koszty. W wielu organizacjach nadal pokutuje przekonanie, że jakość kosztuje, a w związku z tym, żeby dostarczyć produkty o odpowiedniej jakości, klient musi odpowiednio zapłacić. Czyli do standardowych kosztów produkcji należałoby doliczyć dodatek na zapewnienie jakości, co znacznie (w rozumieniu producenta) zwiększy całkowity koszt projektu. Ponieważ zaś w dobie olbrzymiej konkurencji klient może przebierać wśród ofert wielu dostawców, przedsiębiorcy wychodzą z założenia, że należy utrzymać atrakcyjną cenę oferowanego towaru czy usługi, aby to ich ofertę wybrano i by to właśnie oni realizowali projekt. Myślenie w taki sposób powoduje, że jakość spada na dalszy (czasem bardzo daleki) plan, a priorytetem staje się koszt i czas realizacji przedsięwzięcia.

Dostawcy rozwiązań tłumaczą wysokie koszty zapewnienia jakości wydatkami ponoszonymi na szczegółową analizę, przygotowanie dokładnej specyfikacji produktu, dalej kontrolę jakości tej specyfikacji, wzmożone testowanie (zwykle na kilku poziomach i w co najmniej kilku cyklach) i wszelkie „dodatkowe” czynności, które muszą przedsięwziąć, aby uzyskać pożądaną przez klienta poziom jakości. Przedstawiają szczegółowe wyceny, by udowodnić, że osiągnięcie pewnego poziomu jakości musi kosztować tyle i tyle. Rzecz jasna wielu klientów przeraża sama myśl o ponoszeniu dodatkowych kosztów, nie mówiąc już o tym, że co najmniej osobliwe wydaje im się żądanie zapłaty za jakość, czyli oczywistą cechę zamawianego produktu (nabywając np. krzesło, oczekujemy, że będzie nam służyło przez krótszy czy dłuższy czas, zapewniając pewien komfort odpoczynku; nikt raczej nie spodziewa się otrzymania produktu, który okaże się niezdatny do użytkowania, niewygodny czy nietrwały). Należy uwzględnić fakt, że stosunkowo spory odsetek klientów branży IT nie zdaje sobie sprawy z tego, że o ile dla nich jakość zamawianego produktu jest oczywistością, o tyle dla dostawców parametry jakościowe to cechy produktu, które podlegają wycenieniu, i jeśli nie zostaną uwzględnione w szacowaniu kosztów projektu, nie zostaną po prostu zaimplementowane. Tym sposobem klient oczekuje taniego, dobrego produktu w ustalonym terminie, dostawca zaś balansuje na krawędzi katastrofy, starając się zmieścić w ograniczonym budżecie i czasie, by dostarczyć klientowi jakiś produkt, przy czym *jakiś* jest tu słowem kluczowym.

To przekonanie prowadzi do kuriozalnych sytuacji, w których realizuje się projekty, mając na uwadze jedynie wymagany zakres funkcjonalny produktu, budżet przewidziany na realizację przedsięwzięcia i czas dostawy produktu do klienta. Zapomina się w tym wyścigu o jednej istotnej kwestii — jakości. Nie jest sztuką oddanie klientowi produktu, który nie spełni oczekiwań. W takich przypadkach obie strony są przegrane: klient, ponieważ dostał nie to, czego *potrzebował*, i dostawca, bo wytworzył produkt, z którego niskiej jakości zdaje sobie doskonale sprawę, i nie napawa go dumą efekt własnych prac.

Sytuacja, można by rzec, patowa. Jak tego typu problemów uniknąć? Po pierwsze i chyba najważniejsze, należy budować świadomość jakościową. Po obu stronach: zamawiającego i producenta. Jeśli dla stron jakość produktu ma znaczenie, trzeba się liczyć z pewnymi konsekwencjami. Konsekwencje te funkcjonują już na zasadzie prawd w świecie IT (nie tylko IT zresztą, jako że dotyczą dowolnych usług), znanych pod nazwą prawo dwóch trzecich:

Szybko i dobrze — nie będzie tanio.

Tanio i szybko — nie będzie dobrze.

Tanio i dobrze — nie będzie szybko.

Prawo dwóch trzecich urosło do rangi żartu branżowego, doczekało się nawet własnego demotywatora (rysunek 2.2).

Rysunek 2.2.
Prawo dwóch trzecich
(SplaszFX 2011)



Mimo zastosowanego tu lekkiego podejścia do tematu trzeba jednak przyjąć do wiadomości i zaakceptować fakt, że prawo dwóch trzecich nie jest żartem informatyków — w taki sposób naprawdę działa wszelki biznes. Nie należy oczekiwać wysokiej jakości, jeśli nie chce się zainwestować odpowiednich nakładów. Niestety, klienci często sami

sobie szkodzą, wybierając najtańszą ofertę (słynne kryterium ceny, widoczne swego czasu zwłaszcza w przetargach publicznych) i dziwiąc się później, że „to nie jest to, czego chcieliśmy”. Szkodzą też sobie dostawcy, nie wyjaśniając jasno, że nie da się połączyć niskiej ceny z wysoką jakością. „Chytry dwa razy traci” — mówi polskie przysłowie, trafnie podsumowując zasady obowiązujące w świecie IT. Niska jakość produktu oznacza, że klient otrzymał produkt niespełniający jego oczekiwań. Konsekwencje tego mogą zasadniczo przybrać następujące formy:

- ♦ Klient składa reklamację i żąda naprawy lub wymiany produktu na taki, który będzie zgodny z oczekiwaniami. Dodatkowo klient może żądać pokrycia strat poniesionych w wyniku użytkowania wadliwego produktu i zapłaty kar umownych przewidzianych w ramach umowy z producentem.
- ♦ Klient zraża się do firmy producenta — może zaakceptować poniesione na wadliwy produkt koszty, jednak opinia, jaką głosi o produkcie i samym dostawcy, mogłaby skutecznie zniechęcić aktualnych i przyszłych klientów. W tym przypadku producent teoretycznie nie ponosi bezpośrednich kosztów, jednak w ogólnym rozrachunku koszty biznesowe oraz koszty utraty klientów i możliwości biznesowych są olbrzymie, mogą doprowadzić nawet do upadku firmy.

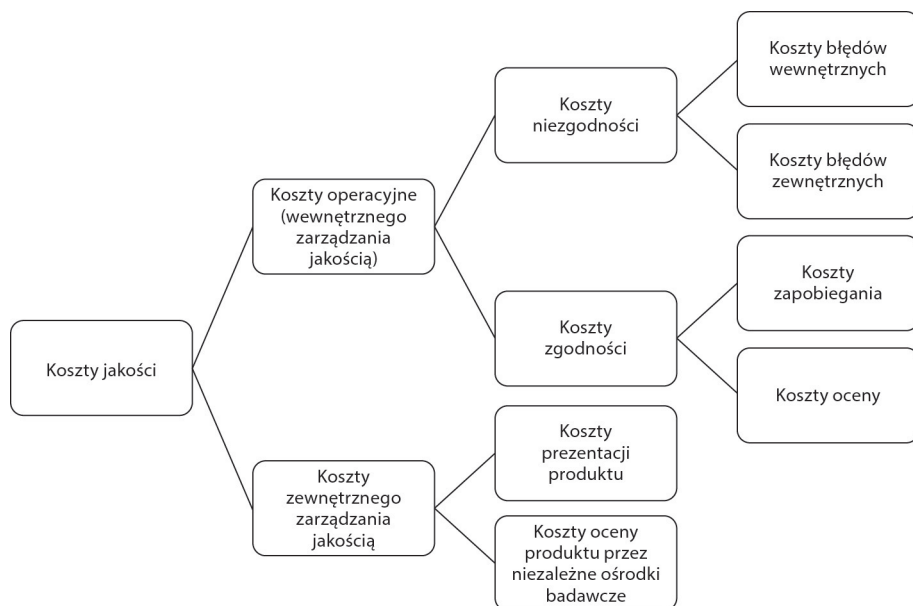
Opisane wyżej konsekwencje to skutki braku jakości. Nie musi do nich dojść, jeśli zaplanujemy i wdrożymy środki mające na celu zapewnienie jakości, ponosząc przy tym rzecz jasna pewien koszt, jednak z pewnością niższy niż koszty braku jakości.

Tym sposobem dochodzimy do tematu kosztów jakości. Jakie nakłady trzeba ponieść, by zapewnić jakość produktów?

Nakłady te znane są w zarządzaniu jakością pod nazwą koszty jakości. Dotyczą one wszelkich wydatków (czy raczej inwestycji) poniesionych na uzyskanie pewności, że produkty oddawane klientowi wykonane są zgodnie z oczekiwaniami i potrzebami. Według Johna Banka (1996) „Koszt jakości to niejako skrócony wzór wszystkich kosztów poniesionych przy wytwarzaniu wysokiej jakości produktu lub usługi. Składają się na nie koszty profilaktyki, koszty szacowania, koszty wewnętrznych błędów, koszty związane z przekroczeniem wymagań klientów, a także koszty wynikające z utraconych korzyści. W sumie koszty te mogą pochłonąć 20 – 30 proc. przychodów lub obrotów firmy”.

Koszty jakości są więc sumą wszystkich kosztów operacyjnych związanych z osiągnięciem jakości. Stanowią one element ogólnych kosztów wytwarzania produktu i jako takie powinny być ujęte w całkowitym budżecie realizacji przedsięwzięcia. Ponadto koszty jakości umożliwiają ilościową ocenę skuteczności funkcjonowania systemu zarządzania jakością stosowanego w danym przedsiębiorstwie, są zatem pomocnym wskaźnikiem w optymalizacji procesów.

Koszty jakości można sklasyfikować na kilka sposobów. Norma ISO 9004: 1994 rekomenduje podział kosztów jakości na koszty związane z operacjami wewnętrznymi oraz z działalnością zewnętrzną przedsiębiorstwa (rysunek 2.3). Koszty wewnętrzne to koszty zgodności, czyli nakłady na zapobieganie usterkom i ocenę jakości produktu, oraz koszty niezgodności, czyli te związane z błędami. Koszty zewnętrzne wiążą się



Rysunek 2.3. Koszty jakości według ISO 9004: 1994

z reprezentacją i dowodami wymaganymi jako obiektywne wykazanie jakości. Zalicza się do nich (Łańcucki 1994) koszty oceny zgodności systemu jakości przez instytucje certyfikacyjne (np. koszty uzyskania certyfikacji ISO 9001) oraz koszty badań i oceny właściwości produktu przez niezależne ośrodki badawcze i audytujące.

Klasyfikacja ISO nie przewiduje jednak pewnego istotnego czynnika wpływającego na to, że koszty wynikające z braku jakości są w rzeczywistości jeszcze większe, jeśli uwzględnimy koszty niewymierne. Mówi o tym TQM, wyróżniając nie tylko koszty wewnętrzne i zewnętrzne, ale i koszty widoczne i niewidoczne (tabela 2.3).

Tabela 2.3. Klasyfikacja kosztów w TQM (Dahlgaard i in. 2004)

Wyszczególnienie	Koszty wewnętrzne	Koszty zewnętrzne	Łącznie
<i>Koszty widoczne</i>	1a. Koszty braków i napraw 1b. Koszty zapobiegania i ocen	2. Koszty gwarancji i koszty reklamacji	1 + 2
<i>Koszty niewidoczne</i>	3a. Utrata wydajności wskutek niskiej jakości i niewłaściwego zarządzania 3b. Koszty zapobiegania i ocen	4. Utrata reputacji wskutek niskiej jakości i niewłaściwego zarządzania	3 + 4 (?)
<i>Łącznie</i>	1 + 3 (?)		1 + 2 + 3 + 4 (?)

Widoczne w tabeli znaki zapytania wskazują, że poza znanymi kosztami, które możemy obliczyć (1 + 2), istnieją jeszcze inne czynniki powodujące, że koszt łączny pozostaje nieznany.

Skorowidz

5 x dlaczego, 80

A

analiza

- kosztów jakości, 45
- opłacalności, 44
- Pareto, 80
- postępu i wyników, 210
- procesów, 45
- przyczyn niepowodzeń projektów, 19
- raportów incydentów, 210
- wartości brzegowych, 188

anomalie, 148, 151, 209, 233

cykl życia, 155

APQC, 38

artefakt, 156

aspekty

- jakości oprogramowania, 94
- praktyczne, 94

ASQ, American Society for Quality, 79, 95

atrakcyjność, Attractiveness, 110

atrybuty

- awarii, 154
- defektu, 150, 151
- funkcjonalności, 108
- niezawodności, 109
- procesu, 256
- przenaszalności, 111
- wydajności, 110

audyt, 45, 102, 133

audyt wewnętrzny, 48

awaria, 149

B

bezpieczeństwo, Security, 109, 112

błąd, 149

budowa

oprogramowania, 93

procesu, 36

C

cele

biznesowe, 266

jakości, 53

testowania, 183, 266

charakterystyki jakościowe

dla procesu, 106

dla produktu, 106

ciągłe doskonalenie, 50

CMMI, 243, 248, 256

obszary procesów, 252

standardowa praktyka, 250

standardowy cel, 250

CTP, Ctritical Testing Processes, 257, 271–274

cykl życia

anomali, 155

defektu, 158

czynniki wpływu, 105

D

defekt, 18, 66, 147–150, 155

definicja jakości, 11, 13

Deming William Edwards, 56

diagram

Ishikawy, 20, 30, 45, 79

przypadków użycia, 193

dojrzałość, Maturity, 109

dokumentacja

ISO 9001, 51

raportów z testów, 233

specyfikacji testów, 226

testów, 169, 217, 245

- dokumentacja
 wykonania testów, 232
 zarządcza, 220
- doskonalenie
 jakości, 247
 procesów organizacyjnych, 248
 procesu testowego, 256
- działania
 korygujące, 47
 zapobiegawcze, 47
- dzienniki testów, 217
- E**
- elementy
 procesu, 62
 systemu zarządzania, 89
- F**
- faza wykrycia defektu, 18
- filozofia
 kaizen, 58
 lean, 275
- FMEA, 16, 79, 226, 247
- FTR, Formal Technical Review, 97
- funkcjonalność, Functionality, 108
- funkcjonowanie kół jakości, 78
- G**
- gęstość błędów, 215
- grupy procesów, 44
- H**
- harmonogram, 103
- I**
- IDEAL, 257, 258
- identyfikacja
 procesów, 36, 37
 ryzyka, 178
- IEEE
 1008, 237
 1012, 162
 1028, 124–129, 162, 243
 1044, 150, 154, 162, 243
 1061, 162, 245
 12207, 82, 118
 1233, 161
 1362, 161
- 2014, 94
 29148, 161
 610, 160, 167, 243
 730, 162
 828, 102, 160, 243
 829, 169, 218, 237
 830, 102, 161
- incydenty, 235
- inspekcja, inspection, 129, 132
- interoperacyjność, Interoperability, 109
- Ishikawa Kaoru, 75
- ISO
 19011, 91
 31000, 160
 9000, 36, 79, 95
 9001, 16, 51, 90
 9004, 16, 24, 90
 9241, 159
- ISO/IEC
 15504, 254
 15504-2, 92
 25000, 159, 245
 9126, 108, 111
 9126-1, 107–109
- ISO/IEC/IEEE
 29119, 204, 240
 29119-3, 245
- ISO/TR 10017, 91
- ISO/TS 16949, 91
- ISO-19011, 91
- ISTQB, 164, 204, 275
- ISTQB/SJSI 2012, 175
- J**
- jakość, 11, 80, 93
 cele, 53
 definicje
 strategiczne, 14
 wielowymiarowe, 13, 14
 związane z produkcją, 13
 związane z produktem, 13
 związane z tworzeniem wartości, 13
 związane z użytkownikiem, 13
- doskonalenie, 247
- klasyfikacja kosztów, 24
- koło, 75
- koszty, 21
- księga, 53
- metody poprawy, 55
- normy, 16
- optymalny poziom, 31
- osiem zasad, 49

plan zapewnienia, 99
 polityka, 52
 procesy zarządzania, 41
 redukcja kosztów, 26
 rozważania Garvina, 14
 trójkąt, 114
 zarządzanie kosztami, 32
 JIT, just-in-time, 80

K

kaizen, 58
 kategorie
 procesów, 38, 255
 przyczyn problemu, 30
 klasyfikacja
 anomalii, 149, 155, 162
 definicji jakości, 13
 kategorii anomalii, 153
 kosztów, 24
 krytyczności anomalii, 151
 procesów, 38
 technik testowych, 185
 usterek, 215
 kodeks postępowania
 interes publiczny, 86
 ja, 87
 kierownictwo, 86
 klient i pracodawca, 86
 osąd, 86
 produkt, 85
 współpracownicy, 86
 zawód, 86
 koła jakości, 75, 78
 koncepcja
 jakości, 11
 lean, 275
 zarządzania jakością, 63
 kontekst testowania, 173
 kontrola
 jakości, 10, 41, 46, 67, 148
 testów, 211
 koszty
 jakości, 21, 23
 jakości organizacji, 25
 naprawy defektu, 18
 niewidoczne, 24
 oceny, 27
 wadliwości, 27
 wewnętrzne, 24
 widoczne, 24
 zapobiegania, 27
 zewnętrzne, 24, 27
 księga jakości, 53

L, Ł

lista kontrolna, 45, 137, 139
 LSD, Lean Software Development, 275
 łatwość
 adaptacji, Adaptability, 111
 analizy, Analysability, 110
 instalacji, Installability, 111
 konserwacji, Maintainability, 108, 110, 112
 nauki, Learnability, 110
 testowania, Testability, 110
 wprowadzania zmian, Changeability, 110
 zastąpienia, Replaceability, 111
 zrozumienia, Understandability, 110

M

macierz
 oceny dojrzałości, 271
 RACI, 225
 RASCI, 225
 śledzenia wymagań, 208
 V&V, 103
 manifest jakości, 87
 maszyna stanów, 190
 metody
 inżynierii oprogramowania, 93
 poprawy jakości, 55
 metryki, 143, 214
 dla jakości w użyciu, 107
 jakości, 45
 wewnętrzne, 107
 zewnętrzne, 107
 miary produktu, 215
 MMAST, 274
 model
 Boehma, 118
 CMMI, 250
 CTP, 274
 jakości McCalla, 115, 116
 jakości oprogramowania, 109
 kosztów procesów, 26
 MMAST, 274
 PAF, 26
 PMBOK, 41
 procesów testowych, 240
 procesu, 61
 SPICE, 256
 strat społecznych jakości, 26
 TIM, 275
 TOM, 275
 TMMi, 264
 TPI® Next, 269
 TQM, 60
 TSM, 274

modele
 doskonalenia procesu testowego, 274
 jakości, 106
 jakości produktu, 107
 wzrostu niezawodności, 215
 monitorowanie testów, 211
 możliwość współistnienia, Co-existence, 111

N

narzędzia
 do zarządzania testami, 146
 do zarządzania wymaganiami, 146
 kontroli jakości, 45
 zarządzania jakością, 79
 niezależność testowania, 170
 niezawodność, Reliability, 108, 112
 normalizacja, 15
 normy procesowe, 243
 nota wydania produktu, 218
 notacja UML, 192

O

OAT, Operational Acceptance Testing, 183
 ocena
 kryteriów zakończenia testów, 210
 porównawcza, 44
 odporność na błędy, Fault-tolerance, 109
 operatywność, Operability, 110
 opis
 procedury testowej, 232
 przypadków testowych, 230
 warunków testowych, 227
 oprogramowanie
 budowa, 93
 ekonomia inżynierii, 94
 jakość, 93
 modele, 93
 pielęgnacja, 93
 projektowanie, 93
 testowanie, 93, 94
 wymagania, 93
 zapewnienie jakości, 96
 zarządzanie inżynierią, 93
 zarządzanie konfiguracją, 93
 optymalny poziom jakości, 31
 organizacja testów, 170
 orientacja na klienta, 49

P

PDCA, 32, 49, 76, 247, 257
 osoby, 202

perspektywy produktu, 114
 pielęgnacja oprogramowania, 93
 plan
 doskonalenia procesów, 45
 jakości oprogramowania, 44, 82
 testów, 132, 207, 217, 245
 V&V, 121
 zapewnienia jakości, 99
 zarządzania jakością, 45
 podejmowanie decyzji, 50
 podejście
 procesowe, 50
 systemowe do zarządzania, 50
 podstawowy proces testowy, 170, 203
 podstawy
 inżynierii, 94
 matematyki, 94
 obliczeń, 94
 polityka
 jakości, 52
 testów, 217, 267
 pomiary pokrycia/staranności, 216
 pomyłka, 149
 posiew usterek, 216
 poziom dojrzałości
 mierzony, 263
 optymalizacja, 263
 organizacji, 93
 wstępny, 259
 zarządzany, 261
 zdefiniowany, 262
 poziomy
 niezależności testowania, 172
 testów, 179
 praktyki inżynierii oprogramowania, 93
 prawo dwóch trzecich, 22
 procedury testowe, 217
 proces, 35
 inżynierii oprogramowania, 93, 106
 oceny dojrzałości, 269
 planowania jakości, 44
 przeglądu formalnego, 135
 realizacji przeglądu, 135
 testowy, 170, 174, 203, 256, 261
 zarządzania jakością, 41, 45
 procesy
 dynamiczne, 205
 operacyjne, 38
 organizacyjne, 248
 pomocnicze, 37
 wspomagające, 38
 zarządcze, 205
 profile operacyjne, 202
 program Bugzilla, 158

projektowanie
 i implementacja testów, 207
 oprogramowania, 93
 przypadków testowych, 169
 przedziały równoważności, 187
 przegląd, 45, 102, 124
 formalny, 135
 kierowniczy, 127
 techniczny, 126
 zarządzania, 47
 koleżeński, 103
 na zakończenie fazy, 103
 przejrzanie, 124
 przenaszalność, Portability, 108, 112
 przepływ
 kontroli dla testów decyzji, 200
 zadań, Workflow, 155
 przyczyny
 niepowodzeń projektów, 19
 występowania przerw, 29
 przypadek
 testowy, 167, 217
 użycia, 140, 193, 194
 przywództwo, 49

Q

QA, Quality Assurance, 95
 QFD, Quality Function Deployment, 72

R

R&D, Research & Development, 72
 RAM, responsibility assignment matrix, 225
 raport
 anomalii, 217
 incydentu, 234
 podsumowujący testy, 217, 233
 realizacja przeglądu, 135
 redukcja kosztów jakości, 26
 reengineering, 57
 relacje z dostawcami, 50
 Risk Management, 160
 rola, 101, 102
 rozważania Garvina, 14
 rozwój środowiska testowego, 209
 ryzyko, 178
 ryzyko niepowodzenia projektu, 19

S, Ś

Six Sigma, 56, 80
 specyfikacja wymagań, 139

SPICE, 254
 atrybuty procesu, 256
 poziomy dojrzałości, 255
 SQA, 101, 102
 SQM, Software Quality Management, 81
 SQuaRE, 111, 159
 stabilność, Stability, 110
 standardy, 88
 statystyki usterek, 215
 strategie testowe, 174–177, 217, 268
 struktura
 dokumentacji testów, 218
 modelu TMMi, 264
 procedury testowej, 231
 przypadku testowego, 228, 229
 raportu incydentu, 234
 raportu podsumowującego, 234
 SWEBOK, 80, 93, 118, 163–166, 179, 216
 system zarządzania, 89
 jakością, 48, 62, 79
 śledzenie wymagań, 208

T

tablica
 decyzyjna, 189
 przejść stanów, 192
 techniki
 analityczne, 83
 dynamiczne, 83
 kontroli jakości, 45
 oparte na
 charakterze systemu, 202
 intuicji i doświadczeniu, 186
 kodzie, 195
 ludziach, 83
 przepływie danych, 201
 przepływie kontroli, 196
 specyfikacji, 187
 usterekach, 201
 użyciu, 202
 statyczne, 83
 testowe, 184
 walidacji i weryfikacji, 120
 techniki zarządzania jakością, 79, 83
 testalia, 167
 testowanie, 83, 163
 ad hoc, 186
 decyzji, 198
 eksploracyjne, 187
 instalacji, 184
 instrukcji, 196
 konfiguracji, 184

testowanie
 oparte na ryzyku, 177
 oparte na specyfikacji formalnej, 192
 oprogramowania, 93
 użyteczności, 184
 wydajności, 184
 zgodności, 184

TestSPICE, 275

testy
 akceptacyjne, 181, 184
 alfa, 183
 beta, 183
 integracji, 180
 jednostkowe, 179
 mutacyjne, 202
 obciążeniowe, 184
 odtwarzalności, 184
 regresji, 184
 systemowe, 181

Tickit, 91
 Tickit plus, 91, 254
 TMM, 258, 275
 TMMi, 55, 97, 177, 220, 256–268, 275
 struktura modelu, 264
 zastosowania, 265

TOM, 275
 Tom Gilb, 87
 TPI Next, 257, 269–271
 TPI® Next, 269
 kluczowy obszar, 270
 macierz oceny dojrzałości, 271

TQC, 56
 TQM, Total Quality Management, 15, 55–80, 248
 trójkąt jakości, 114
 TSM, 274

typy
 strategii testów, 176
 testów, 184
 usterek, 215

U, V

UAT, 181
 UML, 192
 usterka, 149, 215
 użycie listy kontrolnej, 139
 użyteczność, Usability, 108–111
 V&V, verification and validation, 118, 120, 123

W

walidacja, 103, 118
 walidacja wymagań, 103
 wdrażanie TQM, 57

weryfikacja, 103, 118
 właściwości
 audytu, 134
 inspekcji, 130, 131
 przeglądu kierowniczego, 128, 129
 przeglądu technicznego, 127
 przejrzenia, 125, 126

wskaźnik
 inspekcji, 132
 mutacji, 217

wsparcie narzędziowe, 235
 wydajność, Efficiency, 108, 111
 wykonywanie
 przeglądu, 102
 testów, 209

wykorzystanie
 czasu, Time Behaviour, 110
 zasobów, Resource Behaviour, 110

wykres
 liczby błędów, 147, 148
 Pareto, 29

wykrycie defektu, 18
 wyrocznia testowa, 169

Z

zaangażowanie ludzi, 49
 zadania SQA, 101, 102
 zamknięcie testów, 213
 zapewnianie jakości, 45, 96, 99
 zarządzanie
 anomaliami, 155
 incydentami, 209
 inżynierią oprogramowania, 93
 jakością oprogramowania, 35, 40, 80–82, 94
 konfiguracją oprogramowania, 84, 93
 kosztami jakości, 32
 procesowe, 35
 projektem, 84
 przez jakość, 56, 58
 ryzykiem, 84
 testami, 146
 wydaniem, 84

zasady
 Deminga, 63–75
 jakości, 49
 LSD, 276

zastosowanie przeglądu, 120
 zdolność do odtworzenia, Recoverability, 109
 zgadywanie błędów, 201
 znaczenie jakości, 16

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA

 **Helion SA**

Zapewnianie wysokiej jakości oprogramowania to niełatwe zadanie. Osiągniesz ją, jeśli będziesz przestrzegać wysokich standardów procesu wytwarzania i dopilnujesz, by każdy problem został rozwiązany do końca. Jednak zaskakująco wielu producentów nie traktuje poważnie sygnałów o błędach. Ujawniają się one dopiero podczas testowania, czyli na etapie, gdy niewiele można zrobić. Ta książka podpowie Ci, jak na pierwszych etapach tworzenia kodu wykrywać i rozwiązywać pojawiające się problemy. Popraw efektywność swojej pracy już dziś!

Karolina Zmitrowicz zebrala najistotniejsze koncepcje z dziedziny zarzadzania jakością oprogramowania i uzupełniła je o własne doświadczenia. Znajdziesz tu omówienie podstaw testowania oraz pomoc w organizacji i planowaniu pracy. Nauczysz się tworzyć jakość, a nie tylko ją sprawdzać. Poznasz przydatne metody weryfikacji i walidacji, podstawy tworzenia dokumentacji wyników i narzędzia Lean Software Development. Dzięki wskazówkom zawartym w książce udoskonalisz swoje produkty, zoptymalizujesz proces ich wytwarzania i powiększysz grono zachwyconych klientów.

- Definicja jakości i jej znaczenie w projektach informatycznych
- Zarządzanie procesowe, jakością i przez jakość
- Zasady Deminga i koła jakości
- Zarządzanie jakością oprogramowania
- Manifest jakości i standardy
- Planowanie procesu zapewnienia jakości
- Weryfikacja i walidacja, metryki
- Anomalie – charakterystyka i sposób obsługi
- Podstawy testowania, organizacja i techniki testów
- Metryki związane z testowaniem i dokumentacja testów
- Wsparcie narzędziowe i standardy w testowaniu
- Doskonalenie procesów organizacyjnych i procesu testowego
- Lean Software Development (LSD)

**Dbaj o jakość – pamiętaj,
że stać Cię na więcej!**



31317 numer katalogowy

księgarnia internetowa



<http://helion.pl>

zamówienia telefoniczne



0 801 339900



0 601 339900

Sprawdź najnowsze promocje:

• <http://helion.pl/promocje>

Książki najchętniej czytane:

• <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

• <http://helion.pl/nowosci>

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

sięgnij po **WIĘCEJ**



KOD KORZYŚCI

ISBN 978-83-283-0156-6



9 788328 301566

Informatyka w najlepszym wydaniu

cena: 59,00 zł