

Zygmunt MAZUR, Hanna MAZUR, Teresa MENDYK-KRAJEWSKA  
Politechnika Wrocławska, Instytut Informatyki

## KONTROLA JAKOŚCI WEWNĘTRZNEJ I KWALIFIKACJE ZESPOŁU PROJEKTOWEGO A JAKOŚĆ SYSTEMU BAZY DANYCH

**Streszczenie.** Ocena jakości systemu bazy danych, pomimo, że w dużej mierze zależy od wielu atrybutów mierzalnych, jest również oceną subiektywną. Jakość systemu zależy nie od tylko od tego, ile system ma zalet, ale również od tego, w jakim stopniu jest pozbawiony wad. Ze względu na wiedzę, innowacyjność, zapał do pracy i niskie koszty zatrudnienia, producenci oprogramowania – tworząc zespoły projektowe – zatrudniają ludzi młodych, w tym również studentów kierunków informatycznych. Celem pracy jest podkreślenie roli oceny jakości wewnętrznej wytwarzanego systemu bazy danych oraz wpływu na jakość systemu nie tylko wiedzy, ale także doświadczenia, cech osobowościowych i postaw członków zespołów wykonawczych (niezależnie od fazy projektu, w której uczestniczą), w tym świadomości celu i potrzeby zgodności ze specyfikacją każdego wykonywanego zadania.

**Słowa kluczowe:** jakość systemu bazy danych, model jakości, zespół projektowy

## INTERNAL QUALITY CONTROL AND PROJECT TEAM QUALIFICATIONS VERSUS DATABASE SYSTEM QUALITY

**Summary.** Although quality evaluation of a database system is dependent on numerous measurable attributes, it is also subjective. The quality of a system depends not only on the number of its advantages, but also on to what extent it is free from faults. Taking into account knowledge, innovativeness, eagerness to work and low costs of employment, software producers employ young people, including computer science students to set up their project teams. The aim of this work is to stress the role of internal quality evaluation of a produced database system and the influence on the system quality of not only knowledge, but also personal qualities, experience and attitude of the executive team members (irrespective of the phase of the project they participate in), including the awareness of the project's aim and the need for accuracy with the specification of each realised task.

**Keywords:** quality of database system, quality model, project team

## 1. Wprowadzenie

Zagadnienie jakości systemów baz danych (ang. *database system quality*) jest wielopłaszczyznowe i obejmuje jakość oprogramowania (ang. *software quality*) oraz jakość bazy danych (ang. *database quality*) – a więc wszystkie te elementy, które w sumie wpływają na ciągłość i poprawność działania systemu (niezawodność), odpowiednią do potrzeb (akceptowalną) wydajność, elastyczność (na poziomie narzędziowym i sprzętowym umożliwiającą dostosowanie do specyficznych wymagań), możliwość i łatwość rozbudowy systemu, satysfakcję z jego użytkowania, a także zapewniają integralność, spójność i poufność gromadzonych i przetwarzanych danych. Użytkownik określając swoje wymagania i oczekiwania, niektóre z nich uznaje za bezwzględnie konieczne (np. bezpieczeństwo danych), a inne za życzeniowe (np. ergonomiczny interfejs czy możliwość zmiany kolorów na formularzu). Wymaganiom tym można przypisać wagi oznaczające priorytet ważności dla danego systemu (np. w zależności od potrzeb klienta). Atrybuty i ich miary jakości powinny być ustalone, a następnie wyznaczone i kontrolowane od samego początku tworzenia projektu, przez wszystkie jego etapy. Podejście takie jest zgodne z modelem oceny jakości wewnętrznej zdefiniowanym w normie ISO/IEC 9126-1:2009 [11].

Na początku 2009 roku trzydzieści organizacji związanych z wytwarzaniem oprogramowania podjęło się opracowania listy najniebezpieczniejszych błędów popełnianych przez programistów. Do błędów takich zalicza się na przykład pozostawianie luki w oprogramowaniu (ang. *backdoor*), umożliwiającej przeprowadzanie ataków, czy zapisywanie haseł w kodzie programu. Podjęta inicjatywa zmierza do określenia minimalnych wymagań dla tworzonego oprogramowania użytkowego. Pełna lista ostatnio opublikowanych błędów znajduje się na stronie CWE (Common Weakness Enumeration) [16]. Być może dobrym rozwiązaniem wymuszającym zapewnienie odpowiedniej jakości oprogramowania byłaby jego certyfikacja przeprowadzana przez jednostki zewnętrzne (audytorów), ale jak na razie, nie jest ona wymagana i przeprowadzana.

Nie negując tego, co na temat jakości systemów baz danych zostało dotychczas wypowiedziane i napisane, zdaniem autorów, zasadniczą kwestią w zapewnieniu im wymaganej jakości jest kontrola jakości wewnętrznej oraz kompetentni i rzetelnie pracujący ludzie. Właściwy projekt, implementacja, testowanie, dokumentowanie, wdrożenie i utrzymywanie systemu z dużym prawdopodobieństwem zapewni zadowolenie z jego użytkowania. Dostępne narzędzia wspierają projektowanie i wytwarzanie systemów, ale w dalszym ciągu dużo zależy od ludzi uczestniczących w tym procesie. Na efekt końcowy wpływa również brak udziału w projekcie ludzi z odpowiednimi kwalifikacjami i predyspozycjami, potencjalnych użytkowników systemu, ekspertów dziedzinowych, uczestników procesów biznesowych – ze względu, na przykład, na zaniedbania przy tworzeniu zespołów projektowych, trudności

z pozyskaniem wysokiej klasy specjalistów lub dodatkowe (wysokie) koszty ich zatrudnienia.

Jakość systemu bazodanowego zależy od jakości bazy danych, obsługującego ją oprogramowania, wykorzystywanego systemu zarządzania bazą danych i całego środowiska teleinformatycznego. Do zagadnienia zapewnienia wysokiej jakości oprogramowania dochodzi więc, sprawiające często bardzo duże trudności, zadanie prawidłowego zaprojektowania bazy danych (przedstawione np. w [8]), jej implementacji, optymalizacji, użytkowania i utrzymywania, zabezpieczania, odtwarzania po awarii itd.

Wytwarzane obecnie systemy informatyczne, w tym systemy baz danych, są coraz bardziej złożone i zapewnienie im pożądanej jakości jest zadaniem niezmiernie trudnym. Wiele zależy w tej dziedzinie od wspierających proces wytwarzania metodyk inżynierii oprogramowania, narzędzi programistycznych oraz edukacji w tym zakresie.

Nauczyciele akademicy prowadzący zajęcia na kierunkach informatycznych mają duży wpływ zarówno na wiedzę przekazywaną studentom z zakresu metodyk, technik i narzędzi związanych z projektowaniem baz danych i wytwarzaniem oprogramowania, jak również na kształtowanie cech i postaw niezbędnych do pracy w zespołach projektowych (takich jak komunikatywność, systematyczność, odpowiedzialność, krytyczność, samokrytyczność i umiejętność reagowania na krytykę, rzeczowość i merytoryczność argumentacji itd.).

Celem pracy jest podkreślenie roli kontroli jakości wewnętrznej wytwarzanego systemu bazodanowego oraz tzw. miękkich aspektów wytwarzania systemów bazodanowych, czyli kwalifikacji zespołu projektowego. Oprócz wiedzy merytorycznej i technicznej niezwykle ważne są odpowiednie kompetencje<sup>1</sup>, w tym umiejętności (komunikacyjne, mediacyjne, negocjacyjne), cechy osobowościowe i postawy członków zespołów wykonawczych, świadomość celu i potrzeby zgodności ze specyfikacją każdego wykonywanego zadania oraz umiejętność oceny tego. W pracy podkreślono istotne elementy kształcenia studentów kierunków informatycznych w tym zakresie oraz ich wpływ na jakość projektowanych i implementowanych systemów bazodanowych.

## 2. Jakość systemów bazodanowych

Niezwykle trudno jest podać precyzyjną definicję jakości i zgodnie z nią określić jednoznacznie i bezwzględnie jakość danego produktu. Pierwszą definicję jakości traktowanej jako *"pewien stopień doskonałości"* przypisuje się Platonowi (427-347 r. p.n.e.).

---

<sup>1</sup> Kompetencje rozumiane jako wiedza deklaratywna (wiem, co robić), umiejętności (wiedza proceduralna – wiem, jak robić i potrafię to robić) i odpowiednia postawa (chcę i mogę to robić) [6].

W normie terminologicznej PN-ISO 8402:1996 jakość została zdefiniowana jako „ogół cech i właściwości wyrobu lub usługi decydujący o zdolności wyrobu lub usługi do zaspokojenia stwierdzonych lub przewidywanych potrzeb użytkownika produktu” [10]. Od 15 grudnia 2003 roku obowiązuje nowa norma terminologiczna PN-EN ISO 9000:2001, w której jakość zdefiniowana jest jako „stopień, w jakim zbiór inherentnych właściwości spełnia wymagania”. Według Andrzeja Bliklego, „jakość produktu to brak wad w tym produkcie, a wada produktu jest każda taka negatywna cecha produktu – negatywna z punktu widzenia klienta – której klient ma prawo nie oczekiwać” [3]. Bazy danych i oprogramowanie są traktowane jak produkt.

## 2.1. Modele jakości

Różne definicje jakości eksponują jej różne aspekty, na przykład ekonomiczny, techniczny, filozoficzny, socjologiczny. Jakość systemu bazy danych zależy od przyjętej definicji i modelu jakości, zależy od tego, kto i pod jakim względem go ocenia. Cechy (charakterystyki) systemu uwzględniane w ocenie są różne, w zależności od perspektywy, z jakiej dany system jest oceniany: czy jest oceniany z punktu widzenia projektanta bazy danych, użytkownika czy sponsora systemu. Ten sam system może być też zupełnie inaczej oceniany i postrzegany przez różnych użytkowników, a nawet przez tych samych użytkowników w różnych okresach jego użytkowania. Oprócz wielu czynników mierzalnych jest też wiele czynników niemierzalnych mających wpływ na ocenę jakości systemu.

Aby ocena jakości systemu była możliwie jak najbardziej obiektywna, od wielu lat wprowadzane są różne modele jakości technicznej, w których proponuje się pewne zestawy charakterystyk, czyli istotnych cech (wyrażanych opisowo atrybutów abstrakcyjnych), z których każda może być zbudowana w podobny sposób, czyli składać się z wyróżnionych podcharakterystyk. W efekcie otrzymuje się model w postaci drzewa cech (rys. 1). Cechom z najniższego poziomu można przypisać metryki umożliwiające ilościowe (wymierne) pomiary, dzięki którym można wyrazić obiektywnie jakość w kontekście danej charakterystyki.

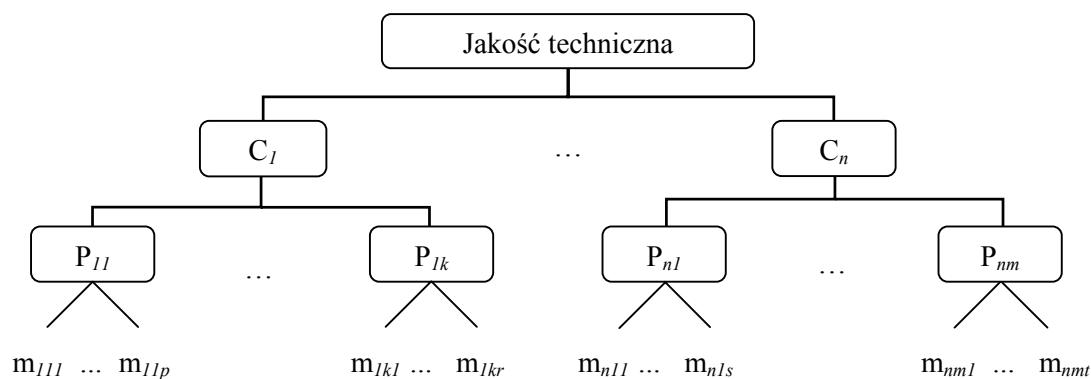
## 2.2. Modele jakości w normie ISO/IEC 9126-1:2001

W normie ISO/IEC 9126-1:2001 [12] wyróżnia się trzy modele jakości, które dotyczą:

- jakości wewnętrznej (ang. *internal quality*),
- jakości zewnętrznej (ang. *external quality*),
- jakości użytkowej (ang. *quality in use*).

Model jakości wewnętrznej i model jakości zewnętrznej obejmuje te same charakterystyki (atrybuty) jakości, do których zalicza się: funkcjonalność (ang. *functionality*), niezawodność (ang. *reliability*), użyteczność (ang. *usability*), wydajność (ang. *efficiency*), utrzymy-

walność (ang. *maintainability*) i przenośność (ang. *portability*) – rozpatrywane jednak z różnych perspektyw.



$C_i$  –  $i$ -ta charakterystyka,

$P_{ij}$  –  $j$ -ta podcharakterystyka  $i$ -tej charakterystyki,

$m_{ijk}$  –  $k$ -ta miara  $j$ -tej podcharakterystyki  $i$ -tej charakterystyki.

Rys. 1. Schemat modelu jakości technicznej

Fig. 1. Scheme of model of technical quality

Jakość wewnętrzna służy do oceny systemu (a właściwie produktów pośrednich) na etapie jego wytwarzania, natomiast jakość zewnętrzna jest oceną po zakończeniu procesu wytwarzania, gdy system został już wykonany. Charakterystyki jakości składają się z podcharakterystyk, dla których mogą być obliczane zdefiniowane metryki. Przykład standaryzowanego opisu metryki o nazwie „Adaptowalność struktur danych” znajduje się w pracy [2].

Jakość użytkowa – na którą składają się cztery charakterystyki: efektywność (ang. *effectiveness*), produktywność (ang. *productivity*), bezpieczeństwo (ang. *safety*) i satysfakcja (ang. *satisfaction*) – jest to ocena systemu przez klienta, o której decyduje możliwość wykorzystania go do planowanych zadań. Ocena jakości użytkowej wyraża zdolność oprogramowania do osiągnięcia założonych celów i do bezawaryjnej pracy. Wyraża też relację zużywanych zasobów do osiągniętej efektywności oraz stopień zadowolenia klienta, na który mają wpływ jego oczekiwania, potrzeby, doświadczenia, upodobania, przyzwyczajenia, gust, percepcja, budowa fizyczna, stan psychiczny itd.

### 2.3. Inne modele jakości

Prace nad modelami jakości trwają od wielu lat, i spośród wielu modeli jakości oprogramowania wymienić można na przykład model jakości Boehma [4] (brakuje w nim atrybutu funkcjonalności), Boeinga (model bardzo rozbudowany, jest w nim zdefiniowanych 15 charakterystyk i 27 podcharakterystyk), FURPS (*Functionality, Usability, Reliability, Performance, Supportability*, czyli funkcjonalność, użyteczność, niezawodność, wydajność, wspie-

ralność), FURPS+ (model FURPS dostosowany do specyfikacji konkretnego zadania) oraz model McCalla (brak w nim atrybutu funkcjonalności).

#### **2.4. Ocena jakości systemu bazodanowego**

Pomimo opracowania wielu modeli jakości oprogramowania, ocena jakości nie jest jednoznaczna i stanowi trudne zadanie, często bardzo kosztowne – szczególnie w odniesieniu do systemów baz danych. Złożoność i skalowalność systemów bazodanowych utrudnia (lub wręcz uniemożliwia) wyodrębnienie charakterystyk i ich pomiary, ze względu na niemożność stworzenia rzeczywistego środowiska eksploatacyjnego i testowego. Przykładowo, jednym z częstych wymagań wobec systemu bazy danych jest odpowiednia wydajność systemu, która może być traktowana jako czas oczekiwania użytkownika na odpowiedź (wynik wykonania zapytania), lub jako możliwość zrealizowania określonej liczby transakcji bazodanowych w jednostce czasu. W systemach o złożonej strukturze bazy, dużej liczbie danych i wykonywanych transakcji, wielu użytkownikach, złożonej logice biznesowej – pomiar i ocena wydajności systemu na etapie jego wytwarzania są zadaniem niezwykle trudnym.

### **3. Jakość systemów bazodanowych a bezpieczeństwo danych**

Bazy danych zawsze stanowiły atrakcyjny cel ataków, dlatego też problem projektowania systemów wysokiej jakości dotyczy ich w sposób szczególny. Na bezpieczeństwo systemów baz danych należy kłaść nacisk już w fazie ich projektowania.

Jakość użytkowanego oprogramowania (systemu zarządzania bazą danych, systemu operacyjnego, przeglądarki internetowej itd.) ma zasadniczy wpływ na bezpieczeństwo systemów informatycznych, a zatem przechowywanych i przesyłanych danych. To właśnie wady oprogramowania stwarzają możliwość podejmowania skutecznych ataków za pomocą specjalnych narzędzi (exploitów) i postrzegane są jako główna przyczyna zagrożeń bezpieczeństwa sieciowego. Wynikają one z błędów programistów oraz niedostatecznego testowania opracowanych systemów, które są coraz bardziej złożone. Informacje o wykrywaniu wad pojawiają się nieustannie [9]. Rozwiązaniem problemu są dostarczane przez producentów tzw. nakładki systemowe lub nowe wersje oprogramowania, jednak w uaktualnionych już systemach często znajdują się nowe luki.

Producenci systemów baz danych podczas ich wytwarzania i testowania muszą zwracać uwagę na wiele różnorodnych aspektów, takich jak: poprawność projektu bazy danych – jej modyfikowalność i skalowalność, zgodność implementacji bazy danych z projektem, optymalizację zapytań, poprawność poszczególnych funkcji i modułów systemu oraz systemu

jako całości. Ponadto muszą oni mieć na uwadze zapewniający ergonomię pracę interfejs, czytelne raportowanie wyników, różne poziomy dostępu do systemu i danych (dla różnych użytkowników). Każdy z tych elementów jest ważny i nie może być zaniedbany, jednak najważniejsze jest zapewnienie bezpieczeństwa zgromadzonym danym – a więc ich dostępności, spójności, integralności oraz ochrony poufności, a także umożliwienie odtworzenia danych po awarii.

Wiele systemów bazodanowych operuje na danych poufnych, tajnych lub ściśle tajnych – jak na przykład bazodanowe systemy bankowe, finansowe, medyczne, administracji państwowej. Zapewnienie wysokiego poziomu ochrony danym wymusza stosowanie odpowiednich systemów zabezpieczeń, rozwiązań zarówno sprzętowych, jak i programowych (na przykład do szyfrowania plików, kontroli dostępu, tworzenia kopii zapasowych i archiwizacji danych itd.) [15]. Zastosowane mechanizmy bezpieczeństwa powinny być adekwatne do stopnia ważności danych i nie mogą wpływać negatywnie na jakość systemu (spowalniając jego pracę poprzez nadmierne kontrole, zbyt częste wykonywanie kopii zapasowych itp.).

Rozwój technologii internetowych umożliwia implementację aplikacji ergonomicznych, zapewniających użytkownikowi komfort pracy z systemem, jak na przykład wybór danych z listy czy podpowiedzi spośród wcześniej wprowadzonych do formularza (i zapamiętanych) danych. Rozwiązania te stwarzają jednak niebezpieczeństwo pozyskania danych z aplikacji przez osoby do tego nieupoważnione.

Jakość systemu bazy danych jest ściśle powiązana z bezpieczeństwem środowiska, w którym system jest eksploatowany. Szczególnie jest to ważne w systemach informatycznych zwanych programowalnymi systemami elektronicznymi PES (*Programmable Electronic System*), które realizują między innymi funkcje sterowania, ochrony i monitorowania oraz obejmują takie elementy sprzętowe, jak zasilacze, czujniki, magistrale komunikacyjne i urządzenia wykonawcze [7]. Ze względu na wagę problemu zapewnienia bezpieczeństwa, opracowywane są normy regulujące zasady wytwarzania, testowania i certyfikowania oprogramowania dla systemów PES. Niestosowanie się do tych norm i brak oficjalnej certyfikacji oprogramowania tych systemów może doprowadzić nie tylko do zakazu stosowania oprogramowania, ale i wykluczenia z rynku wytwarzających je firm, a także urządzeń i systemów zawierających takie oprogramowanie.

Znane są liczne przykłady błędnie działających systemów baz danych, w których doszło do ujawnienia danych poufnych, a usunięcie wad było bardzo kosztowne [13]. Przykładem źle opracowanego systemu może być system do zbierania wyników wyborów samorządowych w Polsce w październiku 2002 roku, w którym testy wydajnościowe zostały przeprowadzone dla zbyt małej liczby przesyłanych danych i nie przewidziano efektu dużej skali. Transmisja danych po zakończonych wyborach była bardzo powolna lub nawet przerywana, przez co nie zebrano danych ze wszystkich lokali wyborczych. Inna niepożądana sytuacja

wystąpiła w lutym 2008 roku przy okazji uruchomienia systemu internetowej rezerwacji biletów na Euro 2008. Po zalogowaniu się do systemu można było oglądać dane osobowe, w tym numery PESEL i dowodów osobistych, wszystkich osób wcześniej zapisanych do udziału w losowaniu. Z kolei w lipcu 2008 roku w internetowym systemie rekrutacji do pracy w Pe-kao SA, w którym nienależycie zabezpieczono dane, upublicznione zostały listy motywacyjne i życiorysy kandydatów (pliki z danymi były umieszczone na serwerze w ogólnodostępnym katalogu publicznym).

Przykładów atakowania systemów bazodanowych w wyniku występowania luk w oprogramowaniu (np. w SQL Server, MySQL czy Oracle) można wymieniać wiele. Szczególnie w starszych wersjach systemów, lecz ciągle powszechnie używanych, znajdują się liczne wady, w tym często o znaczeniu krytycznym, które pozwalają na zdalne wykonanie dowolnego kodu. Jako przykład może posłużyć luka występująca w procedurze `sp_replwritetovarbin` w SQL Server 2000, SQL Server 2000 Desktop Edition, SQL Server 2005 Express Edition, SQL Server 2005 oraz Windows Internal Database [18, 20]. Mimo stosowania coraz bardziej zaawansowanych mechanizmów ochrony w nowych wersjach, wiele popularnych systemów zarządzania bazami danych nadal nie jest pozbawionych wad. Przykładowo, firma Oracle, która przywiązuje dużą wagę do problemu bezpieczeństwa, w styczniu 2008 roku przedstawiła 51 poprawek usuwających między innymi pięć poważnych luk znalezionych w Oracle Database, Oracle Application Server, Oracle E-Business Suite i w innym oprogramowaniu, umożliwiających przeprowadzenie zdalnych ataków. Z kolei na początku 2009 roku opublikowano dalszych 41 poprawek [19]. Najbardziej narażony na ataki jest komponent Oracle Listener odpowiedzialny za komunikację pomiędzy klientem a serwerem oraz między serwerami. Jeśli dany komponent nie jest zabezpieczony hasłem (ustawienie domyślne) – istnieje możliwość przeprowadzenia na niego ataku przy użyciu dostępnego w Internecie skryptu `tnscmd` i uzyskania wielu istotnych informacji na temat systemu bazy danych.

Zagrożenia wynikają też z użytkowania programów napisanych w językach Java oraz JavaScript, w których modułach często wykrywane są niebezpieczne luki.

Jakość systemu i bezpieczeństwo danych zależą także w dużym stopniu od właściwego administrowania danymi. Pracownik, któremu powierza się stanowisko administratora, musi mieć nie tylko rozległą wiedzę teoretyczną i praktyczną, ale również być bardzo odpowiedzialny, systematyczny, komunikatywny i lojalny wobec pracodawcy. Musi przestrzegać wartości etycznych i moralnych ze względu na dostęp do wielu danych o charakterze poufnym bądź tajnym. Powinien szybko lokalizować i usuwać awarie (określać ich przyczyny, ewentualne zagrożenia), a także na bieżąco poznawać i stosować dostępne metody i środki ochrony systemów.



#### 4. Czynniki ludzkie a jakość systemów bazodanowych

Jakość systemu bazy danych zależy ściśle od ludzi związanych z danym zadaniem: analityków i projektantów, którzy powinni być specjalistami w tej dziedzinie, osób pozyskujących wymagania i uczestniczących w wywiadach (przeprowadzających wywiady i dostarczających informacji), ekspertów dziedzinowych, programistów, testerów, wdrożeniowców, administratorów itd. Dlatego też niezwykle istotny jest ich właściwy dobór, a więc poziom wiedzy, cech osobowościowych, chęci, motywacji i umiejętności, które powinny posiadać wybrane osoby, a także przydzielanie ich do odpowiednich zadań.

Z rozmów z pracownikami firm wytwarzających oprogramowanie wynika, że – niestety, w wielu firmach całe zadanie związane z wytworzeniem systemu bazodanowego wykonuje często bardzo mały zespół pracowników, a nierzadko tylko jeden pracownik. Tymczasem dana osoba może być na przykład bardzo dobrym analitykiem czy projektantem baz danych, ale może nie nadawać się na testera oprogramowania.

Producenci systemów baz danych podczas ich wytwarzania i testowania muszą zwracać uwagę na wiele różnorodnych aspektów. Zaniedbanie jednego szczegółu projektowego wymaga niejednokrotnie gruntownych zmian, co może znacząco opóźnić realizację danego zadania i podnieść koszty. Błąd znaleziony na każdym następnym etapie wytwarzania systemu powoduje zwykle wielokrotny wzrost kosztów jego naprawy w stosunku do wcześniejszego etapu [5]. Dlatego niezwykle istotny jest właściwy dobór ludzi do przydzielanych zadań oraz kontrolowanie jakości wewnętrznej, czyli jakości wytwarzanych produktów pośrednich, a w szczególności projektu bazy danych oraz zgodności implementacji bazy z projektem.

Często zdarza się, że producent dostarcza system, który ocenia jako dobrej jakości, i czeka na uwagi klienta, który w zasadzie od razu zauważa jego wady, na przykład brak pewnych funkcjonalności czy nieodpowiednie raportowanie wyników. Przyczyn tej dość powszechnej sytuacji może być wiele, na przykład:

- wady rzeczywiście nie zostały przez producenta zauważone,
- nie umiano ich poprawić w wymaganym czasie (albo w ogóle),
- dostrzeżone wady zostały zlekceważone (w nadziei, że klient ich nie zauważy),
- pierwotnie zauważone wady zostały przeoczone (np. skutek złej organizacji pracy) lub nie zdążono ich poprawić,
- producent poprawiając jeden błąd spowodował kolejny, którego już nie zauważył.

Każda z tych sytuacji nie powinna się zdarzyć przy prawidłowej organizacji procesu wytwórczego.

System ma duże szanse być dobrej jakości, jeśli:

- będą odpowiednio dobrane zespoły kompetentnych osób przydzielonych do konkretnych zadań, ponieważ to ludzie pracujący przy przedsięwzięciu decydują o jego jakości,
- zostaną opracowane dobre harmonogramy prac i będą one systematycznie przestrzegane – aby tak było potrzebne są realne harmonogramy, uwzględniające możliwe opóźnienia i ryzyka, oraz potrzebni są pracownicy zdyscyplinowani, systematyczni, odpowiednio przypisani do zadań (ze względu na kwalifikacje i predyspozycje), i w odpowiedniej liczbie (np. zgodnie z teorią ról zespołowych – zob. tabela 1 i tabela 2),
- zostanie prawidłowo wykonana analiza kosztów i na potrzeby projektu będą zabezpieczone wymagane środki (finansowe, materiałowe, sprzętowe, lokalowe),
- na każdym etapie wykonawcy zadań będą stawiali sobie pytanie o cel i zgodność z wymaganiami (specyfikacją), i będą za wszelką cenę starali się określony cel zrealizować,
- jakość wytwarzanych produktów pośrednich będzie kontrolowana na bieżąco.

W każdej dziedzinie zdarzają się nieudane lub nieukończone w terminie projekty (np. w budownictwie). Często przyczyną tego jest brak środków finansowych na ich ukończenie. Głównym powodem niepowodzenia przedsięwzięć informatycznych jest jednak niedoszacowanie czasu na ich realizację. Z badań przeprowadzonych w 2007 roku przez Tata Consultancy Services wynika, że ok. 41% projektów nie dostarcza oczekiwanych korzyści i zwrotu z inwestycji, 33% nie spełnia wyspecyfikowanych wymagań, 62% kończy się z przekroczeniem harmonogramu, 49% kończy się z przekroczeniem budżetu, 47% przekracza założone koszty utrzymania [17].

#### 4.1. Przykłady wytwarzania systemów bazodanowych

Obserwacja powstających systemów bazodanowych nie napawa optymizmem. Przedstawione cztery przykłady rzeczywistych projektów pozostawione są bez komentarza, gdyż wnioski są oczywiste.

**Przykład 1.** W bardzo dużym systemie dla bardzo dużego przedsiębiorstwa nie zebrano wymagań od potencjalnych przyszłych użytkowników. System przekazany do eksploatacji zawierał typowe, książkowe wady, np. brak niezbędnych, podstawowych funkcjonalności, nie ergonomiczny interfejs, złe raportowanie, kłopoty z drukowaniem, brak możliwości eksportu danych do pliku (np. arkusza kalkulacyjnego). Przedstawiciele producenta systemu niechętnie przyjmowali krytyczne uwagi, zastrzegając, że pewnych rzeczy nie będą mogli poprawić, gdyż opracowanie formularzy i raportów w dostarczonej postaci już i tak zajęło im bardzo dużo czasu. System jest eksploatowany od kilku lat, ciągle testowany i krytykowany przez użytkowników. Niektóre ze zgłaszanych uwag zostały przez wykonawcę systemu uwzględnione, ale nadal jest wiele niedopracowanych elementów utrudniających pracę.

**Przykład 2.** Wraz ze zmianą organizacji pracy w firmie wprowadzono nowy, nieukończony i nie przetestowany system bazy danych. Ze względu na źle przeprowadzoną analizę wymagań funkcjonalnych i pośpiech przy wytwarzaniu systemu popełniono wiele błędów projektowych i implementacyjnych. Do problemów wynikających z reorganizacji w firmie doszły problemy związane z wdrożeniem nowego nieukończonego systemu, w którym brakuje wielu potrzebnych funkcjonalności. Co pewien czas do systemu dodawana jest nowa funkcjonalność, która czasami wymaga dodatkowych atrybutów danych i zmian w schemacie bazy danych. Krytyczne uwagi użytkowników są przez wykonawcę całkowicie ignorowane i kwitowane słowami, że lepszy jest taki system, niż żaden.

**Przykład 3.** Bardzo duży system bazy danych został zamówiony przez organizację społeczną. Zgodnie z zasadą, że w zespole projektowym powinni być przedstawiciele klienta, w organizacji powstał zespół roboczy klienta, w skład którego weszły osoby chętne, niestety, oddalone geograficznie, niekoniecznie eksperci dziedzinowo-biznesowi, bez żadnych określonych zadań i nie rozliczani z tej pracy (praca społeczna). Powstający projekt opiera się na wywiadach przeprowadzanych z tymi osobami (kontakty są telefoniczne, poprzez pocztę elektroniczną lub osobiste). Rodzi się zatem pytanie o odpowiedzialność ludzi z zespołu roboczego klienta, szczególnie istotną w przypadku złego wykonania specyfikacji systemu (a potem systemu). Obciążenie ich odpowiedzialnością powinno wymuszać rozliczanie ich z wykonanej pracy i odpowiednie wynagradzanie. Od początku należałoby też ustalić, kto poniesie koszty ewentualnych koniecznych poprawek w systemie.

**Przykład 4.** Wiele lat temu został opracowany system dla firmy państwowej. Na początku obejmował niewielki zakres działalności firmy, zatem i baza danych była niewielka, kod źródłowy niezbyt długi i skomplikowany, więc nie została opracowana żadna dokumentacja systemu. Zgodnie z potrzebami, informatyk co pewien czas dopisywał pewne funkcjonalności, modyfikował i rozbudowywał bazę i oprogramowanie. Po latach powstał obszerny i złożony system, niestety, nieudokumentowany, a jedyną osobą, która go zna, jest informatyk.

#### **4.2. Jakość systemów baz danych opracowywanych przez studentów**

Ze względu na wiedzę, innowacyjność, zapał do pracy oraz niskie koszty zatrudnienia, producenci oprogramowania, tworząc zespoły projektowe, zatrudniają ludzi młodych, w tym również studentów kierunków informatycznych.

Wieloletnia obserwacja pracy studentów w czasie zajęć projektowych przy opracowywaniu systemów bazy danych w zespołach dwuosobowych potwierdza, że dobrze dobrane pary rzetelnych i sumiennych studentów, dokładnych i systematycznych, przestrzegających harmonogram i wymagania, opracowują systemy o wysokiej jakości w zaplanowanym terminie. Systemy te są dobrze zaprojektowane, właściwie zaimplementowane, przetestowane i udo-

kumentowane. Niestety, część studentów ma kłopoty ze spełnieniem postawionych wymagań głównie z dwóch powodów: ze względu na brak odpowiedniej wiedzy i umiejętności oraz ze względu na brak systematyczności, rzetelności lub chęci (do czego niejednokrotnie się sami przyznają). Jest wyraźna różnica w jakości systemów zaprojektowanych i wytworzonych przez osoby świadome celów opracowania każdego elementu systemu i realizujące zadania zgodnie z wymaganiami i przeznaczeniem.

Studenci są znacznie bardziej krytyczni wobec systemów, których nie są autorami, niż wobec własnych. Radość z opracowania własnego dzieła, wkład pracy i czas poświęcony na wykonanie zadania powodują, że bardzo się do swojego systemu przywiązują i z trudem zauważają popełnione błędy w projekcie i implementacji bazy danych, oprogramowaniu, opracowanych zapytaniach, formularzach i raportach czy w dokumentacji. Bardzo eksponują błędy zauważone w pracach kolegów, natomiast gdy źle działa system przez nich opracowany, problem minimalizują lub wręcz bagatelizują.

Obserwowane są różne rodzaje postaw członków zespołu:

- każda praca przekazywana do oceny jest dobra – postawa idealna, niestety, niezwykle rzadko spotykana,
- oceniane prace mają wiele wad, których autor nie zauważa, nie rozumie, nie umie poprawić, trudno mu się pogodzić ze złą oceną pracy (w jego mniemaniu poprawnej) – postawa negatywna, dość często spotykana,
- prace przekazywane do oceny zawierają dużo wad, a każde zwrócenie uwagi jest komentowane stwierdzeniem: „to drobiazg, ja to mogę poprawić”, po czym kolejna wersja zawiera błędy tego samego typu, tylko w innym miejscu, nadal są niepoprawione błędy wcześniej nie omówione, dodatkowo występują inne błędy – postawa negatywna i utrudniająca współpracę,
- przedłożone do oceny prace zawierają wady, ale reakcja na zwrócenie uwagi jest rzeczowa, merytoryczna, błędy poprawiane są szybko, skutecznie, przy okazji znajdowane są inne (jeśli były), które są też eliminowane; osoba jest bardzo komunikatywna, szybko się uczy na błędach i zdobywa doświadczenie – postawa pozytywna, pożądana.

O ile rzeczą naturalną jest, że w przypadku studentów, którzy są dopiero na etapie zdobywania wiedzy i bez żadnych doświadczeń, spotykane są wszystkie typy postaw, o tyle w profesjonalnych zespołach wytwarzających systemy baz danych osoby z podejściem typu drugiego i trzeciego nie powinny się znajdować. Z takimi osobami praca jest bardzo uciążliwa, trudno jest przestrzegać harmonogramów, osoby te wpływają negatywnie na cały zespół. Dlatego też, kształcąc studentów, należy tak zorganizować zajęcia dydaktyczne, by móc nie tylko zaobserwować postępy w przyswajaniu wiedzy, ale też wykształcanie i rozwój cech poświadczonych w pracy zespołowej i z klientem, by uświadomić studentowi, nad jakimi cechami niezbędnymi w dalszej pracy powinien pracować. Należy zwracać baczną uwagę na wszelkie

zachowania studentów: rzetelność, uczciwość, koleżeństwo, uczynność, spostrzegawczość, systematyczność, obowiązkowość, dbałość o szczegóły, umiejętność argumentacji, negocjacji i przyznania się do błędu, a także na kulturę osobistą, sposób bycia i wyrażania myśli, dbałość o estetykę prac i własny wygląd. Wszystkie te cechy będą niezbędne w pracy zawodowej. W tabeli 1 zestawiono wymagane cechy u pracowników związanych z wytwarzaniem systemu bazy danych w zależności od pełnionej roli, na podstawie ogłoszeń pracodawców poszukujących pracowników na dane stanowisko. Na każdym stanowisku, oczywiście, niezbędne są przede wszystkim kompetencje rozumiane jako wiedza, umiejętności i chęć do pracy.

Tabela 1

## Przykłady cech przydatnych do pełnienia różnych ról w zespole projektowym

Stanowisko w zespole	Cechy pożądane
Kierownik projektu	Umiejętność organizacji pracy, w tym pracy zespołowej, odpowiedzialność, doświadczenie, odporność na stres, umiejętność pracy z klientem, zdolność kierowania zespołem, umiejętność oceny pracy i ludzi
Analitik biznesowy	Intuicja, doświadczenie, wiedza dziedzinowa, dociekliwość, zdolność logicznego myślenia i wyciągania wniosków, komunikatywność, umiejętność pracy z klientem, dyskrecja
Projektant bazy danych	Dokładność, intuicja, doświadczenie, sumiennność, zdolność analitycznego myślenia, dbałość o szczegóły, skrupulatność
Programista	Umiejętność pracy zespołowej, dokładność, odpowiedzialność, terminowość, samodzielność, zdolność do szybkiego uczenia się, odporność na stres
Tester	Systematyczność, cierpliwość, intuicja, kreatywność, perfekcjonizm, dociekliwość, odkrywczność, rozsądek, umiejętność podejmowania decyzji (np. kiedy przerwać testy), dyplomacja (np. w przekazywaniu uwag krytycznych programiście), umiejętność przekonywania (np. o konieczności dokonania poprawki)
Dokumentalista	Dokładność, cierpliwość, sumiennność, terminowość, rzetelność, pedantyczność, znajomość i rozumienie zagadnienia, systematyczność

Studentów należy uczyć nie tylko dokonywania poprawek w opracowywanym systemie, ale przede wszystkim wymuszać na nich, by sami umieli obiektywnie oceniać jakość wykonanych przez siebie prac, lokalizować błędy i je poprawiać. Często spotykana jest postawa roszczeniowa: proszę wskazać, gdzie jest błąd, to go poprawię. Tymczasem największym problemem jest ocena wykonanej pracy, znalezienie niezgodności ze specyfikacją i wymaganiami realizacji zadania, wykazanie spełnienia celu. Tego typu postawy spotykane są nawet u studentów ostatnich lat studiów, co bardzo im utrudnia pisanie i poprawianie prac dyplomowych, gdyż nie zauważają popełnianych przez siebie błędów.

Wieloletnie obserwacje pracy różnych zespołów i osiąganych przez nie efektów wskazują, że nie tylko wykształcenie, wiedza i umiejętności, ale również typ osobowości mają istotny wpływ na efektywność zespołów roboczych [14].

Według teorii ról zespołowych (ang. *team role theory*) Belbina, w każdym zespole powinny się znaleźć osoby o cechach wymienionych w tabeli 2 [1].

Tabela 2

## Role, zalety i wady członków zespołu według teorii ról zespołowych Belbina

Rola w zespole	Zalety	Wady
Kierownik	Zdolności organizacyjne, dokładny, drobiazgowy	Nie umie zmotywować i zachęcić członków grupy do działania
Przewodnik	Umie zmotywować do działania	Nie potrafi zaplanować szczegółów
Twórca	Kreatywny, inspirujący, otwarty, generuje ciągle nowe pomysły	Szybko traci zainteresowanie danym pomysłem, bo już zajmuje się nowym, pomysły często nierealne do wykonania
Radykał	Prowokując (świadomie) odmiennym zdaniem pobudza do znalezienia rozwiązania	Irytuje odmiennym zdaniem od pozostałych członków zespołu
Harmonia grupy	Zdolności interpersonalne, łagodzi konflikty	Skupia się na relacjach między członkami zespołu zamiast na pracy
Ekspert techniczny	Posiada wiedzę techniczną, dostrzega szczegóły	Zbyt długo zajmuje się detalami
Krytyk	Skłonność do oceniania, poprawiania	Skłonność do krytykowania, wytykania błędów
Koordinator	Poczucie odpowiedzialności za innych, za zespół	Zestresowany, choroby psychosomatyczne
Polityk	Zdolności interpersonalne, manipulacja członkami zespołu, charyzmatyczność	Manipulowanie ludźmi
Promotor	Zdolności interpersonalne, manipulacja ludźmi spoza zespołu, promowanie zespołu na zewnątrz	Manipulowanie ludźmi

Jedna osoba ma zazwyczaj dwie lub trzy takie cechy, w tym jedną dominującą. Badania przeprowadzone przez Belbina wykazały, że poziom wykonania zadania jest wprost proporcjonalny do poziomu wiedzy merytorycznej, ale tylko do pewnego momentu, natomiast stały wzrost zapewnia właściwy dobór również osobowości. W wyniku dalszych badań wyodrębniono kolejne role zespołowe: kreator (innovacyjny i samodzielny), ewaluator (wnikliwy i analityczny), koordinator (intuicyjny i poważny), implementer (pracowity i konsekwentny), perfekcjonista (precyzyjny i pilny), poszukiwacz źródeł (optymistyczny i pomysłowy), lokomotywa (dynamiczny i ambitny), dusza zespołu (spokojny i bezkonfliktowy), specjalista (profesjonalny i badający).

Badania i obserwacje wykazują, że osoby, które potrafią współpracować, negocjować i konsekwentnie dążyć do celu, są bardzo pożądane przez pracodawców, dlatego należy te cechy kształtować i rozwijać równocześnie ze zdobywaniem wiedzy specjalistycznej.

## 5. Zakończenie

Dobrze zaprojektowany i wdrożony system informatyczny to taki, który zapewnia niezawodne działanie w określonym zakresie funkcjonalnym oraz komfort pracy, spełniając ocze-

kiwania użytkowników. W systemach bazodanowych wiele błędów jest wykrywanych dopiero w czasie eksploatacji. Niestety, wiele firm wytwarzających te systemy uznaje za dobry system, którego błędy nie uwidaczniają się podczas pracy (nie występują dla typowych danych). Dla producenta oprogramowania jest korzystniej (ze względu na koszty), dopuścić do użytku wadliwy program, i usuwać błędy dopiero po ich wykryciu przez użytkownika. Testowanie w warunkach rzeczywistych nie zawsze jest możliwe, a czasami musiałoby być bardzo długotrwałe i kosztowne ze względu na różnorodność indywidualnych konfiguracji i konieczność przetestowania systemu dla ogromnej liczby równocześnie dokonywanych operacji. Producenci często odpowiednio zabezpieczają się w umowach licencyjnych, nie gwarantując poprawnej pracy systemu w każdej konfiguracji.

Wszyscy członkowie zespołu opracowującego system bazodanowy (analitycy, projektanci, programiści, użytkownicy, testerzy, sponsorzy) muszą mieć świadomość konieczności zapewnienia dobrej jakości wytwarzanego produktu. Jednym z najważniejszych warunków powodzenia projektu jest właściwa rekrutacja członków zespołu wykonawczego i dobór zadań odpowiednio do ich predyspozycji i kompetencji.

Systemy bazodanowe są głównie wykorzystywane do wprowadzania, gromadzenia, przeglądania, modyfikowania i wizualizacji danych. Dlatego też niezwykle ważna jest jakość interfejsu do wykonywania tych zadań, który wpływa na komfort pracy oraz na ocenę jego użyteczności. Niestety, wiele systemów ma poważne niedociągnięcia w tym zakresie. Bywa też, że ignorowane przez wykonawcę błędy językowe (np. w komunikatach) dla użytkownika są tak rażące, że przeszkadzają w pracy z systemem.

Na każdym etapie wytwarzania systemu wykonawca powinien kontrolować zgodność opracowania danego elementu z wymaganiami i celem. Każda funkcjonalność, każdy artefakt, aby były dobrze wykonane i służyły użytkownikowi, muszą być zgodne ze specyfikacją, muszą odpowiadać określonej procesowi biznesowemu. Stopień, w jakim spełniona jest ta zgodność ze specyfikacją, wpływa na jakość produktu. Zgodność z celem jest w procesie tworzenia systemu bazodanowego oczywistym wymaganiami. Niestety, zdarza się, że system ma wiele nieistotnych funkcji (nie zamówionych i nie wymaganych), kosztem zapewnienia tych podstawowych w pełnym zakresie. Dodatkowa zbyteczna funkcjonalność, grafika czy niepotrzebnie gromadzone dane zwiększają rozmiar bazy i oprogramowania prowadząc do spowolnienia pracy systemu i utrudnienia jego obsługi. W wielu firmach nadal wykorzystywane są proste, spełniające oczekiwania użytkownika systemy bazodanowe z interfejsem tekstowym o bardzo małych wymaganiach sprzętowych, ponieważ wysoka jakość wdrożonych systemów sprawia, że użytkownicy przez wiele lat nie odczuwają potrzeby ich wymiany.

**BIBLIOGRAFIA**

1. Belbin M.: Twoja rola w zespole. Gdańskie Wydawnictwo Psychologiczne, Gdańsk 2003.
2. Bilski E., Dubielewicz I.: Cykl życia oprogramowania – modele, procesy, jakość w normach ISO. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2007.
3. Blikle A.: Doktryna jakości. Warszawa 2007.
4. Boehm B., Brown J., Lipow M., MacCleod G.: Characteristics of software quality. North Holland 1978.
5. Górski J.: Inżynieria wymagań, czyli jak unikać zbędnej pracy. Informatyka, nr 3/98, Warszawa 1998.
6. Kossowska M., Sołtysińska I.: Szkolenia pracowników a rozwój organizacji. Oficyna Ekonomiczna, Kraków 2002.
7. Kwiecień A.: Certyfikacja oprogramowania. Napędy i sterowanie, Racibórz 2008.
8. Mazur H., Mazur Z.: Projektowanie relacyjnych baz danych. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2004.
9. Mendyk-Krajewska T., Mazur Z.: Zagrożenia baz danych w aspekcie wad oprogramowania systemów komputerowych. Prace Naukowe Instytutu Informatyki P.Wr., red. Z. Mazur, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2009.
10. Norma PN-ISO 8402:1996 Zarządzanie jakością i zapewnienie jakości – Terminologia.
11. Norma PN-EN ISO 9001:2009 Systemy zarządzania jakością – Wymagania (ang. EN ISO 9001:2008 Quality management systems – Requirements), 2009.
12. Norma ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model.
13. Patton R.: Testowanie oprogramowania. MIKOM, Warszawa 2002.
14. Phillips J.: Zarządzanie projektami IT. Wydawnictwo HELION, Gliwice 2005.
15. Stokłosa J., Bilski T., Pankowski T.: Bezpieczeństwo danych w systemach informatycznych. PWN, Warszawa 2001.
16. The 2009 CWE/SANS Top 25 Most Dangerous Programming Errors. Common Weakness Enumeration Report, [cwe.mitre.org](http://cwe.mitre.org), 2009.
17. A Call for Raising Industry Benchmarks in IT Service Delivery. Tata Consultancy Services, [www.tcs.com/thought\\_leadership/Pages/Raising-Industry-Benchmarks-IT-Service-Delivery.aspx](http://www.tcs.com/thought_leadership/Pages/Raising-Industry-Benchmarks-IT-Service-Delivery.aspx), 2007.
18. [www.microsoft.com/poland/technet/security/bulletin/MS09-004.msp](http://www.microsoft.com/poland/technet/security/bulletin/MS09-004.msp).
19. [http://bezpieczenstwo.onet.pl/1526563,item,0,Oracle\\_lata\\_41\\_dziur,wiadomosci.html](http://bezpieczenstwo.onet.pl/1526563,item,0,Oracle_lata_41_dziur,wiadomosci.html).
20. [www.heise-online.pl/security/news/item/Krytyczna-luka-w-zabezpieczeniach-starszych-wersji-SQL-Server-775785.html](http://www.heise-online.pl/security/news/item/Krytyczna-luka-w-zabezpieczeniach-starszych-wersji-SQL-Server-775785.html).



Recenzenci: Prof. dr hab. inż. Stanisław Kozielski  
Dr inż. Łukasz Wyciślik

Wpłynęło do Redakcji 31 stycznia 2010 r.

### **Abstract**

Although quality evaluation of database system is dependent on numerous measurable attributes, it is also subjective. The quality of a system depends not on the amount of its advantages, but also on how much it is free from faults. Scheme of the model of technical quality is presented in fig. 1. Taking into account knowledge, innovativeness, eagerness to work and low costs of employment, software producers employ young people, including computer science students to set up their project teams. The aim of this work is to stress the role of internal quality evaluation of a produced data base system and the influence on the system quality of not only knowledge, but also personal qualities, experience and attitude of the executive team members (irrespective of the phase of the project they take participate), including the awareness of the aim of and need for accuracy with the specification of each realized task. Table 1 presents examples of features needed to fulfil various roles in a project team, while table 2 introduces roles, merits and shortcomings of team members according to Belbin's team roles theory.

### **Adresy**

Zygmunt MAZUR: Politechnika Wroclawska, Instytut Informatyki, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Polska, [zygmunt.mazur@pwr.wroc.pl](mailto:zygmunt.mazur@pwr.wroc.pl) .

Hanna MAZUR: Politechnika Wroclawska, Instytut Informatyki, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Polska, [hanna.mazur@pwr.wroc.pl](mailto:hanna.mazur@pwr.wroc.pl) .

Teresa MENDYK-KRAJEWSKA: Politechnika Wroclawska, Instytut Informatyki, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Polska, [teresa.mendyk-krajewska@pwr.wroc.pl](mailto:teresa.mendyk-krajewska@pwr.wroc.pl) .