

Wiesława Gryncewicz

Uniwersytet Ekonomiczny we Wrocławiu

ROLA UŻYTKOWNIKA W PROCESIE ZARZĄDZANIA JAKOŚCIĄ OPROGRAMOWANIA

Streszczenie: Problematyka jakości oprogramowania jest w ostatnim czasie jedną z najważniejszych we wszystkich zastosowaniach informatyki. W artykule przedstawiono genezę oraz ewolucję nauki o jakości i ukazano na tym tle zagadnienia związane z zarządzaniem jakością oprogramowania. Szczególny nacisk został położony na wykazanie roli użytkownika końcowego w tym procesie. Zgodnie z tradycyjnym podejściem testowanie i nadzorowanie jakości oprogramowania umieszczone było na krańcowych etapach cyklu produkcji. Wprowadzenie zarządzania jakością wymaga kontroli jakości oprogramowania w całym procesie produkcyjnym. Umieszczenie użytkownika w centrum tego procesu, powiązanie jego wymagań i celów biznesowych z testami pozwoli firmom informatycznym na lepsze dopasowanie swoich produktów do potrzeb i oczekiwań klientów.

Słowa kluczowe: jakość, jakość oprogramowania, zarządzanie jakością, inżynieria wymagań.

1. Wstęp

Problematyka jakości oprogramowania jest w ostatnim czasie jedną z najważniejszych we wszystkich zastosowaniach informatyki. Do niedawna większość publikacji z tego zakresu była poświęcona kwestii jakości procesu wytwarzania oprogramowania, o wiele mniej miejsca zajmowała ocena jakości wytworów tego procesu. Proporcje zaczynają się wyrównywać ze względu na rosnące wymagania użytkowników systemów. Niniejszy artykuł stanowi skromny przyczynek w tej sprawie.

Mówiąc o jakości, rozważa się szereg własności, które produkt powinien posiadać. Katalog własności zależy od tego, kto go definiuje. Innych cech oczekuje użytkownik, inne są istotne dla programisty, a jeszcze inaczej postrzega jakość klient, który płaci za produkt. Pierwszy z nich koncentruje się na własnościach dostarczonego oprogramowania. Dla programisty istotna jest testowalność i czytelność kodu projektowanego systemu. Klient – zleceniodawca jest zainteresowany najbardziej spełnieniem potrzeb biznesowych, a zwłaszcza terminem i kosztami realizacji.

W artykule problematyka jakości oprogramowania będzie analizowana z perspektywy użytkownika końcowego. Główny akcent zostanie położony na te elementy zarządzania jakością oprogramowania, które są związane z inżynierią wymagań.

Pominięte zostaną wszelkie wątki związane z technologiczną stroną produkcji oprogramowania, takie jak organizacja testowania i poprawiania błędów.

Celem artykułu jest przedstawienie genezy oraz ewolucji nauki o jakości i ukazanie na tym tle problematyki jakości oprogramowania. W dalszej części dokonana zostanie próba uwypuklenia roli użytkownika końcowego w procesie zarządzania jakością oprogramowania.

2. Geneza i przedmiot nauki o jakości

Termin „jakość” ma w znaczeniu naukowym i potocznym wiele interpretacji. Wynika to z jego „rozmytego” charakteru, silnie nacechowanego intuicyjnym, subiektywnym postrzeganiem przez człowieka.

Kategoria ta pojawiła się pierwszy raz u Platona (427-347 p.n.e.), twórcy idealizmu obiektywnego, pod greckim terminem *poiotes* [Hamrol, Mantura 2008, s. 20]. W jego systemie filozoficznym świat rzeczywisty stanowił niedoskonałe odzwierciedlenie realnie istniejących doskonałych idei. W tym kontekście jakość konkretnych rzeczy oznaczała stopień osiągniętej przez nie doskonałości. Warto zauważyć, że takie rozumienie jakości jest zbieżne z dominującym współcześnie, także w dziedzinie oprogramowania, definiowaniem jakości jako stopnia spełnienia przez przedmioty stawianych im wymagań.

Odmiennego zdania był Arystoteles (384-322 p.n.e.), twórca kierunku materialistycznego. Zaliczał on jakość do zbioru dziesięciu najogólniejszych kategorii opisu rzeczywistości i rozumiał przez nią zespół swoistych cech odróżniających dany przedmiot od innych przedmiotów tego samego rodzaju [*Nowa encyklopedia...* 2011]. A zatem można uznać, że był on zwolennikiem stworzenia powszechnie akceptowalnego, obiektywnego katalogu cech opisujących daną rzecz. Takie zbiorcze określenie cech Arystoteles nazywał jakością. Od czasów starożytnych toczono dyskusje dotyczące charakteru jakości – pojęcia absolutnego czy też poddającego się kategoryzacji?

Amerykański słownik Webstera definiuje jakość jako coś nieodzownego, immanentnego (*inherent*), a jednocześnie będącego jednoznaczny wyróżnikiem (*distinctive*) pojedynczego egzemplarza wyrobu spośród pewnej klasy [Merriam-Webster's... 2011]. W definicji tej zwrócono też uwagę na dwa różne aspekty pojmowania jakości:

- jakość to synonim doskonałości. To podejście neguje możliwość czy też konieczność wprowadzania miar jakości: jakość jest jedna i oznacza niedościgniony wzór, cel, do którego należy dążyć;
- jakość to stopień doskonałości. W tym określeniu kładzie się nacisk na stopniowanie jakości, jej mierzalność. Synonimy jakości w takim ujęciu to np. klasa, stopień, stan, pozycja, poziom.

Ta definicja jest obecnie powszechnie uznawana. Coraz częściej odchodzi się bowiem od uznawania jakości za niedefiniowalny, abstrakcyjny byt. Rozwijane są

techniki pomiaru jakości w sposób pośredni, tj. poprzez pomiar innych parametrów będących atrybutami jakości (np. dla jakości oprogramowania będą to takie cechy, jak efektywność, wydajność, niezawodność, bezpieczeństwo itp.).

Nieodzowność, nieusuwalność własności zwanej jakością podkreśla Polska Norma, będąca odpowiednikiem standardu ISO 9000. Według niej jakość to „stopień, w jakim zbiór inherentnych właściwości spełnia wymagania” [Polska Norma: PN-EN... 2006, s. 25].

3. Ewolucja nauki o jakości

Niezależnie od przyjętego sposobu definiowania geneza nauki o jakości wiąże się z procesami wytwórczymi. W miarę postępu i komplikacji procesów wytwarzania, produktów, procesów obrotu towarowego i eksploatacji oraz ciągłego wzrostu wymagań odbiorców kategorie jakościowe systematycznie umacniały swoją pozycję, rozwijało się także instrumentarium narzędziowe sterowania jakością w jednostkach gospodarczych. Można wyodrębnić cztery charakterystyczne stadia tej ewolucji [Dahlgaard i in. 2005, s. 170]:

- inspekcja jakości (ok. 1910 r.),
- kontrola jakości (ok. 1924 r.),
- zapewnienie jakości (ok. 1950 r.),
- kompleksowe zarządzanie jakością (ok. 1980 r.).

Celem inspekcji było oddzielenie wyrobów o niskiej jakości, wykrytych przez inspektorów, od wyrobów o zaakceptowanej jakości, a następnie ich wycofanie lub sprzedaż po niższej cenie.

Wraz z rozwojem przemysłu pojawił się drugi etap. Jakość poddano kontroli poprzez nadzór nad umiejętnościami pracowników, poprzez wymagania ustalone na piśmie, poprzez pomiary i standaryzację. Rozwinięto wówczas statystyczną kontrolę jakości, zwiększono kontrolę nad procesami; pojawiły się karty kontrolne i procedury odbioru.

Trzecie stadium, określane jako zapewnienie jakości, zawiera w sobie wszystkie poprzednie stadia. Dla uzyskania pewności, że wyroby i usługi zaspokajają potrzeby klientów, wprowadzono dodatkowo takie rozwiązania, jak: zwięzłe księgi jakości, analizy kosztów jakości czy audyty systemu. Wszystko to służyło odejściu od kontroli jakości na rzecz zapewnienia jakości, czyli odejściu od wykrywania niskiej jakości na rzecz działań zapobiegawczych.

Czwarty okres nazywany jest kompleksowym zarządzaniem jakością (*Total Quality Management* – TQM). Podejście to jest szeroko opisywane w wielu publikacjach zajmujących się problematyką zarządzania jakością. Charakteryzuje się ono kompleksowym podejściem do spraw jakości w organizacji i w jej otoczeniu oraz stosowaniem nowoczesnych technik zarządzania jakością. Wynika to z założenia, że jakość produktów finalnych jest determinowana działalnością wszystkich podsystemów organizacji i sytuacją w jej otoczeniu. Zadaniem TQM jest ciągle polepszanie

funkcjonowania firmy i zwiększanie zadowolenia klientów poprzez stałe doskonalenie realizowanych procesów, poprzez zaangażowanie w sprawy jakości wszystkich pracowników, dostawców i kooperantów, poprzez myślenie systemowe oraz poprzez zainteresowanie się nie tylko wyrobami i usługami, ale również jakością organizacji pracy, warunkami pracy i poziomem wiedzy pracowników.

Podejście subiektywne jest podstawą filozofii TQM, a także zarządzania jakością oprogramowania. Ocena jakości produktu nie jest zdeterminowana przez sposób zastosowania, ale poprzez postrzegane przez użytkownika korzyści wynikające z jego posiadania. Jakość jest w tym wypadku „subiektywną opinią użytkownika, wypadkową obiektywnie postrzeganych cech produktu i indywidualnych potrzeb i preferencji, która powstaje w drodze specyficznych procesów oceny i postrzegania” [Gardini 1997, s. 5]. Zgodnie z tą definicją jakość jest kategorią dynamiczną, ponieważ potrzeby i preferencje odbiorców zmieniają się w czasie.

Trudność w zarządzaniu jakością polega nie tylko na określeniu oczekiwań klienta, ale też na zaprojektowaniu produktu/usługi odpowiadających tym oczekiwaniom i na stałym monitorowaniu poziomu zadowolenia odbiorców/użytkowników.

4. Jakość oprogramowania

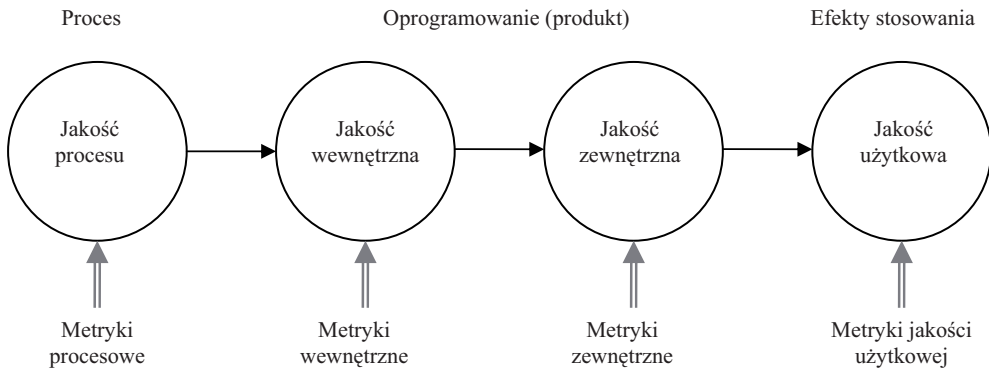
W dziedzinie oprogramowania jakość również nie jest w sposób jednoznaczny zdefiniowana. Istnieje wiele norm i standardów z tego zakresu. W niniejszym artykule zostanie wykorzystana norma ISO 9126 [ISO/IEC 9126:2001...], ponieważ uwypukla ona rolę użytkownika. Norma ta wprowadza ogólny model jakości oprogramowania, który obejmuje cały cykl życia: od procesu wytwórczego, poprzez testowanie akceptacyjne, aż po okres eksploatacji.

Ze względu na to, że uczestnicy wszystkich wymienionych etapów inaczej postrzegają tę problematykę, w normie wprowadzono podział na trzy zasadnicze perspektywy [Sacha 2010, s. 306]:

- perspektywę wewnętrzną (*internal quality*), która obejmuje cechy dające się ocenić na podstawie produktów pośrednich powstających w procesie wytwórczym, odnoszące się głównie do kodu programowego, specyfikacji i projektu,
- perspektywę zewnętrzną (*external quality*), która obejmuje cechy oprogramowania postrzegane podczas oceny gotowego produktu,
- perspektywę użytkową (*quality in use*), która obejmuje cechy określające stopień spełnienia potrzeb użytkownika.

Wszystkie perspektywy powiązane są zależnością, zgodnie z którą jakość użytkowa wynika z jakości zewnętrznej, która powinna wynikać z jakości wewnętrznej, a ta – z jakości procesu wytwórczego (rys. 1).

Jakość oprogramowania nie jest pojęciem statycznym, zmienia się w trakcie prac nad produktem i zależy od tego, kto ją ocenia i na jakim etapie. Jakość oprogramowania może być mierzona za pomocą miar wewnętrznych (zwykle są to statyczne miary kodu źródłowego programu) lub miar zewnętrznych, oceniających zachowa-



Rys. 1. Model jakości oprogramowania według normy ISO 9126

Źródło: [Sacha 2010].

nie się programu w czasie jego wykonywania. Do oceny produktu z punktu widzenia użytkownika służą miary jakości użytkowej.

Modele jakości zewnętrznej i wewnętrznej są takie same i obejmują zbiór cech, które charakteryzują jakość oprogramowania. Zakres oceny każdej z tych cech można opisać ilościowo za pomocą metryk. Mimo identycznego zestawu cech metryki są na ogół różne, gdyż są przeznaczone do stosowania na innych etapach procesu wytwórczego i do innego rodzaju produktów (por. [Sacha 2010, s. 307]). Model ten składa się z następujących charakterystyk:

- funkcjonalność (*functionality*) – zdolność dostarczania funkcji spełniających wyspecyfikowane i niewyspecyfikowane potrzeby użytkownika; te ostatnie mogą wynikać z regulacji prawnych lub zwyczajów panujących w dziedzinie zastosowania,
- niezawodność (*reliability*) – zdolność stałego utrzymywania możliwości działania podczas eksploatacji w założonych warunkach,
- łatwość użycia (*usability*) – łatwość zrozumienia przeznaczenia produktu i określenia jego przydatności do potrzeb użytkownika; łatwość opanowania obsługi i użytkowania oraz atrakcyjność interfejsu,
- wydajność (*efficiency*) – zdolność oprogramowania do wykonywania odpowiedniej ilości pracy w stosunku do ilości zużytych zasobów,
- łatwość utrzymania (*maintainability*) – łatwość poprawiania błędów i wprowadzania zmian podczas ewolucji programu oraz przewidywalność efektów zmiany i testowalność,
- przenośność (*portability*) – zdolność przenoszenia do innego środowiska pracy, zarówno sprzętowego, jak i programowego lub organizacyjnego.

Pomiar jakości użytkowej polega na ocenie rezultatów stosowania oprogramowania, a nie jego charakterystyk [Kobyliński 2003]. Norma ISO 9126 wymienia cztery cechy składające się na model jakości użytkowej:

- efektywność (*effectiveness*) – w zakresie osiągania celów biznesowych użytkownika,
- wydajność (*productivity*) – określona przez stosunek korzyści do nakładu zasobów (zasoby te mogą obejmować czas pracy, wysiłek, materiały lub nakłady finansowe),
- bezpieczeństwo (*safety*) – wyrażające możliwość osiągnięcia akceptowalnego poziomu ryzyka uszkodzenia ludzi, biznesu lub środowiska (wynikają one zwykle z niezadowolającej funkcjonalności, niezawodności, łatwości użycia lub modyfikowania oprogramowania),
- satysfakcja (*satisfaction*) – mierzona stopniem zadowolenia użytkownika w zakresie zastosowania produktu.

Wysoka ocena jakości użytkowej zależy od spełnienia wymagań nałożonych przez miary zewnętrzne, na które mają wpływ wartości miar wewnętrznych (por. rys. 1). Jednak nie jest to warunek wystarczający, ponieważ ocena jakości użytkowej jest oceną subiektywną [Kobyliński 2003].

Użytkownika końcowego nie interesuje jakość wewnętrzna ani sposób tworzenia oprogramowania. Postrzega on jakość poprzez pryzmat bezpośrednich własności dostarczonego oprogramowania i świadczeń towarzyszących wynikających z umowy (pomoc techniczna, szkolenie itp.). Ocenia on produkt od momentu, gdy ten pracuje już w środowisku docelowym. Dla użytkownika ważne jest to, czy ów produkt umożliwia osiągnięcie konkretnych celów efektywnie, wydajnie, bezpiecznie i satysfakcjonująco.

Nie oznacza to jednak, że jego rola ogranicza się tylko do tej oceny. Aby ta końcowa ocena okazała się wysoka i zadowolająca, użytkownik musi być włączony w cały proces zarządzania jakością oprogramowania, o czym będzie mowa w kolejnym punkcie.

5. Zarządzanie jakością oprogramowania – udział użytkownika

Zarządzanie jakością oprogramowania wykształciło się jako odpowiedź na potrzebę sprawniejszego zarządzania projektami informatycznymi. Zaczęto wówczas stosować zasady inżynierii jakości wywodzące się z przemysłu, o których była mowa w poprzednich punktach artykułu. Również zarządzanie jakością oprogramowania ewoluowało podobnie jak w wspomnianych dziedzinach – od tzw. systemów kontroli jakości, poprzez systemy zapewnienia jakości, do zarządzania nią.

Początkowo osiągnięcie jakości w projektach informatycznych utożsamiano ze zbudowaniem systemu zgodnego z przyjętymi uprzednio wymaganiami. W systemach kontroli jakości rozwinęło się testowanie funkcjonalne, czyli sprawdzanie, czy aplikacja zachowuje się w oczekiwany, z góry założony sposób. Testowanie to było wykonywane pod koniec cyklu wytwarzania oprogramowania. Jakość była oceniana głównie przez producenta czy też ekspertów – specjalistów z dziedziny informatyki, a wady traktowano jedynie jako odstępstwa od specyfikacji.

Jak już wspomniano wcześniej, cechą jakości jest jej dynamika i subiektywizm, a zatem i zmienność. Podejście do tej problematyki ewoluowało z czasem w kierunku systemów zapewnienia jakości. W tym okresie nastąpiła rozbudowa aparatu normalizującego i standaryzującego, a audyty jakości dotyczyły procesów biznesowych, w ramach których tworzone jest oprogramowanie.

Obecnie jakość rozumie się jako stopień, w jakim system czy produkt informatyczny spełnia, a nawet przekracza potrzeby i oczekiwania klienta. Systemy zarządzania jakością oprogramowania oparte są na analizie celów biznesowych, analizie wymagań, badaniach konkurencji i badaniach marketingowych oraz na ocenie zadowolenia klientów.

Opisana powyżej ewolucja ilustruje zmiany w podejściu do problematyki jakości oprogramowania. Nie wszystkie firmy informatyczne podążają jednak „z jej duchem” i tkwią niejako w poprzedniej epoce. Tabela 1 zawiera zestawienie cech charakteryzujących tradycyjne oraz nowoczesne pojmowanie omawianego zagadnienia.

Tabela 1. Tradycyjne a nowoczesne podejście do jakości oprogramowania

Podejście tradycyjne	Podejście nowoczesne
<ul style="list-style-type: none"> • Jakość to <ul style="list-style-type: none"> – Zgodność produktu z wymaganiami • Jakość ocenia <ul style="list-style-type: none"> – Producent, ekspert • Co to jest wada? <ul style="list-style-type: none"> – Odstępstwo od specyfikacji • Kryteria jakości <ul style="list-style-type: none"> – Obiektywne i mierzalne – Dotyczą atrybutów, charakterystyk i parametrów technicznych • System zapewnienia jakości <ul style="list-style-type: none"> – Analiza wymagań, zdefiniowanie i kontrola procesu wytwarzania, standaryzacja, dokumentacja 	<ul style="list-style-type: none"> • Jakość to <ul style="list-style-type: none"> – Stopień spełnienia oczekiwań użytkownika (klienta) • Jakość ocenia <ul style="list-style-type: none"> – Klient, użytkownik • Co to jest wada? <ul style="list-style-type: none"> – Niespełnienie oczekiwań użytkownika • Kryteria jakości <ul style="list-style-type: none"> – Subiektywne oceny użytkownika – Stopień satysfakcji użytkownika • System zapewnienia jakości <ul style="list-style-type: none"> – Analiza celów biznesowych, analiza wymagań, badania konkurencji i marketingowe, ocena zadowolenia klientów

Źródło: [Dobrosielski 2010].

Podejście tradycyjne było nastawione na wykrywanie defektów, natomiast zarządzanie jakością motywuje do szerokiego oceniania jakości wszystkich produktów częściowych oraz całego procesu wytwórczego.

Zarządzanie jakością oprogramowania to zespół czynności obejmujących:

- planowanie jakości (*quality planning*) – definiowanie standardów jakościowych, które powinny być zachowywane w trakcie realizacji projektu informatycznego, oraz sposobów ich osiągnięcia,
- zapewnienie jakości (*quality assurance*) – dostosowanie wszystkich elementów procesu projektowego do stawianych wymagań,

- kontrolę jakości (*quality control*) – kontrolowanie i nadzorowanie działań projektowych w celu zapewnienia założonej jakości oraz eliminowanie czynników mogących obniżyć jej poziom.

Podczas wszystkich wymienionych wyżej etapów kluczowego znaczenia nabiera dążenie do takiego zrozumienia potrzeb klienta, aby można je było w pełni zaspokoić lub nawet zaoferować więcej. Wynik prowadzonych prac musi być zgodny ze specyfikacją i powinien zaspokajać rzeczywiste potrzeby, dla których projekt został podjęty.

Zarządzanie jakością oprogramowania to proces, który zasadniczo zmienia tradycyjne podejście do tworzenia oprogramowania, włączając już od samego początku zagadnienia jakościowe w proces dostarczania oprogramowania oraz umieszczając klienta w jego centrum. Pozwala to na połączenie priorytetów biznesowych i oczekiwań jakościowych z wymaganiami projektu, działaniami programistycznymi oraz testowaniem oprogramowania. Umożliwia to zespołom tworzącym oprogramowanie [Portal inwestycji... 2011]:

- wyeliminowanie źródła najczęstszych problemów w procesie tworzenia oprogramowania dzięki wykorzystaniu wysokiej jakości wymagań, które można testować,
- wydajne identyfikowanie problemów i wad podczas tworzenia oprogramowania, kiedy koszt ich usunięcia czy naprawy jest relatywnie niski,
- podejmowanie strategicznych decyzji w oparciu o sprawdzone dane i wyliczenia,
- nadzorowanie wszystkich testów manualnych oraz automatycznych w ramach zarządzanego procesu,
- wykorzystanie heterogenicznych i rozproszonych środowisk testowych w ramach jednej platformy ds. jakości,
- szybsze dostarczanie aplikacji o wyższej jakości i wydajności.

W żadnej z dziedzin zarządzanie jakością nie może być jakimś odrębnym, niezależnym procesem, który realizuje się w celu zmierzenia poziomu jakości osiągniętych wyników. Zarządzanie jakością musi być częścią procesu zarządzania organizacją. W przypadku firmy informatycznej zarządzanie jakością oprogramowania musi się zaczynać w momencie rozpoczęcia projektu i trwać nieprzerwanie aż do jego zamknięcia. Ponadto wszyscy członkowie organizacji powinni się czuć odpowiedzialni za jakość tworzonych programów i systemów.

Podsumowując niejako dotychczasowe rozważania, należy przyjąć, że obecnie użytkownik końcowy jest ostatecznym sędzią jakości oprogramowania. Musi on zatem być w centrum procesu zarządzania jakością, co powinno też mieć odzwierciedlenie w sposobie funkcjonowania firm informatycznych i w zaangażowaniu ich pracowników. Współcześnie całość działań w projektach informatycznych związana jest z pracą nad wymaganiami użytkownika, począwszy od początkowej fazy formułowania założeń, poprzez realizację systemu według określonych wymagań, a skończywszy na weryfikacji zgodności rozwiązań w gotowym systemie [Łada 2011]. W związku z koniecznością usystematyzowania działań w tym zakresie wykształciła

się dziedzina zwana inżynierią wymagań. Opisuje ona schemat pozyskiwania, dokumentacji, analizy i zarządzania wymaganiami [Sommerville 2006]. Cały proces inżynierii wymagań dotyczy wszystkich działań związanych z wymaganiami – od ich pozyskiwania we wstępnej fazie projektu, poprzez wszystkie zmiany, do ich finalnego ustalenia, co w przypadku projektów informatycznych często trwa przez cały cykl pracy nad projektem, aż do jego zakończenia. Dzięki tak zorganizowanej pracy wykonawcy systemu mają pewność, że wyspecyfikowane warunki będą satysfakcjonowały użytkownika oraz zaspokajały jego potrzeby. Przekłada się to na wytworzenie wysokiej jakości systemu, zgodnego z potrzebami klienta [Łada 2011].

Opracowanie wymagań narzuca konieczność współpracy każdej z trzech stron, o których była mowa we wstępie: klienta (zleceniodawcy) i użytkownika, którzy znają dziedzinę zastosowania i swoje potrzeby, oraz wykonawcy, który zna dostępne technologie i rozumie wynikające z nich ograniczenia. Główna odpowiedzialność za prawidłowe realizowanie zasad inżynierii wymagań spoczywa na wykonawcy. Klient odpowiada za dostarczenie informacji oraz weryfikację przedstawianej dokumentacji. Jednak to przyszły użytkownik musi sprecyzować to, czego oczekuje od systemu, i to zgodnie z tą specyfikacją system ów będzie tworzony w ramach ograniczeń i różnych warunków brzegowych ustalonych ze zleceniodawcą.

Zaangażowanie użytkownika w proces tworzenia oprogramowania jest warunkiem sukcesu całego przedsięwzięcia, sukcesu, który będzie mierzony jakością zgodną z jego oczekiwaniami oraz zadowoleniem pozostałych uczestników projektu.

6. Zakończenie

Produkty programistyczne są złożone i wieloaspektowe. Dodatkową trudnością w ocenie i zarządzaniu ich jakością jest konieczność uwzględnienia, oprócz mierzalnych atrybutów technicznych, czynników subiektywnych. Zgromadzenie informacji na temat obu tych grup potrzeb, zrozumienie punktu widzenia użytkownika oraz włączenie go w proces wytwórczy to główne czynniki sukcesu wiodących firm produkujących oprogramowanie.

Zgodnie z tradycyjnym podejściem testowanie i nadzorowanie jakości oprogramowania umieszczone było na krańcowych etapach cyklu produkcji. Wprowadzenie zarządzania jakością wymaga kontroli jakości oprogramowania w całym procesie produkcyjnym oraz wczesnego testowania i mierzenia błędów. Umieszczenie użytkownika w centrum tego procesu, powiązanie jego wymagań i celów biznesowych z testami pozwoli firmom informatycznym na lepsze dopasowanie swoich produktów do potrzeb i oczekiwań klientów.

Zwrócenie szczególnej uwagi na problematykę inżynierii wymagań w projektach informatycznych jest również istotne ze względu na koszty naprawy błędów w kolejnych cyklach życia oprogramowania. Jeśli pozyskane informacje okażą się niepełne lub niespójne, koszt likwidacji błędu, będącego tego następstwem, rośnie wykładniczo w kolejnych etapach projektu. Słyszcy się nawet o regule 1:10:100, która mówi, że niewielki koszt zmiany wymagania podczas projektowania systemu

rośnie dziesięciokrotnie w przypadku fazy kodowania lub testowania oprogramowania, natomiast jest stokrotnie wyższy, gdy zmianę należy wprowadzić w gotowym systemie podczas fazy użytkowania [Łada 2011].

Literatura

- Bobkowska A., *Należyta staranność w inżynierii wymagań i modelowaniu systemów*, [w:] *Należyta staranność w informatyce: Prawne aspekty jakości oprogramowania i systemów informatycznych*, Materiały konferencyjne, Warszawa 2008.
- Dahlgaard J., Kristensen K., Kanji G., *Fundamentals of Total Quality Management*, Routledge 2005.
- Dobrosielski R., *Zapewnienie jakości w projekcie informatycznym*, „CORE” 2010, nr 3.
- Gardini M.A., *Qualitätsmanagement in Dienstleistungsunternehmen – dargestellt am Beispiel der Hottellerie*, Lang, Frankfurt/Main 1997.
- Hamrol A., Mantura W., *Zarządzanie jakością. Teoria i praktyka*, Wydawnictwo Naukowe PWN, Warszawa 2008.
- Polska Norma: PN-EN ISO 9000:2006 System zarządzania jakością – Podstawy i terminologia, Polski Komitet Normalizacyjny, Warszawa 2006.
- ISO/IEC 9126:2001, Software engineering – Product Quality – Part 1: Quality model, ISO, Geneva 2001.
- Kobyliński A., *ISO/IEC 9126 – analiza modelu jakości produktów programowych*, [w:] T. Porębska-Miącz, H. Sroka (red.), *Systemy Wspomagania Organizacji 2003*, Katowice 2003.
- Leffingwell D., Widrig D., *Zarządzanie wymaganiami*, WNT, Warszawa 2003.
- Łada M., *Znaczenie inżynierii wymagań w projektach informatycznych*, http://www.4pm.pl/artukul/znaczenie_inzynierii_wymagan_w_projektach_informatycznych-45-1407.html, 10.01.2011.
- Merriam-Webster's Collegiate Dictionary, <http://www.webster.com>, 10.01.2011.
- Nowa encyklopedia powszechna PWN*, <http://encyklopedia.pwn.pl>, 10.01.2011.
- Portal inwestycji informatycznych*, <http://it-portal.pl>, 10.01.2011.
- Sacha K., *Inżynieria oprogramowania*, Wydawnictwo Naukowe PWN, Warszawa 2010.
- Sommerville I., *Software Engineering*, Addison Wesley Publishing Company 2006.

ROLE OF THE USER IN THE PROCESS OF SOFTWARE QUALITY MANAGEMENT

Summary: The issue of software quality is recently one of the most important problems in computer science applications. The article describes the origins and evolution of science of quality. Against this background, it shows the problems related to software quality management. Particular emphasis was placed on demonstrating the role of the end user in this process. According to the traditional approach, testing and monitoring the quality of software was located in the final stages of the production cycle. The introduction of quality management requires the control of software quality throughout the entire manufacturing process. Placing the user at the centre of this process, linking his requirements and business objectives with software tests will enable IT organizations better align their products to customer needs and expectations.

Key words: quality, software quality, quality management, engineering requirements.