INFORMATYKA EKONOMICZNA BUSINESS INFORMATICS 4(30) · 2013

ISSN 1507-3858

Grzegorz Tomala
Szkoła Główna Handlowa w Warszawie
e-mail: tomala.grzegorz@gmail.com

Jarosław Wilk
Politechnika Warszawska
e-mail: jaroslaw.wilk@ee.pw.edu.pl

WZORZEC SPECYFIKACJI WYMAGAŃ SYSTEMOWYCH DLA PRZEDSIĘWZIĘĆ DOSKONALENIA ISTNIEJĄCYCH SYSTEMÓW OPROGRAMOWANIA BIZNESOWEGO*

Streszczenie: Celem niniejszego artykułu jest zaproponowanie, w odniesieniu do przedsięwzięć rozwoju systemów oprogramowania biznesowego (*Business Software Systems Development and Enhancement Projects*), koncepcji uruchomienia prac analitycznych na dwóch poziomach szczegółowości: wysokim, opisującym zmiany, które wprowadza dany projekt, oraz niskim, opisującym wszystkie funkcjonalności od początku budowy systemu. Dla tak zaproponowanej koncepcji utworzono wzorce specyfikacji wymagań, w tym wzorce specyfikacji przypadków użycia, wymagań graficznych ekranu i metryki zmian.

Słowa kluczowe: inżynieria oprogramowania, specyfikacja wymagań oprogramowania, analiza systemowa.

1. Wstęp

Specyfikacja wymagań jest zbiorem wymagań; określa zakres funkcjonalności i ograniczenia zamawianego systemu. Specyfikacja wymagań może obejmować listę zdarzeń niepożądanych i wymagany algorytm odpowiedzi na te zdarzenia [Pressman 2010, s. 51]. Określenie wymagań oprogramowania jest tym etapem projektu informatycznego, gdzie stykają się interesy wszystkich jego udziałowców [Sacha 2010, s. 50; Wiegers 2003; *Guide to the Software*…], m.in.:

* Artykuł zawiera obszerne fragmenty pracy dyplomowej jednego z autorów − G. Tomali, zatytułowanej *Wzorzec specyfikacji wymagań systemowych dla przedsięwzięć doskonalenia istniejących systemów oprogramowania biznesowego,* której promotorem był dr inż. Jarosław Wilk, napisanej na Politechnice Warszawskiej w 2012 r.

- klientów, którzy finansują dany projekt, oczekując, że przyniesie im korzyści biznesowe,
- użytkowników, którzy będą korzystać z nowego systemu i liczą, że ułatwi on wykonywanie zadań,
- kierownika projektu odpowiedzialnego za oddanie gotowego produktu w określonym czasie i przy wykorzystaniu skończonych zasobów,
- deweloperów, którzy na podstawie wymagań konstruują gotowy system,
- testerów, dla których wymagania są wzorcem poprawnego działania systemu,
- dokumentalistów, tworzących instrukcje użytkownika i pliki pomocy,
- personelu prawnego czuwającego nad kwestiami prawnymi,
- analityków systemowych, którzy są odpowiedzialni za wyspecyfikowanie wyżej wymienionych wymagań.

Specyfikacja wymagań jest swego rodzaju łącznikiem pomiędzy światem biznesu a światem projektu. Stanowi często załącznik do kontraktu − na jej podstawie możemy oszacować wielkość zamówienia, a co za tym idzie − pracochłonność danego projektu.

D. Hamlet i J. Maybee wyrazili pogląd, że [Dick, Maybee 2003, s. 10]: „w literaturze poświęconej inżynierii oprogramowania nie ma spójności nazw części opracowania oprogramowania. Panuje zgoda co do działań, jakie należy podjąć, ale nie na ile różnych etapów te działania powinny być podzielone i jak nazwać poszczególne etapy". Dlatego też autorzy niniejszego opracowania, bazując na własnych doświadczeniach zawodowych, przedstawili rekomendowany przebieg projektu informatycznego w podziale na poszczególne etapy i umiejscowili w nim etap specyfikacji wymagań (por. rys. 1), zgodnie z ogólnie przyjętą zasadą, że analiza wymagań jest pierwszym technicznym etapem w procesie wytwórczym oprogramowania. Jest ona podstawą dla kolejnych aktywności wykonywanych w projekcie [Pressman 2010, s. 294].

Rys. 1. Etapy życia projektu informatycznego

Źródło: opracowanie własne.

Proces specyfikacji wymagań niesie ze sobą ryzyko zbudowania systemu, który nie spełnia wymagań zamawiającego. Program naprawczy może być długotrwały i kosztowny, dlatego też bardzo ważna jest wysoka jakość wyspecyfikowanych wymagań − na każdym etapie życia projektu.

Standish Group w swoich badaniach [The Standish Group… 1995], przeprowadzonych wśród kadry menedżerskiej IT, dotyczących czynników sukcesu w projektach informatycznych, podaje, że 13% odpowiedzi wskazuje na jasne sformułowanie wymagań jako czynnik wpływający na sukces projektu, przy czym sukces rozumiany jest jako projekt zakończony w ramach zaplanowanego czasu, w granicach oszacowanego budżetu oraz zgodny z wymaganiami, co stawia ten czynnik na 3. miejscu (zaraz po zaangażowaniu użytkowników w działania projektowe i wsparciu projektu ze strony zarządu) wśród tych mających największy wpływ na powodzenie projektów informatycznych. Biorąc pod uwagę czynniki, które według respondentów opisywanego badania oddziałują na przekroczenie w projekcie planowanego czasu/budżetu/brak zgodności lub niską zgodność z wymaganiami, na 2. i 3. miejscu w rankingu plasują się czynniki odwołujące się do wymagań użytkownika; są to: odpowiednio niekompletna specyfikacja wymagań i zmiana w trakcie trwania projektu specyfikacji wymagań. Z kolei jako przyczyny całkowitego porzucenia projektów wyżej wymienione czynniki zajmują odpowiednio 1. i 6. pozycję.

Kolejnym, ważnym argumentem, świadczącym o istotnej roli specyfikacji wymagań w procesie wytwórczym oprogramowania, jest zależność przedstawiona przez A. Davisa [Davis 1990, s. 13]: „jeżeli koszt jednostkowy równy jeden jest przypisany wysiłkowi potrzebnemu do wykrycia i naprawienia błędu podczas etapu kodowania, to koszt wykrycia i naprawienia błędu podczas etapu określania wymagań jest mniejszy pięcio-, dziesięciokrotnie. Co więcej, koszt wykrycia i naprawienia błędu podczas etapu pielęgnacji oprogramowania jest dwadzieścia razy większy", co wskazuje, jak duży wpływ na koszty projektu może mieć jakość wyspecyfikowanych wymagań.

Na podstawie doświadczenia zdobytego w projektach informatycznych autorzy niniejszego opracowania uważają, że ustandaryzowanie struktury dokumentu wymagań ułatwia zespołowi zorganizowanie swojej pracy i przyspiesza proces tworzenia specyfikacji.

Celem artykułu jest zaproponowanie wzorca specyfikacji wymagań dla przedsięwzięć rozwoju systemów oprogramowania biznesowego. W tekście zostały opisane cechy dobrze wyspecyfikowanych wymagań zgodne z normą IEEE 830-1998, a dla wyżej wymienionych przedsięwzięć zaproponowano wzorzec specyfikacji wymagań systemowych.

2. Cechy dobrze wyspecyfikowanych wymagań

Brakuje jednoznacznych cech dobrze wyspecyfikowanych wymagań − cechy te są różnie przedstawiane przez różnych autorów. Dla większości badaczy dobre wyma-

gania charakteryzują się właściwościami, które pomagają utrzymać spójność działań i powodują, że dokumentacja wymagań jest maksymalnie przydatna w pozostałych etapach cyklu projektowego.

Dobre praktyki i strukturę dokumentu specyfikacji wymagań opisuje standard IEEE 830 [*830-1998 − IEEE Recommended Practice…*]. Według niego cechy dobrze wyspecyfikowanych wymagań powinny spełniać m.in. następujące założenia [*830-1998 − IEEE Recommended Practice…*]:

Poprawność

Poprawność oznacza, że dane wymaganie wypełnia wszystkie lub część potrzeb biznesowych w ramach uruchomienia danego projektu. W celu określenia poprawności należałoby porównać specyfikację wymagań z innymi dokumentami projektu i standardami, które system musi przestrzegać.

Jednoznaczność

Specyfikacja jest jednoznaczna, gdy każde wymaganie w niej zawarte ma tylko jedną interpretację. Aby zapobiec wieloznaczności, dobrą praktyką jest zdefiniowanie słownika skrótów i pojęć, których znaczenie może być niejednoznaczne. Należy również używać formalnych metod opisu wymagań. Wymagania powinny być jednoznaczne dla osób zarówno je tworzących, jak i korzystających z nich w kolejnych etapach projektu.

Kompletność

Specyfikacja wymagań jest kompletna tylko wtedy, gdy spełnione są następujące warunki:

- zawiera wszystkie istotne wymagania (funkcjonalne i niefunkcjonalne),
- określa zachowanie systemu na każdy bodziec wejściowy (zarówno dla danych poprawnych, jak i niepoprawnych),
- nawiązuje do wszystkich diagramów, tabel, pojęć słownikowych.

Wymagania zawierające zapisy typu: „zostaną określone”, „do ustalenia” (*to be determined*) należy uznać za niekompletne.

Spójność

Kryterium spójności oznacza, że jeżeli specyfikacja wymagań nie jest zgodna z dokumentem wyższego poziomu, to nie jest poprawna. Niespójność może wynikać m.in.:

- ze sprzecznych zapisów dotyczących tego samego obiektu (np. jedno wymaganie mówi, że dany raport powinien być prezentowany w formie tabelarycznej, natomiast drugie wymaganie − że należy to uczynić w formie tekstowej),
- ze sprzecznych zapisów dotyczących logiki działania (np. jedno wymaganie mówi, że dany algorytm potrzebuje x danych wejściowych, natomiast drugie wymaganie, że danych wejściowych powinno być y),
- z używania różnych terminów do opisu tych samych obiektów.

Priorytetyzajca

Kryterium priorytetyzacji oznacza, że specyfikacja wymagań powinna być uporządkowana według istotności/stabilności wymagań − ułatwia to pracę przy pro-

jekcie i pozwala zidentyfikować kluczowe funkcjonalności. Określenie stabilności wymagań pomaga rozpoznać te obszary specyfikacji wymagań, w których prawdopodobieństwo zmian w trakcie życia projektu jest najmniejsze − takie wymagania mogą być implementowane w pierwszej kolejności.

Autorzy niniejszego opracowania uważają, że w specyfikacjach wymagań mogą być stosowane również inne kryteria priorytezyzacji od opisanych powyżej.

Weryfikowalność

Weryfikowalność wymagania oznacza, że w bezsporny sposób można stwierdzić, czy na podstawie danego wymagania powstał produkt końcowy, a także czy dany produkt końcowy został wykonany zgodnie z wymaganiem. Specyfikacja wymagań jest weryfikowalna, jeżeli wszystkie wymagania są weryfikowalne. Wymagania, które zawierają zapisy typu: „będzie zachowywać się poprawnie", „interfejs użytkownika będzie przyjazny" itp., należy uznać za nieweryfikowalne.

Modyfikowalność

Specyfikacja wymagań jest modyfikowalna, jeżeli struktura i styl dokumentu pozwalają zmieniać wymagania w prosty, kompletny i spójny sposób. Elementy, które utrudniają modyfikację wymagań, są następujące:

- odwołania do danego wymagania znajdują się w wielu miejscach,
- struktura specyfikacji wymagań jest skomplikowana,
- wymagania w dokumencie zapisane są w sposób losowy.

Możliwość śledzenia powiązań

Możliwość śledzenia powiązań wymagań oznacza, że można sprawdzić pochodzenie poszczególnych wymagań (odwołanie się do dokumentu wyższego poziomu, na podstawie którego zostało utworzone dane wymaganie), a także znane są powiązania danego wymagania z innymi obiektami powstałymi na jego podstawie.

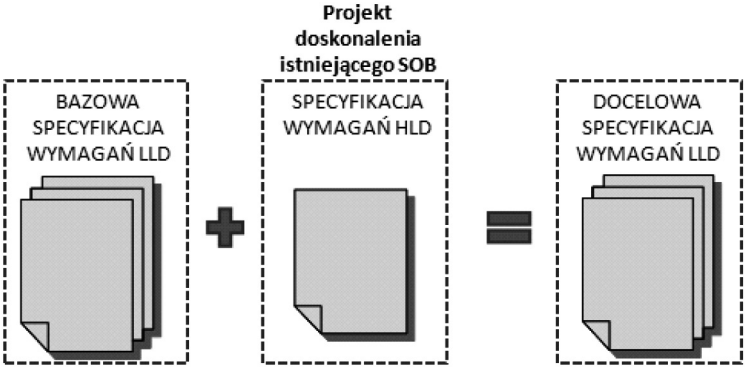
3. Przedsięwzięcia rozwoju systemów oprogramowania biznesowego − koncepcja podziału prac analitycznych

Przedsięwzięcia rozwoju systemów oprogramowania biznesowego (*Business Software Systems Development and Enhancement Projects*) można podzielić na [Czarnacka-Chrobot 2009, str. 29-42]:

- budowę systemu oprogramowania biznesowego (SOB) od podstaw − wyjściowy zbiór funkcji SOB jest pusty,
- doskonalenie istniejącego SOB − polegające na dodaniu/usunięciu/modyfikacji funkcji.

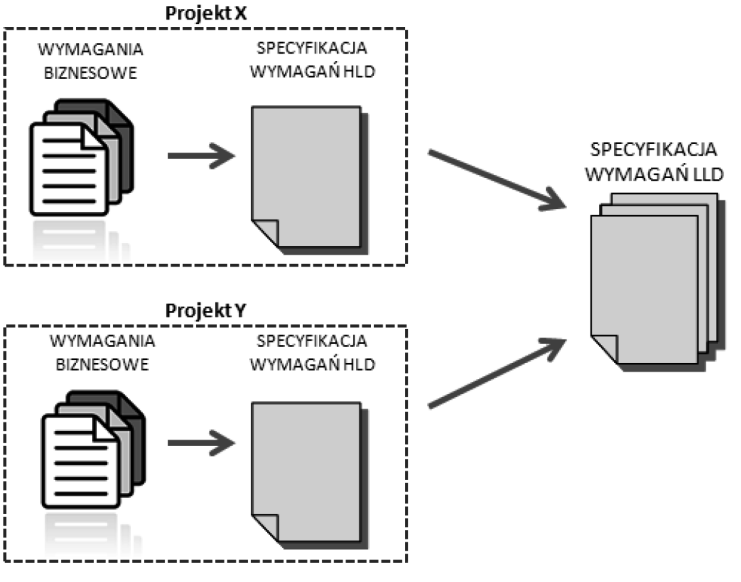
W ramach doskonalenia istniejącego SOB (zakładając, że doskonalenia istniejącego SOB dokonujemy w ramach konkretnego projektu) należy uwzględnić uruchomienie prac analitycznych na dwóch poziomach szczegółowości: wysokim (*High Level Definition* − HLD) oraz niskim (*Low Level Definition* − LLD). W takim przypadku dokument wymagań HLD będzie dokumentem opisującym zmiany, które zostaną wprowadzone w ramach uruchomionego projektu. Będzie to specyfikacja

wymagań opisująca nowe funkcjonalności i modyfikację lub usunięcie już istniejących. Dokument ten będzie podlegał akceptacji oraz wersjonowaniu − na etapie specyfikacji wymagań dokument powinien być konsultowany z zamawiającym dany SOB. Dokument LLD będzie natomiast dokumentem opisującym wymagania systemowe dla całego systemu, tj. specyfikacją dotychczasowych funkcjonalności wraz z uwzględnieniem zmian w ramach realizowanego projektu (por. rys. 2).



Rys. 2. Proces specyfikacji wymagań systemowych dla koncepcji uruchomienia prac analitycznych na dwóch poziomach szczegółowości

Źródło: opracowanie własne.



Rys. 3. Mapowanie wymagań HLD na wymagania LLD

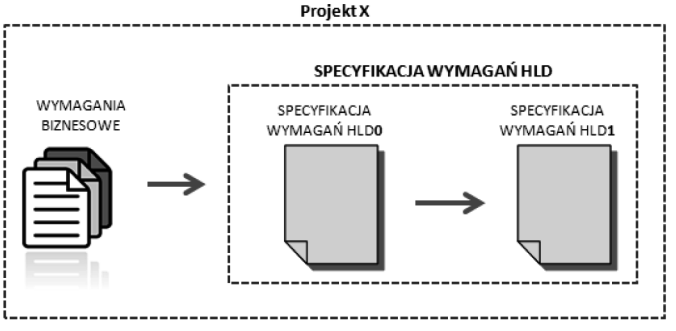
Źródło: opracowanie własne.

Zakres wymagań HLD zdefiniowany jest poprzez wymagania biznesowe dla konkretnego projektu. Specyfikacja wymagań LLD jest zatem o wiele bardziej obszerniejsza (zawiera funkcjonalności opisywane we wszystkich dokumentach HLD), jak również powinna być napisana o wiele bardziej szczegółowo niż na etapie HLD.

Mapowanie wymagań HLD na wymagania LLD przedstawia rys. 3.

Dopuszcza się, aby dokument wymagań HLD składał się z kilku poziomów, np.: HLD poziom 0, HLD poziom 1, … HLD poziom *N*. Każdy z nich powinien być uszczegółowieniem poprzedniego. Takie podejście do pisania wymagań HLD sprawdza się podczas pracy nad dużymi projektami − pozwala na przyrostowe uszczegóławianie wymagań. Wadą omówionego podejścia jest wydłużenie czasu specyfikowania wymagań (zakładając niezmienne zasoby).

Podział specyfikacji wymagań HLD na dwa poziomy: 0 i 1, przedstawia rys. 4.



Rys. 4. Podział specyfikacji wymagań HLD na dwa poziomy: 0 i 1

Źródło: opracowanie własne.

Wzorzec wymagań HLD

Dokument wymagań HLD powinien m.in. opisywać przypadki użycia systemu, wymagania niefunkcjonalne i graficzne projekty ekranu.

Wzorzec dokumentacji przypadków użycia

Przypadki użycia podane w dokumencie HLD powinny opisywać zmiany, które wprowadza dany projekt.

Przypadek użycia powinien mieć unikatowy identyfikator i nazwę odpowiadającą czynnościom, które opisuje. Powinien zawierać także scenariusz główny (sekwencja kroków, która musi zostać wykonana, aby osiągnąć cel dla danego przypadku użycia), a także scenariusze alternatywne. Może on zawierać również dodatkowe informacje dotyczące opisywanej funkcjonalności (Priorytet, Aktorzy, Warunki wstępne, Warunki końcowe itp.).

Tabela 1 przedstawia zaproponowany wzorzec dokumentacji przypadków użycia na etapie specyfikacji wymagań HLD.

Tabela 1. Zaproponowany wzorzec dokumentacji przypadków użycia na etapie specyfikacji wymagań wysokiego poziomu

Identyfikator	
Nazwa	
Priorytet	
Opis	
Aktor główny	
Aktorzy pozostali	
Warunki wywołania	
Rezultat wykonania	
Dane wejściowe	
Dane wyjściowe	
Scenariusz główny	
Scenariusze alternatywne	
Wymagania powiązane	
Uwagi	

Źródło: opracowanie własne.

Wzorzec dokumentacji graficznych projektów ekranu

Każdy prototyp okna, oprócz projektu graficznego, powinien zawierać również opis poszczególnych elementów znajdujących się na formatce, uwzględniając: typ elementu, wartości początkowe, sposób wyliczenia, warunki aktywności, a także reakcje systemu na niepożądane działanie (por. tab. 2).

Tabela 2. Wzorzec opisu pól graficznego projektu ekranu na etapie specyfikacji wymagań wysokiego poziomu

Element	Typ elementu	Wartość początkowa / sposób wyliczenia	Warunki aktywności	Opis

Źródło: opracowanie własne.

Typ elementu oznacza rodzaj danego elementu. Może to być m.in. *button, text box, checkbox*, *radiobutton, drop-down list*. W przypadku elementów prezentujących wybrane informacje (m.in. *text box, drop-down list*) możliwe jest zdefiniowanie wartości początkowych/domyślnych, które będą widoczne podczas prezentacji danego elementu. Każdy z elementów powinien mieć określone warunki aktywności, określające widoczność i możliwości wywołania lub modyfikacji danego elementu z poziomu graficznego interfejsu użytkownika (*Graphical User Interface* – GUI).

Ważnym zagadnieniem jest określenie walidacji pól i reakcji systemu na niepożądane działanie (por. tab. 3). Może to mieć miejsce m.in. w sytuacji, kiedy dane pole tekstowe może przyjmować wyłącznie określony zbiór wartości, a użytkownik wprowadził wartość, która jest poza danym zbiorem. Reakcją na takie działanie może być na przykład wyświetlenie komunikatu z dedykowaną informacją.

Tabela 3. Wzorzec opisu walidacji pól graficznego projektu ekranu na etapie specyfikacji wymagań wysokiego/niskiego poziomu

Element	Rodzaj operacji wykonanej na elemencie	Walidacja, która wywoła reakcję	Reakcja

Źródło: opracowanie własne.

W specyfikacji graficznych interfejsów użytkownika powinna znaleźć się również informacja o tzw. funkcjonalnościach wywoływanych, tj. o przypadkach użycia lub graficznych interfejsach użytkownika możliwych do wywołania z poziomu danego ekranu (por. tab. 4).

Tabela 4. Wzorzec opisu funkcjonalności wywoływanych z poziomu graficznego projektu ekranu na etapie specyfikacji wymagań wysokiego/niskiego poziomu

Sposób wywołania	Opis działania

Źródło: opracowanie własne.

Wzorzec metryki zmian

Metryka zmian służy do rejestrowania informacji o zmianach dokonanych w poszczególnych przypadkach użycia systemu, wymaganiach niefunkcjonalnych oraz graficznych projektach ekranu. Metryka zmian powinna być nierozłączną częścią każdego wymagania. Powinna przechowywać najważniejsze informacje administracyjne, m.in. autora, datę utworzenia i ewentualnie daty i przyczyny modyfikacji danego wymagania (por. tab. 5).

Tabela 5. Wzorzec metryki zmian

Data zmiany	Źródło zmiany	Imię i nazwisko osoby modyfikującej	Szczegóły zmiany

Źródło: opracowanie własne.

Wzorzec wymagań LLD

Wymagania LLD są bardziej szczegółowe niż wymagania HLD i obejmują całościowy opis systemu. Szczegółowość polega m.in. na odniesieniu się do modelu klas, a dla przypadków użycia − na uwzględnieniu wszystkich scenariuszy alternatywnych.

Wzorzec dokumentacji przypadków użycia

Podobnie jak na etapie HLD, przypadek użycia powinien posiadać unikalny identyfikator (różny od identyfikatora z poziomu HLD), nazwę odpowiadającą czynnościom, które opisuje, scenariusze główne i alternatywne. Może on zawierać również dodatkowe informacje (tzw. atrybuty) dotyczące opisywanej funkcjonalności. Atrybuty te mogą być analogiczne do atrybutów na etapie HLD.

Wzorzec dokumentacji graficznych projektów ekranu

Graficzne projekty ekranów opisywane w dokumencie LLD powinny przedstawiać wygląd aktualnych graficznych interfejsów użytkownika. Co do swojej konstrukcji i sposobu prezentacji nie powinny różnić się od projektów z dokumentu HLD.

Podobnie jak w przypadku HLD dla opisu elementów prezentujących wybrane informacje możliwe jest zdefiniowanie wartości początkowych/domyślnych. Każdy z elementów powinien mieć także określone warunki aktywności. Dodatkową informacją, która powinna znaleźć się na etapie specyfikacji LLD, jest odwołanie do modelu klas (jeśli on istnieje). Wzorzec ten przedstawia tab. 6.

Tabela 6. Wzorzec opisu pól graficznego projektu ekranu na etapie specyfikacji wymagań niskiego poziomu

Element	Odwołanie do modelu klas	Typ elementu	Wartość początkowa/ sposób wyliczenia	Warunki aktywności	Opis

Źródło: opracowanie własne.

Określenie walidacji pól, reakcji systemu na niepożądane działanie i informacja o funkcjonalnościach wywoływanych powinny znaleźć się również na etapie LLD − sposób opisu nie powinien się różnić od tego w HLD (por. tab. 3 i 4).

Wzorzec metryki zmian

Metryka zmian na etapie LLD powinna umożliwić rejestrowanie informacji o zmianach dokonanych w poszczególnych przypadkach użycia systemu, wymaganiach niefunkcjonalnych i graficznych projektach ekranów, a także nieść informację o projekcie, w ramach którego dana zmiana została wykonana. Wzorzec metryki zmian na etapie specyfikacji wymagań LLD przedstawia tab. 7.

Tabela 7. Wzorzec metryki zmian na etapie LLD

Data zmiany	Źródło zmiany	Imię i nazwisko osoby modyfikującej	Szczegóły zmiany
Nazwa projektu 1			
Nazwa projektu 2			

Źródło: opracowanie własne.

4. Podsumowanie

Podział prac analitycznych w ramach projektu doskonalenia istniejącego systemu oprogramowania biznesowego na wymagania wysokiego i niskiego poziomu wydaje się zasadny. Podejście takie sprawdza się podczas pracy nad dużymi projektami, wymagającymi szczegółowej analizy. Modyfikacja zakresu w trakcie trwania projektu pozwala na zmianę wymagań w prosty, kompletny i spójny sposób. Ponadto przy omówionym podejściu specyfikacja wymagań staje się bardziej przejrzysta − pozwala szybko zapoznać się z treścią wymagań. Dodatkowo specyfikacja wymagań opisująca zmiany, które wprowadza konkretny projekt, pozwala na dokładniejsze szacowanie zakresu produktu, a co za tym idzie − pracochłonności, kosztów i czasu realizacji danego projektu. Zastosowanie takiego podejścia przy niedużych i mało skomplikowanych projektach nie jest konieczne − wydłuża jedynie czas specyfikowania wymagań (zakładając niezmienne zasoby).

Literatura

830-1998 − IEEE Recommended Practice for Software Requirements Specifications.

Czarnacka-Chrobot B., *Wymiarowanie funkcjonalne przedsięwzięć rozwoju systemów oprogramowania wspomagających zarządzanie*, Szkoła Główna Handlowa w Warszawie − Oficyna Wydawnicza, Warszawa 2009.

Davis A., *Software Requirements Analysis & Specification*, Prentice-Hall, International Edition, 1990, s. 13, [w:] D. Leffingwell, D. Widriga, *Zarządzanie wymaganiami*, Wydawnictwa Naukowo-Techniczne, Warszawa 2000.

Dick H., Maybee J., *Podstawy techniczne inżynierii oprogramowania*, Wydawnictwa Naukowo-Techniczne, Warszawa 2003.

Guide to the Software Engineering Body of Knowledge 2004 Version, red. A. Abran, J. Moore, P. Bourque, R. Dupuis, IEEE Computer Society.

Leffingwell D., Widrig D., *Zarządzanie wymaganiami*, Wydawnictwa Naukowo- Techniczne, Warszawa 2000
Nawrocki J., Materiały z wykładu *Zaawansowana inżynieria oprogramowania*, http://wazniak.mimuw.edu.pl/index.php?title=Zio-8-wyk-toc.
Pressman R., *Software Engineering: A Practitioner's Approach*, McGraw Hill, Boston 2010.
Sacha K., *Inżynieria oprogramowania*, Wydawnictwo Naukowe PWN, Warszawa 2010.
The Standish Group Report, Chaos, The Standish Group International 1995.
Wiegers K., *Software Requirements*, Microsoft Press, Washington 2003.

REQUIREMENTS SPECIFICATION PATTERN FOR BUSINESS SOFTWARE SYSTEMS DEVELOPMENT AND ENHANCEMENT PROJECTS

Summary: The aim of the paper is to propose the concept of analytical work for Business Software Systems Development and Enhancement Projects on two levels of detail: a high level, which describes changes introduced by a new project, and at low level, which describes all functionalities of a system (including those which had been in use before the project concept was created). Requirements specification patterns were proposed for use case specification, graphical user interface and metric changes.

Keywords: software engineering, software requirements specifications, requirements analysis.