



ROBERT WOJTACHNIK

Metoda zbierania wymagań w projektowaniu systemów informatycznych

Method of Gathering Requirements in the Process of Designing IT Systems

Doktor, Politechnika Warszawska, Wydział Inżynierii Produkcji, Instytut Organizacji Systemów Produkcyjnych, Zakład Systemów Informatycznych; dyrektor zarządzający w Benson Consultants Sp. z o.o., Polska

Streszczenie

W procesie wdrożenia systemu informatycznego dochodzi do konfrontacji oczekiwań firmy wdrażającej oprogramowanie z wyobrażeniami dostawcy na temat tych oczekiwań. Ryzyko niepowodzenia projektu wdrożeniowego wynoszące 50% jest powodem zainteresowania tematem projektowania systemów informatycznych. Kluczowym elementem w definiowaniu wymagań jest właściwe postawienie celów projektu i dążenie do ich realizacji. W artykule skoncentrowano się na zdefiniowaniu cech, jakie powinna spełniać dobra specyfikacja, oraz wskazano optymalną strukturę specyfikacji. Struktura specyfikacji jest wynikiem analizy norm, metod projektowych, jak również doświadczeń autora.

Słowa kluczowe: przypadek użycia, wymagania biznesowe, wymagania funkcjonalne, wymagania нефункционалне, wymagania użytkowników, model kontekstowy, mapa procesów

Abstract

During IT system implementation, the expectations of customer met supplier's ideas about this expectations. The risk of IT system implementation failure reach 50%, is the reason for interest in the process design. A key element in defining the requirements is to properly set project goals and pursue them. This article focuses on defining the characteristics that a good specification should meet and indicates the optimum specification structure. The specification structure is the result of standards analysis, design methods as well as the author's experience.

Keywords: usage case, business requirements, functional requirements, non-functional requirements, context model, usage scenario, process map

Wstęp

Prawdopodobieństwo niepowodzenia projektu wdrożenia systemu informatycznego we wstępnej fazie wynosi ok. 50%. Jest to wystarczający powód, żeby

przyjrzeć się tej fazie projektowania systemów informatycznych, zważywszy, że jakość definicji projektu wpływa na cały proces jego realizacji.

W projekcie dochodzi do konfrontacji oczekiwań jednej strony z wyobrażeniem na temat tych oczekiwań przez drugą stronę. Po dwóch stronach barykady spotykają się ludzie o różnej wiedzy, umiejętnościach, doświadczeniu. Jak więc mają się porozumieć, żeby każda ze stron zrealizowała swoje cele? Pierwszym krokiem jest uświadomienie sobie, że uczestnicy projektu nie rozumieją się i potrzebują zastosować wszelkie możliwe środki, żeby tak się nie stało. Z pomocą przychodzą standardy definiowania wymagań IEEE Recommended Practice for Software Requirements Specification IEEE 830-1998. Dokument SRS nie jest dokumentem projektowym, lecz stanowi materiał wejściowy dla analityków i projektantów systemu. Należy również wspomnieć, że zbieranie wymagań na dane jest częścią procesu projektowania architektury korporacyjnej (Sobczak, 2008, s. 315–316).

Błędne wyobrażenie, jakoby system miał służyć użytkownikom w łatwiejszej i wygodniejszej pracy, prowadzi często do definiowania systemu z punktu widzenia użytkownika końcowego. Tymczasem celem wdrożenia systemu informatycznego jest wspieranie celów strategicznych organizacji. Stąd też użytkownik końcowy nie musi rozumieć jego działania w sensie globalnym. Wynika z tego, że właściwe postawienie celów projektu i dążenie do ich realizacji jest właściwym kierunkiem w definiowaniu wymagań.

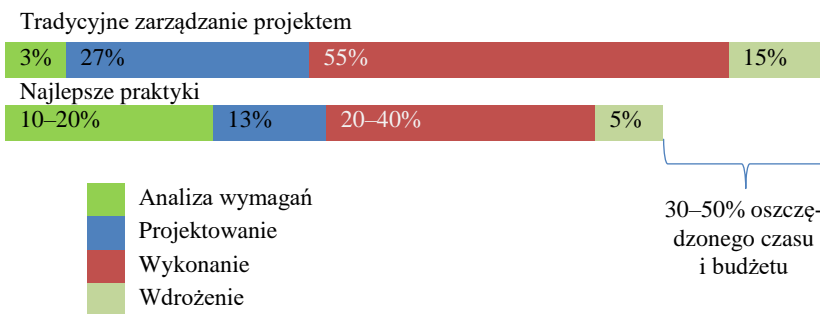
Zasadność zbierania wymagań w projektach systemów informatycznych

Celem przeprowadzenia analizy wymagań jest ustalenie minimalnej liczby warunków, których spełnienie potwierdzi przydatność systemu informatycznego do stawianego mu celu. Analiza to praca, której celem jest zrozumienie klienta, a nie tylko opisanie tego, co powiedział. Rolą analityka nie jest zbieranie wymagań. Wymagania powinien mieć sponsor projektu. O objaśnienie wymagania nie należy pytać użytkownika. Użytkownicy rzadko kiedy czytają przepisy. Źródłem danych w tym zakresie powinni być specjaliści, ustawy i przepisy wykonawcze, literatura fachowa. Analityk biznesowy powinien rozumieć, w jaki sposób działa organizacja, na tej podstawie zdefiniować system informatyczny, który jej pomorze, i opisać wymagania na ten system (Sommerville, 2000).

Drugą kwestią odnoszącą się do wymagań jest ich szczegółowość. Szczegóły są potrzebne w tych miejscach, w których ich brak może doprowadzić do nieprzydatności systemu. Jednym z powodów, dla których nie należy zbierać i dokumentować nadmiernej liczby szczegółów, jest koszt tego procesu.

Według praktyków koszt analizy powinien oscylować w granicach 20% budżetu projektu, a jej czas trwać ok. 50% czasu całego projektu, czyli harmonogramu (Żeliński, 2011). Na rysunku 1 porównano dwie metody organizacji projektu. Pierwsza – tradycyjna – zakłada krótki okres poświęcony na analizę wymagań

(3%). Ponieważ analiza nie została wykonana w należyty sposób, dlatego znacznie bardziej rozbudowany jest etap projektowania aplikacji (27%) i jej produkcji (55%). W drugiej metodzie położono duży nacisk na analizę wymagań (10–20%). Z tak dobrze przeprowadzonej analizy wynika krótki czas projektowania (13%) oraz wykonania systemu (20–40%). Skrócił się również czas wdrożenia systemu z 15 do 5%. Projekty o niskiej jakości specyfikacji wymagań wskazują średnie przekroczenie kosztów o 60% i terminów o 200%. Z doświadczeń autora wynika, że analizę taką warto przeprowadzić dla projektów o wartości powyżej 50 tys. zł, ponieważ koszt takiej analizy będzie niższy niż ponoszenie ryzyka w wysokości 60% budżetu.



Rysunek 1. Porównanie kosztów i czasu trwania projektu prowadzonego z zastosowaniem różnych metod

Źródło: opracowanie na podstawie Żeliński (2011).

Opis procesu zbierania wymagań

Przez lata podstawą specyfikowania wymagań była IEEE Recommended Practice for Software Requirements Specification IEEE 830. W miarę rozwoju systemów informatycznych pojawiało się wiele metod opisu w UML 2 (*Unified Modeling Language*). Jedną z najczęściej cytowanych przez firmy informatyczne metod zbierania wymagań jest zaproponowany przez Wiegiersa proces, który opisał po raz pierwszy w 2000 r. w książce *Software Requirements*. Proces zbierania wymagań rozpoczyna się od zebrania wymagań biznesowych – BRS (*Business Requirements Specification*). Są to opisane cele biznesowe firmy, które mają być osiągnięte dzięki tworzonemu oprogramowaniu. Ten dokument powstaje z perspektywy biznesu i zawiera tylko ogólne wymagania, jakie ma spełniać nowy system, w formie cech systemu (Li, Guzman, Tsiamoura, Schneider, Bruegge, 2015, s. 582–591). Zazwyczaj jest tworzony przez klienta przed rozpoczęciem projektu. Wyższe kierownictwo określa cele strategiczne i cele wdrożenia systemu. W dużych projektach Wiegiers (2003) proponuje tworzenie dokumentu wizji na podstawie dokumentu BRS. Taki dokument pełni wówczas rolę dokumentu inicjującego projekt.

Na podstawie BRS lub dokumentu wizji projektu następuje przejście do fazy zbierania wymagań użytkowników. Wymagania użytkowe wynikają bezpośrednio z BRS. Definiują je kierownicy niższego szczebla oraz szeregowi pracownicy. Wymagania użytkowe opisują, co użytkownicy będą mogli wykonać w systemie. Zbieranie wymagań odbywa się zazwyczaj poprzez serie warsztatów, na których przechodzi się przez kolejne funkcjonalności systemu wpisane w analizowany proces biznesowy. Efektem prac w tym etapie jest finalny dokument o nazwie Specyfikacja Wymagań (SRS – *Software Requirements Specification*). Poniżej opisano główne punkty szablonu SRS.

Definicja wymagań biznesowych

Sekcja dokumentacji wymagań służy opisaniu kontekstu biznesowego projektu. Zawiera cele biznesowe oraz ogólny opis cech systemu, a także ograniczenia projektowe i inne czynniki, których uwzględnienie jest niezbędne do utworzenia prawidłowej i wyczerpującej specyfikacji wymagań na oprogramowanie (Dąbrowski, Stasiak, Wolski, 2009, s. 68–69).

Cel budowy systemu

Celem wytwórców oprogramowania powinien być system, który zaspokoi rzeczywiste potrzeby udziałowców i rozwiąże napotykaną przez nich problemy. W tym punkcie można wymienić kilka celów, które należy uszeregować w hierarchii ważności. Niewątpliwie najważniejszym kryterium oceny sukcesu projektu jest osiągnięcie zakładanego celu zgodnie z zaplanowanym harmonogramem i budżetem. Wartość, jaką generuje produkt wytworzony przez projekt dla klienta, jest również ważnym czynnikiem oceny sukcesu projektu w ujęciu kompleksowym (Young, 2006, s. 14).

Analiza kontekstu użytkowania

Proces definicji wymagań należy rozpocząć od określenia interesariuszy. Można wyróżnić dwie grupy interesariuszy: sponsorów oraz użytkowników. Klientem jest osoba, która finansuje projekt, decyduje o wymaganiach, wpływa na wyniki działania systemu, a jej celem jest wytworzenie systemu. Użytkownikiem natomiast jest osoba kontaktująca się z systemem lub pracująca na nim. Często klient i użytkownik to nie są te same postacie (IEEE Std 830-1998, s. 3).

Analiza kontekstu użytkowania polega na zebraniu szczegółowych informacji dotyczących m.in.:

- przyczyny podjęcia prac nad systemem,
- zakresu projektu,
- tego, kim są docelowi użytkownicy planowanego systemu,
- tego, na czym polegają ich zadania,
- tego, czy i dlaczego będą chcieli używać planowanego systemu,
- tego, jakie są ich umiejętności, doświadczenie i nastawienie,

– tego, jakie ograniczenia techniczne, organizacyjne, psychologiczne i środowiskowe mogą przeszkodzić w udanym wdrożeniu systemu.

Analiza kontekstu użytkownika ma również za zadanie zidentyfikować czynniki ryzyka dla projektu, wynikające z przyczyn organizacyjnych, środowiskowych i personalnych.

Cechy systemu

Jest to etap definicji wymagań klientów w formie cech systemu. Stanowią one podstawę w późniejszej ocenie zasadności wykonywania poszczególnych funkcji systemu wskazanych przez użytkowników. To właśnie cechy systemu będą podstawą oceny stopnia realizacji projektu. W ramach analizy cech systemu należy określić:

- problemy, czyli różnice między stanem istniejącym a pożądanym,
- potrzeby, czyli wyrażenie pożądanego stanu przez klientów, który jest remedium na opisane problemy,
- cechy systemu rozumiane jako obserwowalna cecha systemu, przez którą system bezpośrednio zaspokaja jedną lub więcej potrzeb klientów.

Model kontekstowy

Diagram kontekstowy systemu to narzędzie służące do poznania zakresu działania systemu. Prezentuje projektowany system jako jeden proces (Robertson, Robertson, 2000). Pokazuje granice systemu, źródła i odbiorców danych oraz główne wejścia i wyjścia systemu. Tworząc model kontekstowy systemu, należy odpowiedzieć na poniższe pytania:

- Kto używa systemu?
- Kto pielęgnuje system?
- Kto otrzymuje/przetwarza wyniki systemu?
- Z jakimi innymi systemami projektowany system musi się komunikować?

Analiza wymagań użytkowników

Analitik musi zrozumieć, w jakich warunkach funkcjonuje firma. Stąd też dla lepszego zrozumienia działania firmy stworzono modele biznesowe. Wizualizują one strukturę organizacyjną przedsiębiorstwa i jego procesy biznesowe. Pomagają zrozumieć istniejące problemy. Służą do identyfikacji obszarów, które da się poprawić. Pomagają zidentyfikować wymagania na system, których przedsiębiorstwo potrzebuje.

Reguły biznesowe

Reguła biznesowa to stwierdzenie, które definiuje lub ogranicza pewien aspekt działalności biznesowej. Jej celem jest zapewnienie właściwej struktury przedsiębiorstwa oraz kontrolowanie lub wpływanie na jego zachowanie. Metodę zarządzania regułami biznesowymi metodyki tworzenia oraz opisywania modeli decyzyjnych stworzyła von Halle. Można wyróżnić trzy kategorie reguł,

które powinny być rozważone w procesie analizy (Von von Halle, Goldberg, 2010):

1. Terminy – zdefiniowanie słownika używanych w projekcie terminów w formie rzeczowników, np. handlowiec.

2. Fakty – stwierdzenia, które za pomocą przyimków i czasowników łączą terminy w istotne z punktu widzenia biznesu obserwacje, np. handlowiec może być przydzielony do klienta. Istotny w tych opisach jest rodzaj związku, w jakie wchodzi ze sobą terminy (np. może być przydzielony lub musi być przydzielony).

3. Reguły – stwierdzenia, których wynikiem są nowe informacje lub decyzja o podjęciu pewnej akcji w planowanym systemie. Można wyróżnić 4 rodzaje takich reguł: ograniczenia, wyzwalacze, wnioski, obliczenia.

Mapa procesów

Mapa procesu służy pokazywaniu zależności między elementami procesu. Jest często pierwszym krokiem do modelowania procesów. Żeby opracować mapę procesów, trzeba wykonać poniższe kroki:

- należy zastanowić się, jakie procesy występują,
- przemyśleć, jakie są relacje między procesami,
- narysować mapę pokazującą zależności między wyselekcjonowanymi procesami,
- wybrać procesy wymagające zamodelowania.

Mapa procesów nie jest objęta notacją BPMN. Jest to ogólny poglądowy rysunek zależności między procesami i czynnościami niezbędny do zrozumienia organizacji.

Opis procesów biznesowych

Z analizy cech systemu oraz modelu kontekstowego wiadomo, które procesy trzeba przeanalizować. Szczegółowo należy analizować tylko procesy znajdujące się w zakresie projektu.

Modele procesów powinny odwzorowywać wszystkie działania realizowane przez przedsiębiorstwo na różnych szczeblach zarządzania, które są przedmiotem projektu. Stosując metodę *top-down* do modelowania procesów, należy wyjść od zagregowanego modelu przedsiębiorstwa. Następnie przez dekompozycję uzyskujemy bardziej szczegółowe opisy działalności realizowanej na poziomie funkcji/działów/zespołów. Poziom organizacyjny jest reprezentowany w modelu jako tor. Notacją stosowaną do modelowania procesów jest BPMN.

W podejściu do modelowania procesów można przyjąć założenie, że cała działalność przedsiębiorstwa to reagowanie na zdarzenia zewnętrzne i produkowanie rezultatów (Waćkowski, 1998). Proces można zdefiniować jako sekwencję funkcji, które trzeba wykonać, aby optymalnie zareagować na zdarzenie zewnętrzne poprzez wygenerowanie odpowiednich rezultatów.

Lista wymagań funkcjonalnych

Wymagania funkcjonalne definiują, co system musi robić dla użytkowników. Opisują czynności, operacje, usługi wykonywane przez system. Lista wymagań funkcjonalnych jest stworzona na podstawie dotychczas zgromadzonej wiedzy, a w szczególności na podstawie scenariuszy użycia. Po sporządzeniu listy wymagań należy oszacować priorytet każdego z nich, jak również którą cechę systemu realizuje. Wymagania nierealizujące żadnej cechy systemu są życzeniami użytkowników niezblizającymi projektu do realizacji postawionemu przez klienta celu i powinny być w pierwszej kolejności redukowane.

Lista wymagań niefunkcjonalnych

Wymagania niefunkcjonalne definiują, w jakich warunkach system musi realizować wymagania funkcjonalne. Nie mają one związku z cechami systemu, gdyż określają bezwzględnie konieczne wymogi. Wynikają z potrzeb użytkownika, budżetu, strategii firmy, czynników zewnętrznych, współpracy z innymi systemami. Dotyczą systemu jako całości, a nie poszczególnych cech systemu. W trakcie tworzenia listy wymagań niefunkcjonalnych należy je sklasyfikować w niżej wymienione grupy:

- produktowe (użyteczność, wydajność, niezawodność, mobilność),
- organizacyjne (dostawy, standardy, implementacyjne, czynnik ludzki),
- zewnętrzne (współpracy, etyczne, prawne, zabezpieczenia).

Analiza wymagań na system

Analiza wymagań na oprogramowanie składa się z wymagań funkcjonalnych przedstawionych w formie modelu przypadków użycia oraz z wymagań niefunkcjonalnych udokumentowanych w formie tekstowej.

Definicja aktorów

Modelowanie przypadków użycia wymaga od analityka określenia wszystkich aktorów związanych z wykorzystaniem projektowanego systemu, czyli określenia potencjalnych użytkowników systemu. Aby wykorzystać tę technikę w pozyskiwaniu wymagań na system, należy przygotować: charakterystykę postaci odpowiadającej jednemu użytkownikowi systemu, opis stanowiska pracy, cele, potrzeby/problemy, opis typowych 5 zadań wykonywanych przez użytkownika z pomocą systemu.

Definicja przypadków użycia

Diagram przypadków użycia odgrywa najważniejszą rolę w procesie projektowania systemu, ponieważ opisuje wymagania funkcjonalne, jakim system musi sprostać, i otoczenie, w którym się znajduje. Przypadek użycia jest zbiorem scenariuszy związanych ze sobą wspólnym celem użytkownika. Pozwala na zdefiniowanie przyszłego spodziewanego zachowania systemu. Przypadek użycia musi wchodzić w interakcję chociaż z jednym aktorem (Dąbrowski, 2009,

s. 183–199). Przypadek użycia można opisać za pomocą takich cech, jak (Cockburn, 2004): nazwa, cel biznesowy, aktorzy, warunki wstępne, warunki końcowe, przepływ główny, przepływy alternatywne. W celu oceny priorytetu przypadku użycia należy wziąć pod uwagę korzyść biznesową, koszt, ryzyko, priorytet.

Interfejsy zewnętrzne

Rzadko który system może samodzielnie istnieć bez współpracy z innymi systemami informatycznymi. Stąd też istotną kwestią jest przygotowanie dokumentacji interfejsów, która opisuje szczegóły działania interfejsu: opis, format i struktura interfejsu, tabela mapowań, formaty pól, sposób kodowania, wyzwalacze, harmonogram.

Wymagania dotyczące instalacji i środowiska pracy

Analitycy powinni również określić wymagania użytkowników odnośnie do instalacji i dokumentacji użytkownika oraz środowiska pracy systemu. Są to czynniki istotne, które zostaną wzięte pod uwagę w procesie projektowania systemu. To na ich bazie modelowane są elementy systemu, tworzony diagram komponentów oraz diagram rozmieszczenia.

Weryfikacja wymagań

Specyfikacja wymagań na oprogramowanie powinna być zweryfikowana pod kątem spójności zewnętrznej z wymaganiami na system. W celu weryfikacji tworzy się tabelę zawierającą w wierszach kolejno wymienione wymagania funkcjonalne, a w kolumnach przypadki użycia. Na przecięciu wierszy i kolumn oznacza się dla każdego przypadku użycia, które wymagania funkcjonalne realizuje. Przypadku niespełniające żadnego wymagania funkcjonalnego powinny być zgłoszone klientowi do skreślenia z listy lub realizacji w następnych etapach, jeśli zostaną przez sponsora ocenione jako zasadne.

Podsumowanie

Błędne wyobrażenie, jakoby system miał służyć użytkownikom w łatwiejszej i wygodniejszej pracy, prowadzi często do definiowania systemu z punktu widzenia użytkownika końcowego. Właściwe postawienie celów projektu i dążenie do ich realizacji jest słusznym kierunkiem w definiowaniu wymagań.

Prawdopodobieństwo niepowodzenia projektu wdrożenia systemu informatycznego we wstępnej fazie wynosi ok. 50%. Dlatego koszt analizy powinien oscylować w granicach 20% budżetu projektu, a czas jej trwania wynosić ok. 50% czasu całego projektu (Żeliński, 2011). Analizę taką warto przeprowadzić dla projektów o wartości powyżej 50 tys. zł, ponieważ koszt takiej analizy będzie niższy niż koszt ponoszenia ryzyka.

W artykule zdefiniowano optymalną strukturę specyfikacji składającą się z 3 głównych części: definicji wymagań biznesowych, analizy wymagań użytkow-

ników, analizy wymagań na system. Struktura specyfikacji jest wynikiem analizy norm, metod projektowych, jak również doświadczeń autora. Jest to swoisty przewodnik służący praktykom i studentom w nauczaniu projektowania systemów informatycznych. Stanowi również zbiór wytycznych do pisania prac dyplomowych.

Literatura

- Cockburn, A. (2004). *Jak pisać efektywne przypadki użycia*. Warszawa: WNT.
- Dąbrowski, W., Stasiak, A., Wolski, M. (2009). *Modelowanie systemów informatycznych w języku UML 2.1*. Warszawa: PWN.
- IEEE recommended practice for software requirements specifications. IEEE Std 830-1998.
- Li, Y., Guzman, E., Tsiamoura, K., Schneider, F., Bruegge, B. (2015). Automated Requirements Extraction for Scientific Software. *Procedia Computer Science*, 51, 582–591
- Niemiec-Siwek, B., (2011). Pobrane z: <http://www.bcc.com.pl/akademia-lepszego-biznesu/analiza-fit-gap-w-projektach-sap-rollout.htm> (24.09.2014).
- Sobczak, A. (2008). Analiza wybranych metodyk wykorzystywanych do budowy architektury korporacyjnej. *Monografie i Opracowania Szkoły Głównej Handlowej*, 551, 315–323.
- Sommerville, I. (2000). *Inżynieria oprogramowania*. Warszawa: WNT.
- Robertson, J., Robertson, S. (2000). *Pełna analiza systemowa*. Warszawa: WNT.
- Von Halle, B., Goldberg, L. (2010). *The Decision Model: A Business Logic Framework Linking Business and Technology*. New York: CRC Press.
- Waćkowski, K. (1998). *Analiza strategii biznesu*. Katowice: Wyższa Szkoła Zarządzania Marketingowego i Języków Obcych w Katowicach.
- Wiegers, K. (2003). *Software Requirements*. Microsoft Press.
- Young, H. (2006). *Skuteczne zarządzanie projektami*. Gliwice: Helion.
- Żeliński, J., (2011). *Kilka słów o kosztach analizy przedwdrożeniowej i prawie autorskim*. Pobrane z: <http://it-consulting.pl/autoinstalator/wordpress/2011/05/24/kilka-slow-o-kosztach-analizy-predwdrozeniowej/> (20.05.2017).