jakość,
ocena jakości,
system informatyczny

MAZUR Zygmunt[1]
MAZUR Hanna[1]

OCENA JAKOŚCI SYSTEMU INFORMATYCZNEGO

Ocena jakości systemu informatycznego, pomimo, że w dużej mierze zależy od wielu atrybutów mierzalnych, jest również oceną subiektywną. Celem pracy jest podkreślenie znaczenia oceny jakości wewnętrznej wytwarzanego systemu informatycznego oraz konieczności przeprowadzania testów i weryfikowania zgodności wytwarzanych produktów z wymaganiami i z założonymi celami począwszy od analizy i specyfikacji wymagań.

ASSESSMENT OF INFORMATION SYSTEM QUALITY

Although it depends largely on many measurable attributes, the assessment of information system quality is also a subjective evaluation. The aim of this paper is to emphasise the importance of the assessment of internal quality of an information system and the necessity for testing and verification of the accordance of the manufactured products with the requirements and with the targets starting with the requirements analysis and specification.

1. WSTĘP

Zapotrzebowanie na oprogramowanie jest duże i stale rośnie. Według raportu IDC Poland w 2010 roku w Polsce na zakup oprogramowania przez administrację samorządową, edukację i urzędy centralne wydano ponad 1 mld zł [4]. Informatyzacja obejmuje coraz więcej urzędów. Opracowany *Plan Informatyzacji Państwa na lata 2011-15* przewiduje, że w 2015 roku będzie 5 mln użytkowników systemu informatycznego ePUAP (Elektroniczna Platforma Usług Administracji Publicznej), umożliwiającego komunikację i załatwianie spraw urzędowych za pośrednictwem Internetu. Polskie Towarzystwo Informatyczne w opinii opublikowanej 26 sierpnia 2011 roku, negatywnie oceniło jednak ten dokument, krytykując stosowany do tej pory mechanizm regulowania procesów informatyzacji w Polsce [8].

Czy oferowane oprogramowanie spełnia oczekiwania użytkowników? Czy wszystkie podejmowane przedsięwzięcia informatyczne zostają pomyślnie ukończone, a wytworzone produkty są dobrej jakości?

[1]Politechnika Wrocławska, Wydział Informatyki i Zarządzania, Instytut Informatyki; 50-370 Wrocław; Wybrzeże Wyspiańskiego 27. Tel. +48 71 320 42 23, E-mail: zygmunt.mazur@pwr.wroc.pl, hanna.mazur@pwr.wroc.pl

Od wielu lat trwa próba ustalenia przyczyn nieudanych przedsięwzięć oraz sformułowania wskazówek pomocnych w ich pomyślnej realizacji i zakończeniu. W tym celu proponowane są różne metodyki prowadzenia przedsięwzięć na przykład PMBoK, PMI, PRINCE2, RUP, SCRUM. Za udane uważa się przedsięwzięcie, które zostało ukończone w planowanym czasie, przy planowanym budżecie, jest zgodne z założeniami zamawiającego i spełnia oczekiwania użytkownika.

Pomimo wielu opracowań (norm, dobrych praktyk, metodyk, standardów i zaleceń) nadal, niestety, nie wszystkie przedsięwzięcia (w tym również, a może przede wszystkim, informatyczne) kończą się sukcesem.

Istotnym elementem w realizacji przedsięwzięć jest zapewnienie wysokiej jakości wytwarzanych produktów. W różnych dziedzinach ocena jakości produktu jest zadaniem bardziej lub mniej skomplikowanym. W przypadku systemów informatycznych zadanie to jest bardzo złożone. Ocena jakości wytwarzanego produktu (np. systemu informatycznego), pomimo, że w dużej mierze zależy od wielu atrybutów mierzalnych, jest również oceną subiektywną.

Celem pracy jest podkreślenie znaczenia oceny jakości wytwarzanych produktów (jakości wewnętrznej), konieczności przeprowadzania testów oraz weryfikowania zgodności z wymaganiami i założonymi celami począwszy od analizy i specyfikacji wymagań do testów akceptacyjnych użytkownika, wdrożenia i utrzymywania systemu.

2. DEFINICJE JAKOŚCI

Klienci oczekują od wykonawców dostarczania produktów dobrej jakości, ale nie zawsze precyzyjnie potrafią określić co to dla nich znaczy. Definicji jakości jest wiele. Według Platona (427-347 r. p.n.e.) jakość to "*pewien stopień doskonałości*". W normie PN-ISO 8402:1996 jakość definiuje się jako „*ogół cech i właściwości wyrobu lub usługi decydujący o zdolności wyrobu lub usługi do zaspokojenia stwierdzonych lub przewidywanych potrzeb użytkownika produktu*" a w normie PN-EN ISO 9000:2001 jakość to „*stopień, w jakim zbiór inherentnych właściwości spełnia wymagania*".

Według autorów trafną definicję jakości sformułował Leszek Wasilewski (jeden z pionierów zarządzania jakością w Polsce) a upowszechnił Andrzej Blikle [2, str. 25]: „*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*".

Aby podnosić jakość produktu należy określić cechy (charakterystyki, atrybuty) mierzalne podlegające ocenie, ich sposób oceny, zakres wartości i częstość mierzenia, a następnie systematycznie je kontrolować i reagować na wszelkie wyniki niezgodne z akceptowalnymi.

Podstawowe trzy zasady doktryny jakości to nieustanne doskonalenie (wszystkich i wszystkiego), racjonalność (myślenie systemowe czyli postrzeganie całościowe z uwzględnianiem wszystkich istotnych elementów i związków między nimi) oraz współpraca, poprzez budowanie odpowiednich relacji w zespole [2].

Jakość systemów bazodanowych zależy zarówno od jakości zaimplementowanej bazy danych jak i od jakości oprogramowania wykorzystywanego do jej obsługi. W celu zapewnienia wysokiej jakości wytwarzanego oprogramowania opracowano wiele norm, standardów i zaleceń. Spośród licznych dokumentów warto zapoznać się z normami:

- ISO/IEC 9126 – określającą standard opisu wymagań (charakterystyk jakościowych) dla oprogramowania,
- ISO/IEC 14598 – opisującą proces oceny jakości produktów programowych,
- ISO/IEC 15939 – dotyczącą oceny jakości oprogramowania,
- ISO/IEC 25000 (SQuaRE) – opisującą wymagania dotyczące produktów programowych oraz ich oceny.
- ISO/IEC 15504 (SPICE – *Software Process Improvement and Capability Determination*) – opisującą szczegółowo czynności, jakie należy wykonać przy ocenianiu, czy proces produkcji oprogramowania przebiega poprawnie.

W normie ISO/IEC 9126 przez jakość rozumie się wszystkie charakterystyki jednostki, które dostarczają jej zdolności do spełnienia założonych i przewidywanych potrzeb, a przez model jakości – zbiór charakterystyk i współzależności między nimi. W normie tej wyróżnia się model jakości wewnętrznej (postrzeganej na etapie wytwarzania produktu), zewnętrznej (postrzeganej po wytworzeniu produktu) i użytkowej. Jakość użytkowa to ocena systemu przez klienta, o której decyduje możliwość wykorzystania go do planowanych zadań [1,5].

3. JAKOŚĆ SYSTEMÓW INFORMATYCZNYCH

Na jakość wytwarzanych systemów informatycznych wpływa wiele czynników, między innymi jakość projektu i implementacji bazy danych (w przypadku systemów bazodanowych), jakość projektu i implementacji aplikacji użytkowych, wykorzystywane środowisko programowe – system operacyjny, System Zarządzania Bazą Danych, oprogramowanie użytkowe (edytor tekstu, arkusz kalkulacyjny, program pocztowy, komunikator internetowy itd.).

Ponadto jakość systemu zależy od wysokości przeznaczonego budżetu, od wiedzy i kompetencji wykonawców, znajomości metodyk, technologii i narzędzi, umiejętności szacowania ryzyka i organizacji pracy itd.

Obecne systemy są bardzo złożone (ang. *large scale software systems*) i muszą być wytwarzane przez wieloosobowe zespoły. Trudno jest dla nich od razu precyzyjnie zdefiniować wszystkie wymagania, tym bardziej, że w trakcie prac projektowych często ulegają one zmianom. Trudności przy wytwarzaniu dużych systemów wynikają również z braku ekspertów dziedzinowych i osób kompetentnych, z braku umiejętności pracy w zespole, z konieczności ciągłego uczenia się przez wykonawców (i przez klienta), z konieczności uwzględniania zachodzących zmian oraz z braku widocznych efektów przez długi czas, co powoduje zniechęcenie wykonawców, zamawiającego lub sponsora i zaniechanie prac lub ograniczenie zakresu przedsięwzięcia. Ponadto w czasie długo trwających przedsięwzięć pojawiają się sytuacje nieprzewidziane w harmonogramie (na przykład odejście kluczowych pracowników z zespołu projektowego, awarie sprzętu, wydarzenia losowe) wymuszające działania alternatywne i zmiany w harmonogramie.

Główne czynniki niepowodzeń przedsięwzięć informatycznych to duże rozmiary tych przedsięwzięć (bardzo często konieczna jest ich dekompozycja na mniejsze a następnie ich integracja), niewłaściwe harmonogramy (a zwłaszcza źle zaplanowane, niewystarczające testowanie), niewłaściwie planowane budżety, zła specyfikacja wymagań, źle zorganizowane zespoły projektowe i zła komunikacja między ludźmi.

Poprawę w realizacji podejmowanych przedsięwzięć można osiągnąć poprzez tworzenie odpowiednich zespołów, składających się z osób z wymaganymi umiejętnościami (kompetencjami), postęp nauki w zakresie zarządzania przedsięwzięciami, certyfikację procesu wytwarzania, stałą kontrolę celów, kontrolę jakości wewnętrznej i weryfikację zgodności wytwarzanych produktów pośrednich z wymaganiami.

W odróżnieniu od wielu innych produktów, dla których ocena ich jakości jest jednoznaczna (jednoznacznie mierzalna), ocena jakości systemu informatycznego jest trudna, niejednoznaczna, często bardzo kosztowna lub wręcz niemożliwa na etapie produkcyjnym oprogramowania ze względu na niemożność utworzenia rzeczywistego środowiska testowego i eksploatacyjnego. Ze względu na kluczową rolę systemów informatycznych w wielu zastosowaniach, konieczne jest testowanie i zapewnienie odpowiedniej jakości na każdym etapie realizacji projektu począwszy od specyfikacji wymagań aż do wdrożenia i eksploatacji systemu. Niestety, często planowanie przedsięwzięcia nie uwzględnia etapu eksploatacji systemu (kosztów, ludzi, szkoleń, koniecznych modyfikacji).

W wielu przypadkach do oceny jakości powołuje się audytorów zewnętrznych (spoza danej jednostki) co ma zapewnić niezależną, bardziej obiektywną ocenę. Dość często, niestety, w umowach zakupu oprogramowania spotykana jest klauzula *„as it is”* wyłączenia lub ograniczenia odpowiedzialności za wady oprogramowania.

Ocena systemu przez wykonawcę niejednokrotnie różni się od oceny dokonanej przez zamawiającego czy użytkownika. Według wykonawcy system jest poprawny, tymczasem klient jest niezadowolony i postrzega system jako błędny ponieważ system na przykład:

- wykonuje coś, czego nie powinien,
- nie wykonuje tego, co zgodnie ze specyfikacją powinien robić,
- nie wykonuje czegoś, co powinien, bo nie było tego w specyfikacji (a powinno być),
- wykonuje wszystko to co zostało wyspecyfikowane, ale w sposób niezadawalający (na przykład niewydajny) lub obsługa systemu jest trudna do opanowania i nie intuicyjna,
- nie jest skalowalny i elastyczny, uniemożliwia modyfikacje i dostosowywanie do zmian, jakie zachodzą w procesach biznesowych.

Wiele problemów z uzyskaniem systemów dobrej jakości wynika z faktu weryfikowania tylko tych elementów, które zostały zrobione, a nie pod kątem tego, czego brakuje. Te brakujące elementy (np. bezpieczeństwo danych, funkcjonalności systemu) – często przesądzające o postrzeganiu systemu jako niedopracowanego, niefunkcjonalnego, złej jakości – są zauważane dopiero po pewnym czasie i albo jest już za późno na ich uwzględnienie, albo ich uwzględnienie znacząco podnosi koszty i opóźnia termin zakończenia prac. Autorom znany jest przypadek, gdy koszt prac, spowodowany zbyt późnym uwzględnieniem brakujących wymagań, wzrósł 500-krotnie (a kwota wyjściowa nie była wcale mała). Z tego powodu, niektóre firmy wytwarzające oprogramowanie, od samego początku prowadzenia rozmów z zamawiającym, wprowadzają do zespołów projektowych testerów, których zadaniem jest szeroko pojęta weryfikacja wszystkiego, co się wiąże z danym przedsięwzięciem.

Trudnym zadaniem jest również precyzyjne zdefiniowanie, czym jest wada w oprogramowaniu od strony prawnej.

4. JAKOŚĆ A BEZPIECZEŃSTWO

Spośród wielu czynników istotnych w ocenie jakości systemu informatycznego do najważniejszych należy zaliczyć bezpieczeństwo danych i systemu, a więc jego dostępność, ciągłość działania, spójność, integralność, ochronę poufności, umożliwienie odtworzenia danych po awarii itd. Brak testów wydajnościowych sprawia, że wdrażane systemy nie są w stanie obsłużyć wielu równoczesnych użytkowników. Tak było w przypadku wyborów samorządowych w Polsce w 2002 roku. System zbierający dane z lokali wyborczych przetestowano dla niewielu danych i nie przewidziano efektu skali, a mianowicie, że problemy nie rosną proporcjonalnie w stosunku do wzrostu liczby danych, lecz wykładniczo! Po zakończonym głosowaniu, przy jednoczesnej dużej liczbie połączeń z lokali wyborczych przesyłających wyniki, część użytkowników była z systemu usuwana i nie zostały zebrane wszystkie oddane głosy (ponadto nie wiadomo było, z którego lokalu dane otrzymano, a z którego nie). Innym przykładem jest internetowy system Centralnej Komisji Egzaminacyjnej (www.cke.edu.pl), który w 2010 roku nie poradził sobie z obsługą większej liczby użytkowników i wyświetlana była informacja o przeciążeniu systemu. Problem skali dotyczy wielu systemów (bankowych, finansowych, rejestrów danych).

Badania ankietowe przeprowadzone w USA i w Wielkiej Brytanii w 2009 roku wykazały, że 62% firm doświadczyło naruszenia bezpieczeństwa z powodu wadliwego oprogramowania. Tylko co trzecia z badanych firm przeprowadziła rygorystyczne testy w zakresie bezpieczeństwa przed zatwierdzeniem i wdrożeniem programów opracowanych na zewnątrz pomimo, że ponad połowa z ankietowanych firm korzystała z outsourcingu przy opracowywaniu dla nich aplikacji krytycznych. Ponadto zaledwie 34% wytwórców oprogramowania zachowuje pełny cykl życia projektowania oprogramowania z uwzględnieniem aspektu jego bezpieczeństwa [3]. Nawet w przypadku systemów ważnych instytucji i urzędów zdarzają się problemy i niejednokrotnie na stronach WWW wyświetlany jest komunikat o błędzie.

Niestety, nie zawsze jest możliwość wyboru producenta systemu informatycznego lub gotowego systemu odpowiedniego do danych celów. Z kolei, nawet jeśli jest możliwość wyboru producenta, to nie jest to łatwe zadanie. Pewną pomocą może być wybór certyfikowanych dostawców, na przykład posiadających certyfikaty ISO 9001.

Prace prowadzone przez amerykański *Software Engineering Institute*, nad oceną zdolności zleceniobiorców do realizacji zamówień z zakresu wytwarzania oprogramowania, doprowadziły do opracowania pięciostopniowego modelu znanego pod nazwą CMM (ang. *Capability Maturity Model*). Celem stosowania modelu CMM jest ulepszanie procesu wytwarzania oprogramowania. W CMM wyróżniono pięć poziomów dojrzałości (im wyższy poziom, tym organizacja bardziej dojrzała):

- początkowy (ang. *initial*) – brak formalnych procedur, wytwarzanie oprogramowania nie podlega żadnym regułom i jest słabo kontrolowane,
- powtarzalny (ang. *repeatable*) – śledzenie wytwarzania oprogramowania związane jest z kontrolą kosztów, harmonogramów i funkcjonalności,
- zdefinowany (ang. *defined*) – dokumentowanie procesu wytwórczego za pomocą procedur lub instrukcji,
- zarządzany (ang. *managed*) – stosowanie metryk w odniesieniu do procesu i do jakości,

- optymalizujący (ang. *optimizing*) stosowanie praktyk doskonalących proces wytwórczy.

Zasadniczą różnicą między normami ISO 9001 a modelem CMM jest ich przeznaczenie: organizacje, uzyskują certyfikaty ISO 9001 głównie w celu uwiarygodnienia się w oczach klientów, natomiast wprowadzanie modelu CMM pomaga w lepszym zorganizowaniu wytwarzania oprogramowania, w wyniku czego powstające produkty mogą być na wyższym poziomie technologicznym i jakościowym.

5. JAKOŚĆ UŻYTKOWA

Wiele systemów informatycznych służy do obsługi baz danych w celu ułatwienia wprowadzania, przeglądania, modyfikowania i wizualizacji danych. Dlatego też niezwykle ważna jest jakość interfejsu do wykonywania tych zadań, która wpływa na komfort i przyjemność pracy z systemem oraz na ocenę jego użyteczności. Niestety wiele z systemów użytkowanych/ocenianych przez autorów tej pracy ma poważne niedociągnięcia w tym zakresie. Przykładami błędów są:

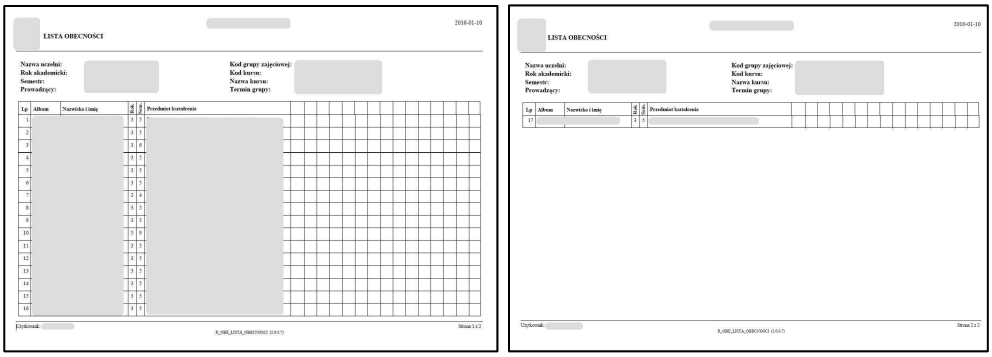
- brak informacji na temat sposobu wprowadzania danych, na przykład daty, numeru telefonu, liczb rzeczywistych itd.,
- zbyt mały rozmiar pola do wprowadzania danych,
- zbyt wiele zbędnych ograniczeń dla danych, na przykład dla kodu miasta, numeru rejestracyjnego pojazdu czy numeru telefonu – zmiana ich dopuszczalnej postaci powoduje konieczność zmian w systemie,
- nie uwzględnianie postaci danych z zagranicy,
- brak wartości domyślnych (początkowych) danych,
- zbyt wiele pól obligatoryjnych i brak ich oznaczenia,
- brak natychmiastowej informacji o błędzie (i przyczynie błędu) w przypadku błędnego wpisania wartości, wyświetlanie komunikatu o błędnym wprowadzeniu danych dopiero po całkowitym wypełnieniu wszystkich pól formularza, wymaganie ponownego wprowadzenia wszystkich danych do formularza,
- brak możliwości sortowania danych według wybranych kryteriów,
- źle zorganizowane wyszukiwanie danych – brak możliwości zmiany tylko niektórych warunków wyszukiwania (często spotykana sytuacja w systemach biur podróży, czy w wyszukiwarkach połączeń komunikacyjnych),
- źle zaprojektowane menu, zbyt duża lub zbyt mała liczba opcji w menu, menu nie intuicyjne, trudne do opanowania,
- niejednolity wygląd formantów, złe rozmieszczenie formantów na formularzach.
- brak czytelnych komunikatów,
- niewłaściwa wizualizacja danych,
- przerywane połączenia z serwerem bez zapisu danych.

6. NIEWŁAŚCIWA REALIZACJA SYSTEMÓW – STUDIUM PRZYPADKÓW

6.1 Przypadek 1 – zła specyfikacja wymagań

W obserwowanej realizacji systemu źle utworzono zespół określający wymagania systemu co zaowocowało złą analizą wymagań. W efekcie powstał system z wieloma

usterkami utrudniającymi pracę lub wręcz uniemożliwiającymi wykonanie niektórych zadań. Na prośbę użytkowników wykonawca część błędów poprawił, ale inne uwagi nie zostały uwzględnione, ze względu na ich dużą czasochłonność i koszt. Na rysunku 1 przedstawiono raporty, które są generowane w postaci nieodpowiadającej prowadzącym zajęcia, bez możliwości ich dostosowania do bieżących potrzeb, na przykład ograniczenia wydruku z dwóch stron do jednej, czy usunięcia zbędnej 50-cio znakowej kolumny zawierającej powtarzający się wieloznakowy symbol przedmiotu kształcenia.



Rys. 1. Nieodpowiednio zaprojektowane listy obecności

6.2 Przypadek 2 – chaotyczne tworzenie i modyfikowanie systemu

W wyniku nieodpowiednio utworzonego zespołu określającego wymagania systemu i źle przeprowadzonej analizy wymagań powstał system pozbawiony istotnych funkcjonalności, za to z wieloma zbędnymi. System jest rozwijany *ad hoc* pod wpływem zgłaszanych uwag, wymusza zmiany dotychczasowych procesów biznesowych w firmie. Przez chaotyczne zmiany w systemie, układ menu jest niespójny i nieintuicyjny. Również formularze i raporty nie mają jednolitej, dobrze zaplanowanej postaci. Wykonawca niechętnie wprowadza modyfikacje, które nie są zgodne z jego wizją, a zamawiający nie potrafi ich wyegzekwować. Brak precyzyjnej umowy pomiędzy wykonawcą a zamawiającym utrudnia prawidłową realizacje systemu.

6.3 Przypadek 3 – brak osób odpowiedzialnych

Do realizacji dedykowanego, bardzo specyficznego przedsięwzięcia, nie wyznaczono osoby odpowiedzialnej ze strony klienta, a zespół określający wymagania został utworzony z osób chętnych do współpracy (niekoniecznie kompetentnych). Zakres przedsięwzięcia wyznaczono bardzo obszerny nie określając harmonogramu ani kosztów. W trakcie realizacji wprowadzano ciągle zmiany w wymaganiach. W efekcie dotychczas opracowany kosztowny system realizuje zaledwie wąski zakres funkcjonalności i zamawiający musi podjąć decyzję, czy kontynuować prace (uwzględniając bardzo wysokie koszty), czy lepiej na tym etapie prace zakończyć, gdyż poniesione koszty już przewyższają spodziewane korzyści.

6.4 Przypadek 4 – jednoosobowa realizacja systemu

Jednoosobowo tworzony system nie posiada żadnej dokumentacji a poprawki są dokonywane na prośbę klienta (oczywiście odpłatnie). Jest to dość często spotykana sytuacja w przypadku małych systemów tworzonych przez jedną osobę. Klient jest bardzo uzależniony od wykonawcy, który jako jedyny zna system i może go modyfikować.

7. PRZYCZYNY NIEPOWODZEŃ REALIZACJI PROJEKTÓW

Wśród przyczyn niepowodzeń przedsięwzięć informatycznych najczęściej wymieniane są: złe harmonogramy (zbyt optymistyczne szacowanie czasu potrzebnego na wykonanie danego zadania), planowane zbyt niskie koszty (w celu wygrania przetargu), brak kompetentnych ludzi do wykonania zadania, brak precyzyjnego, wyczerpującego i jednoznacznego opisu przedmiotu zamówienia, brak analizy ryzyka.

W przypadku zamówień publicznych zamawiający jest zobowiązany do opracowania Specyfikacji Istotnych Warunków Zamówienia (SIWZ) zgodnie z ustawą *Prawo Zamówień Publicznych* (Dz. U. z 2010 r. Nr 113, poz. 759, z późn. zm.). Ustawa ta precyzyjnie określa co powinien zawierać dokument SIWZ, między innymi: opis przedmiotu zamówienia, termin wykonania prac, kryteria oceny ofert, sposób porozumiewania się zamawiającego z wykonawcami, sposób przekazywania istotnych dokumentów takich jak zawiadomienia, oświadczenia, informacje czy wnioski (pisemnie, faksem lub telefonicznie zgodnie z wolą zamawiającego), listę osób uprawnionych do porozumiewania się z wykonawcami itd.

Przyczynami nieporozumień wykonawcy ze zleceniodawcą (a także studenta z nauczycielem) są często ich różne punkty widzenia:

- logika wykonawcy (studenta): zachwyt swoją pracą, oczekiwanie na pochwały, podkreślanie czasu poświęconego na pracę, zmęczenie pracą, ocena włożonego czasu i trudu a nie efektu, ocena produktu zgodnie ze swoją wizją, nie zawsze odpowiadającą wizji zleceniodawcy,
- logika zleceniodawcy (nauczyciela): rozliczanie efektów pracy a więc jakości produktów (bazy danych, oprogramowania, dokumentacji), ich funkcjonalności, zgodności z oczekiwaniami, poprawności działania i bezbłędności a nie czasu realizacji (który być może przekroczył zaplanowany ze względu na brak wiedzy lub złą organizację pracy).

Wdrażane systemy niestety nie są dostatecznie przetestowane głównie z braku czasu. Z doświadczeń firm informatycznych wynika, że głównymi przyczynami niskiej jakości oprogramowania są źle sformułowane wymagania i niewłaściwe (niewystarczające) testowanie. Testowanie powinno być przeprowadzane metodycznie i systematycznie, równolegle do procesu tworzenia, a nie planowane na sam koniec. Dostępne narzędzia ułatwiają prace testerom, pozwalają zautomatyzować wiele prac. Niestety, wprowadzane zmiany i poprawki do systemu wymuszają ponowne wykonywanie testów. Ponadto, wykorzystywanie tych narzędzi wymaga czasu na zapoznanie się z nimi oraz na analizowanie ich wyników.

Bardzo często niedopracowany element, który przez wykonawcę traktowany jest jako rzecz nieistotna, mała usterka czy niedociągnięcie, dla użytkownika może być czynnikiem przesądzającym o odrzuceniu danego produktu.

8. DODATKOWE WYMAGANIA DLA SYSTEMÓW INFORMATYCZNYCH

Rozporządzenie Ministra Pracy i Polityki Socjalnej z dnia 1 grudnia 1998 roku w sprawie bezpieczeństwa i higieny pracy na stanowiskach wyposażonych w monitory ekranowe, określa wymagania jakie pracodawca powinien uwzględnić przy projektowaniu, doborze i modernizacji oprogramowania, a także przy planowaniu wykonywania zadań z użyciem ekranu monitora [9]:

a) oprogramowanie powinno odpowiadać zadaniu przewidzianemu do wykonania,
b) oprogramowanie powinno być łatwe w użyciu oraz dostosowane do poziomu wiedzy i (lub) doświadczenia pracownika,
c) systemy komputerowe muszą zapewniać przekazywanie pracownikom informacji zwrotnej o ich działaniu,
d) systemy komputerowe muszą gwarantować wyświetlanie informacji w formie i tempie odpowiednich dla pracownika,
e) bez wiedzy pracownika nie można dokonywać kontroli jakościowej i ilościowej jego pracy,
f) przy tworzeniu oprogramowania i przetwarzaniu danych powinny być stosowane zasady ergonomii.

Rozporządzenie to określa jakimi kryteriami powinien kierować się pracodawca przy doborze oprogramowania. Kryteria te jednak często są trudne do spełnienia, jeśli pracodawca nie ma odpowiedniego wyboru i żaden z dostępnych systemów nie spełnia wszystkich oczekiwań.

W wielu dziedzinach określa się dodatkowe wymagania na dostarczane produkty na przykład w wojsku, policji, służbie zdrowia, energetyce. Wymagają one odrębnych standardów. Przykładem może być dokument standaryzacyjny NATO AQAP 2110 (*Allied Quality Assurance Publication* – Publikacja Standaryzująca Dotycząca Zapewnienia Jakości), określający wymagania dotyczące systemu jakości dostaw dla wojska lub innych Instytucji Narodowych przewidziany dla dostawców projektujących, prowadzących prace rozwojowe i produkujących/świadczących usługi w ramach kontraktu z wojskiem.

W sektorze zdrowia do najważniejszych zalicza się normę PN-EN 13606 *Informatyka w ochronie zdrowia*. Zarządzanie bezpieczeństwem informacji w ochronie zdrowia z wykorzystaniem normy ISO/IEC 27002 reguluje PN-EN ISO 27799:2010. Innym przykładem może być norma GMP (*Good Manufacturing Practice* – Dobra Praktyka Produkcyjna), stosowana podczas wdrożeń Zintegrowanych Systemów Informatycznych związanych z farmacją, gdzie stała się swoistym standardem. Jej wymogi mogą być zastosowane wszędzie tam, gdzie bezpieczeństwo produkcji i jakość wyrobu jest kluczowa.

Podstawowe pojęcia, definicje i funkcje Systemów Zarządzania i Kontroli Jazdy Pojazdu dla Nadzorowanego Transportu Miejskiego (UGTMS), wymagania systemowe i funkcjonalne a także wymagania dotyczące interfejsów dla systemów zarządzania i kontroli jazdy pojazdów stosowanych w transporcie miejskim zawarto w PN-EN 62290 [6]. Katalog polskich norm dotyczących zastosowania IT w transporcie i handlu jest dostępny na internetowej stronie [7].

Najpopularniejszym międzynarodowym standardem definiującym wymagania odnośnie systemu zarządzania jakością jest opublikowana 15 listopada 2008 roku norma ISO 9001:2008 *Quality management systems – Requirements*. Jej polskim odpowiednikiem jest PN-EN ISO 9001:2009 *Systemy zarządzania jakością – Wymagania*.

9. WNIOSKI

Zarówno analitycy, projektanci, programiści i testerzy a także zamawiający, na każdym etapie wytwórczym powinni stawiać pytanie o cel wykonania danego produktu oraz o jego zgodność z wymaganiami. Zgodność produktu z przeznaczeniem (dobrze zidentyfikowanym celem), wydajność i bezpieczeństwo użytkowania są zasadniczymi oczekiwaniami użytkowników. Niestety zdarza się, że dostarczony system informatyczny ma wiele nieistotnych funkcjonalności (nie zamówionych i nie wymaganych), a brakuje tych oczekiwanych, podstawowych. Dodawanie zbędnych funkcjonalności czy elementów graficznych i dźwiękowych, gromadzenie nadmiarowych danych zwiększa rozmiary oprogramowania (bazy danych), wydłuża czas realizacji i podnosi koszt systemu, wymaga użycia lepszego sprzętu, spowalnia pracę systemu, utrudnia pracę z systemem i opanowanie jego obsługi. Do dziś jeszcze w wielu firmach wykorzystywane są bardzo proste, ale odpowiednio funkcjonalne systemy bazodanowe z interfejsem tekstowym o bardzo małych wymaganiach sprzętowych. Dobra jakość dawno temu opracowanych systemów sprawia, że ich użytkownicy nie czują potrzeby ich wymiany.

Ocena jakości systemu informatycznego dopiero po całkowitym zakończeniu procesu jego wytwarzania często jest bardzo trudna, pracochłonna i kosztowna. Zapewnienie wysokiej jakości możliwe jest poprzez systematyczną kontrolę produktów pośrednich, testowanie systemu na każdym etapie realizacji projektu począwszy od specyfikacji wymagań. Firmy godne zaufania starają się dostarczać systemy jak najlepszej jakości i w ramach obsługi serwisowej usuwać błędy uwidocznione podczas eksploatacji systemu.

Duże korzyści zapewne przynosiłoby certyfikowanie całego procesu wytwarzania systemu przez audytorów zewnętrznych i natychmiastowe reagowanie na wszelkie niezgodności w realizacji.

10. BIBLIOGRAFIA

[1] Bilski E., Dubielewicz I.: *Cykl życia oprogramowania – modele, procesy, jakość w normach ISO*, Wrocław, Oficyna Wydawnicza Politechniki Wrocławskiej 2007.
[2] Blikle A.: *Doktryna jakości*, Warszawa, 2011.
[3] Muszyński J.: *Błędy oprogramowania główną przyczyną naruszenia bezpieczeństwa*, www.securitystandard.pl (2009-05-05)
[4] Jadczak A.: *6 mld zł na ICT w przetargach publicznych w roku 2010*, Computerworld, www.computerworld.pl (2011-08-22)
[5] ISO/IEC 9126-1:2001 *Software engineering – Product quality – Part 1: Quality model.*
[6] PN-EN 62290-1:2007 *Zastosowania kolejowe - Systemy zarządzania i kontroli jazdy pojazdu dla nadzorowanego transportu miejskiego - Część 1: Zasady systemu i pojęcia podstawowe*.
[7] Portal branży narzędziowej. Zastosowanie IT w transporcie i handlu, 2011, www.narzedziownie.pl/?t=k&i=614&n=26099.
[8] *Miażdżąca opinia PTI o projekcie Planu Informatyzacji Państwa*, www.pti.org.pl.
[9] *Rozporządzenie Ministra Pracy i Polityki Socjalnej z dnia 1 grudnia 1998 r. w sprawie bezpieczeństwa i higieny pracy na stanowiskach wyposażonych w monitory ekranowe*. Dz.U. 1998 nr 148 poz. 973, Warszawa, 1998.