



Michał Śmiątek, Kamil Rybiński

INŻYNIERIA WYMAGAŃ OPROGRAMOWANIA

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Małgorzata Kulik

Projekt okładki: Studio Gravite/Olsztyn
Obarek, Pokoński, Pazdrijowski, Zaprucki

Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock.

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 230 98 63

e-mail: helion@helion.pl

WWW: <https://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<https://helion.pl/user/opinie/inwyop>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-289-1268-7

Copyright © Michał Śmiałek, Kamil Rybiński 2024

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

| | | |
|-------------|--|-----------|
| ROZDZIAŁ 1. | Organizacja i jakość wymagań | 7 |
| | 1.1. Czym zajmuje się inżynieria wymagań? | 7 |
| | 1.2. Wymagania, ich odbiorcy i reprezentacje | 9 |
| | 1.3. Klasyfikacja wymagań | 11 |
| | 1.4. Jakość specyfikacji wymagań | 17 |
| | Zadania | 22 |
| | Słownik pojęć | 22 |
| | Co trzeba zapamiętać | 25 |
| | Rozwiązania zadań | 26 |
| ROZDZIAŁ 2. | Wymagania w procesie wytwarzania oprogramowania | 29 |
| | 2.1. Czynności i produkty dyscypliny wymagań | 29 |
| | 2.2. Wymagania w cyklu życia oprogramowania | 33 |
| | 2.3. Wymagania jako podstawa zarządzania projektem | 38 |
| | Zadania | 47 |
| | Słownik pojęć | 48 |
| | Co trzeba zapamiętać | 50 |
| | Rozwiązania zadań | 51 |
| ROZDZIAŁ 3. | Wymagania w organizacji biznesowej | 55 |
| | 3.1. Specyfikowanie środowiska systemu | 55 |
| | 3.2. Analiza biznesu w kontekście systemu oprogramowania | 60 |
| | 3.3. Specyfikowanie wizji systemu | 63 |
| | 3.4. Techniki pozyskiwania wymagań | 67 |
| | Zadania | 75 |
| | Słownik pojęć | 75 |
| | Co trzeba zapamiętać | 77 |
| | Rozwiązania zadań | 78 |
| ROZDZIAŁ 4. | Wprowadzenie do wymagań użytkownika | 81 |
| | 4.1. Czym są i skąd się biorą wymagania użytkownika? | 81 |
| | 4.2. Rodzaje wymagań użytkownika | 84 |
| | 4.3. Organizowanie i dokumentowanie wymagań użytkownika | 87 |
| | Zadania | 92 |
| | Słownik pojęć | 93 |
| | Co trzeba zapamiętać | 93 |
| | Rozwiązania zadań | 95 |

| | | |
|-------------|---|------------|
| ROZDZIAŁ 5. | Specyfikowanie wymagań funkcjonalnych | 97 |
| | 5.1. Identyfikacja jednostek funkcjonalnych systemu | 97 |
| | 5.2. Historie użytkownika | 102 |
| | 5.3. Przypadki użycia i aktorzy | 103 |
| | 5.4. Relacje między przypadkami użycia | 108 |
| | 5.5. Opisy historii użytkownika i przypadków użycia | 114 |
| | Zadania | 118 |
| | Słownik pojęć | 118 |
| | Co trzeba zapamiętać | 119 |
| | Rozwiązania zadań | 121 |
| ROZDZIAŁ 6. | Specyfikowanie wymagań jakościowych i ograniczeń | 125 |
| | 6.1. Sposób opisu wymagań jakościowych | 125 |
| | 6.2. Rodzaje wymagań jakościowych | 129 |
| | 6.3. Formułowanie metryk dla wymagań jakościowych | 137 |
| | 6.4. Ograniczenia techniczne i środowiskowe | 142 |
| | 6.5. Wpływ wymagań jakościowych | 144 |
| | Zadania | 148 |
| | Słownik pojęć | 148 |
| | Co trzeba zapamiętać | 150 |
| | Rozwiązania zadań | 152 |
| ROZDZIAŁ 7. | Specyfikowanie słownika dziedziny | 155 |
| | 7.1. Identyfikowanie pojęć w słowniku dziedziny | 155 |
| | 7.2. Formułowanie słownika dziedziny | 161 |
| | Zadania | 169 |
| | Słownik pojęć | 169 |
| | Co trzeba zapamiętać | 170 |
| | Rozwiązania zadań | 171 |
| ROZDZIAŁ 8. | Wprowadzenie do wymagań oprogramowania | 173 |
| | 8.1. Czym są wymagania oprogramowania? | 173 |
| | 8.2. Podstawowe składniki wymagań oprogramowania | 174 |
| | 8.3. Wymagania oprogramowania w cyklu życia | 177 |
| | 8.4. Techniki specyfikowania i dokumentowania wymagań oprogramowania | 180 |
| | Słownik pojęć | 184 |
| | Co trzeba zapamiętać | 184 |
| ROZDZIAŁ 9. | Pisanie scenariuszy dla wymagań funkcjonalnych | 187 |
| | 9.1. Czym są scenariusze przypadków użycia? | 187 |
| | 9.2. Notacja dla scenariuszy | 189 |
| | 9.3. Graficzna reprezentacja scenariuszy | 194 |
| | 9.4. Ponowne wykorzystanie scenariuszy | 196 |
| | Zadania | 199 |
| | Słownik pojęć | 199 |
| | Co trzeba zapamiętać | 200 |
| | Rozwiązania zadań | 201 |

| | |
|--|------------|
| ROZDZIAŁ 10. Tworzenie szczegółowego modelu dziedziny | 205 |
| 10.1. Specyfikowanie szczegółowego modelu dziedziny | 205 |
| 10.2. Specyfikowanie modelu interfejsu użytkownika | 208 |
| 10.3. Wykorzystanie modelu słownikowego | 211 |
| 10.4. Model słownika jako podstawa ponownego wykorzystania | 212 |
| Zadania | 215 |
| Słownik pojęć | 215 |
| Co trzeba zapamiętać | 216 |
| Rozwiązania zadań | 217 |
| ROZDZIAŁ 11. Tworzenie scenopisów i prototypowanie | 221 |
| 11.1. Zasady projektowania interfejsu użytkownika | 222 |
| 11.2. Projektowanie graficznego interfejsu użytkownika | 224 |
| 11.3. Tworzenie scenopisów i prototypów | 229 |
| Zadania | 234 |
| Słownik pojęć | 234 |
| Co trzeba zapamiętać | 235 |
| Rozwiązania zadań | 236 |

ROZDZIAŁ 1.

Organizacja i jakość wymagań

1.1. Czym zajmuje się inżynieria wymagań?

Inżynieria wymagań jest jedną z podstawowych dyscyplin inżynierii oprogramowania. Dlatego też nasze rozważania warto zacząć od określenia, czym jest **inżynieria oprogramowania**. Ogólnie można powiedzieć, że zajmuje się ona inżynierskim podejściem do tworzenia oprogramowania. Inżynieria oprogramowania dąży do ujęcia działań związanych z budową programów komputerowych w ramy typowe dla innych dziedzin inżynierii. W szczególności w inżynierii oprogramowania stosuje się wiedzę naukową, techniczną oraz doświadczenie w celu projektowania, implementacji, walidacji i dokumentowania oprogramowania. Dyscypliny inżynierii oprogramowania związane są z typowymi etapami działań inżynierskich w dowolnej dziedzinie inżynierii.

Czynności procesu wytwarzania oprogramowania można podzielić na dwie grupy: czynności analizy i czynności syntezy. Taki podział jest naturalną dla człowieka metodą radzenia sobie ze skomplikowanymi zadaniami twórczymi: analiza polega na dokładnym zidentyfikowaniu i zrozumieniu problemu, którego rozwiązanie osiąga się poprzez syntezę, czyli realizację i scalanie mniejszych części.

Podążając za naturalną dla człowieka ścieżką analizy i syntezy (patrz rysunek 1.1) od problemu do jego rozwiązania, proces wytwarzania oprogramowania definiuje się jako ciąg czynności podzielonych na dyscypliny, które prowadzą od postawienia problemu do wytworzenia produktu głównego (systemu oprogramowania) i produktów pośrednich. Do czynności analitycznych należą: opisanie środowiska, specyfikowanie wymagań, natomiast do czynności syntetycznych należą: projektowanie, implementacja, wdrożenie. W niniejszym podręczniku koncentrujemy się na przedstawieniu dyscypliny, której podstawowym zadaniem jest zebranie oraz sformułowanie wymagań dla systemu oprogramowania, czyli **dyscypliny inżynierii wymagań oprogramowania** (w skrócie — dyscypliny wymagań).



RYСУNEK 1.1. Analiza i synteza — naturalny sposób rozwiązywania złożonych problemów

Dyscyplina wymagań dostarcza informacji o kształcie budowanego systemu z punktu widzenia klienta (np. użytkowników). Jest to określenie dość ogólne, albowiem na kształt systemu ma wpływ wiele czynników. Główne z nich to:

- środowisko, w którym system będzie funkcjonował;
- zadania, które system będzie realizował;
- sposób, w jaki system będzie funkcjonował.

Główne produkty wykorzystywane i tworzone w ramach dyscypliny wymagań to opis środowiska (np. środowiska biznesowego) i specyfikacja wymagań. Na podstawie specyfikacji wymagań dyscyplina projektowania wytwarza zbiór modeli projektowych, a dyscyplina implementacji realizuje system. Głównym celem dyscypliny wymagań jest zatem określenie zakresu i kształtu przyszłego systemu. Aby ten cel osiągnąć, konieczne jest dobre zrozumienie środowiska (np. otoczenia biznesowego) dla planowanego systemu oprogramowania, a także wyspecyfikowanie wymagań odnośnie do jego funkcjonalności oraz cech jakościowych. Można zatem zaproponować podział dyscypliny inżynierii wymagań na dwie składowe — analizę środowiska (biznesu) oraz analizę wymagań.

W ramach analizy środowiska opisujemy aktualny oraz docelowy stan środowiska (biznesu). Wykonywane jest to przy współpracy z przedstawicielami klienta (np. zamawiającego system). Na tej podstawie definiuje się główne potrzeby użytkowników systemu, które po uwzględnieniu ograniczeń środowiskowych tworzą wymagania zamawiającego. Poprawnie zdefiniowane wymagania zamawiającego zwykle są podstawą do zawarcia kontraktu między zamawiającym i wykonawcą na dostarczenie produktu spełniającego mieszczące się w tym dokumencie wymagania. Kolejną częścią specyfikacji wymagań jest opis wymagań oprogramowania, przedstawiający szczegóły działania systemu (m.in. szczegółowy opis interakcji systemu z użytkownikiem, wygląd okien).

Opis środowiska (biznesu) stanowi pierwszy zasadniczy produkt wykorzystywany w dyscyplinie wymagań. Jest on opisem fragmentu świata, który jest obszarem działań danej organizacji wraz z jej otoczeniem. Formułowanie takiego opisu polega na stworzeniu precyzyjnego i zrozumiałego modelu, który zawiera wszystkie elementy środowiska (biznesowego) oraz zasady oddziaływania ich na siebie. Takie modele składają się zazwyczaj z setek lub nawet tysięcy różnych elementów oddziałujących na siebie w określony, złożony sposób. Podstawowe elementy organizacji biznesowej to współpracownicy, pracownicy, jednostki organizacyjne, produkty, surowce, podzespoły, dokumenty, systemy informatyczne. Jednym z tych elementów jest budowany system informatyczny. Modelowanie może przebiegać na różnym poziomie abstrakcji: możemy modelować całą firmę lub tylko jakiś jej fragment (np. dział czy placówkę). Model może opisywać tylko stan docelowy (po zbudowaniu systemu informatycznego) lub być uzupełniony o opis stanu aktualnego.

Opis środowiska zawiera zwykle słownik pojęć (biznesowych) reprezentujący strukturę środowiska (biznesowego) oraz składniki procesów (biznesowych). Procesy opisywane są przy wykorzystaniu pojęć ze słownika pojęć. Procesy biznesowe są serią powiązanych ze sobą działań, które prowadzą do osiągnięcia celu biznesowego. Dokładna ich analiza dostarcza informacji na temat rzeczywistych potrzeb użytkowników, które są podstawą do zdefiniowania tzw. wymagań funkcjonalnych. Analiza

środowiska, w którym realizowane są procesy biznesowe, dostarcza informacji na temat uwarunkowań środowiskowych, które są podstawą do sformułowania tzw. wymagań jakościowych (inaczej: pozafunkcjonalnych) wobec budowanego systemu.

Kompletna **specyfikacja wymagań** wynika bezpośrednio z opisu biznesu i można ją porównać do piramidy. Na samym szczycie znajduje się wizja systemu, która dostarcza informacji na temat ogólnych cech systemu w ścisłym powiązaniu z potrzebami biznesowymi klienta. Zakres systemu wyznaczają wymagania użytkownika w postaci uporządkowanego zbioru cech i funkcji systemu, które powinny być podstawą do zawarcia kontraktu. Wizja systemu i wymagania użytkownika tworzą wymagania zamawiającego, które są odzwierciedleniem rzeczywistych potrzeb zdefiniowanych przez biznes. Na najniższym poziomie piramidy są wymagania oprogramowania, które opisują szczegóły funkcjonalne komunikacji między użytkownikami i systemem, wszystkie wymieniane między nimi dane oraz planowany wygląd systemu.

Specyfikacja wymagań, obok ogólnych cech systemu, zawiera opis jego dynamiki oraz struktury. Słownik pojęć (biznesowych) jest podstawą do stworzenia słownika dziedziny. Słownik dziedziny jest rozszerzany w trakcie opisywania cech systemu, wymagań funkcjonalnych i jakościowych. Słownik przyjmuje ostateczny kształt na koniec tworzenia specyfikacji wymagań. Słownik zapewnia spójność całej specyfikacji wymagań i jest istotnym jej elementem. Na podstawie takiej spójnej i kompletnej specyfikacji wymagań można zbudować system informatyczny, będący naturalnym fragmentem rzeczywistości przedstawionej w opisie biznesu i realizujący rzeczywiste potrzeby zamawiającego.

Podsumowując, można powiedzieć, że inżynieria wymagań oprogramowania obejmuje działania polegające na zbieraniu, formułowaniu, modyfikacji i aktualizacji specyfikacji wymagań, która jest podstawą efektywnego kosztowo wytworzenia systemu oprogramowania, z wykorzystaniem wiedzy naukowej oraz technicznej. W ściślejszym (systemowym) sensie inżynieria wymagań to używanie obiektów abstrakcyjnych do tworzenia modelu wymagań wytwarzanego systemu oprogramowania. W procesie inżynierii wymagań można wyróżnić sześć kroków: 1) ustalenie wymagań, 2) analizowanie i negocjowanie wymagań, 3) specyfikowanie wymagań, 4) modelowanie obserwowalnego działania systemu, 5) zatwierdzenie wymagań, 6) zarządzanie wymaganiami. Sekwencja tych kroków wykonywana jest najczęściej w powyższej kolejności, przy czym ich uporządkowanie w czasie zależy w dużym stopniu od przyjętego cyklu życia oprogramowania.

1.2. Wymagania, ich odbiorcy i reprezentacje

Podstawowym pojęciem inżynierii wymagań jest oczywiście „wymaganie”. Przed przystąpieniem do szczegółowego omawiania inżynierii wymagań musimy zatem zdefiniować to pojęcie.

Wymaganie to własność produktu końcowego (systemu oprogramowania), którą musi on posiadać, aby spełnić oczekiwania zamawiającego. Własnością tą może być określony sposób funkcjonowania systemu lub zapewnienia niezbędnej jego jakości. Mówi się, że system spełnia wymagania, jeśli zostało potwierdzone, że posiada wszystkie własności określone tymi wymaganiami.

Zgodnie z definicją wymagania odpowiadają na najistotniejsze w całym procesie budowy oprogramowania pytanie: „jaki system mamy zbudować?”. W dobrze zorganizowanym procesie wytwórczym wymagania sterują konstrukcją systemu. Wszystkie działania związane z projektowaniem, implementacją oraz wdrożeniem systemu są podporządkowane spełnieniu wymagań. Ostatecznym potwierdzeniem spełnienia wymagań jest przejście odpowiedniego zestawu testów akceptacyjnych. Dobrze skonstruowany system spełnia wszystkie testy przeprowadzone przy udziale zamawiającego. Kluczem tutaj jest potwierdzenie przez zamawiającego, że wyrażone wymaganiami potrzeby zostały spełnione.

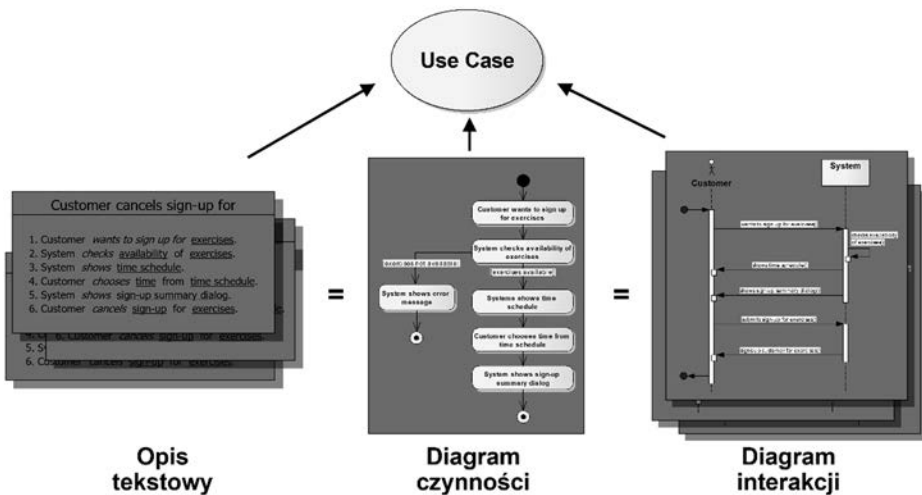
Pisząc wymagania, należy pamiętać o grupach osób, które będą z nich korzystały. Możemy tu wyróżnić osoby zarówno po stronie zamawiającego (np. firmy zamawiającej), jak i po stronie wykonawcy (np. firmy wykonującej system). Warto zauważyć, że dobrą praktyką jest, aby osoby formułujące wymagania nie były zależne ani od zamawiającego, ani od wykonawcy.

- Grupy osób po stronie zamawiającego:
 - sponsorzy — podejmują decyzję o potrzebie zamówienia systemu i uczestniczą w finansowaniu jego budowy;
 - klienci/dostawcy (współpracownicy) zamawiającego — będą musieli współpracować z systemem;
 - przygotowujący przetargi — muszą zorganizować przetarg na budowę systemu zgodnego z wymaganiami;
 - użytkownicy końcowi — będą bezpośrednio korzystać z systemu.
- Grupy osób po stronie wykonawcy:
 - przygotowujący oferty — muszą złożyć ofertę na budowę systemu zgodnego z wymaganiami;
 - wykonawcy systemu (deweloperzy) — muszą zaprojektować i zrealizować zadany system;
 - przygotowujący testy akceptacyjne — muszą napisać testy, które zweryfikują spełnienie wymagań;
 - kierownicy projektów — muszą pokierować realizacją systemu tak, aby był zgodny z wymaganiami.

Należy zauważyć, że zadanie napisania zrozumiałej dla wszystkich tych grup specyfikacji wymagań wydaje się zadaniem karkołomnym. Wynika to zarówno z różnego przygotowania technicznego tych osób, jak i bardzo różnej wiedzy na temat budowy systemów oprogramowania. Osoby bez przygotowania informatycznego często nie rozumieją sformułowań „branżowych”. Z drugiej strony informatycy nie zawsze rozumieją terminologię danej dziedziny problemu, którą posługują się zamawiający (np. terminologię finansową). Konieczne jest zatem zastosowanie notacji, które byłyby zrozumiałe dla określonych uczestników procesu wytwarzania oprogramowania.

Notacje wykorzystywane w dyscyplinie wymagań można podzielić na tekstowe i graficzne (patrz rysunek 1.2). Najbardziej podstawową notacją (jeśli tak to można nazwać) jest wykorzystanie nieograniczonego języka naturalnego (tekstu swobodnego). Taka „notacja” prowadzi jednak zazwyczaj do różnych nieporozumień z uwagi na różne sposoby formułowania takich wymagań i niejednoznaczność języka naturalnego.

Dlatego też powstały różnego rodzaju notacje wykorzystujące ograniczony język naturalny (tekst kontrolowany). Stosowane są np. wzorce wymagań, które wypełnia się według podanych zasad. Przykładem ustrukturalizowanej notacji są historie użytkownika oraz tekstowe scenariusze przypadków użycia (obydwa rodzaje wymagań omawiamy w jednym z następnych rozdziałów). Oprócz notacji tekstowych dużą popularnością cieszą się notacje graficzne, a szczególnie język UML. Dostyc często stosowane są np. diagramy przypadków użycia jako podstawowy sposób określania wymagań funkcjonalnych. Słownik pojęć można sformułować, używając diagramów klas. Diagramy czynności można wykorzystać do graficznego przedstawienia sieci scenariuszy przypadków użycia lub do określenia procesów biznesowych (tutaj również można wykorzystać diagramy w języku BPMN). W kolejnych rozdziałach podręcznika będziemy omawiać te notacje bardziej szczegółowo.

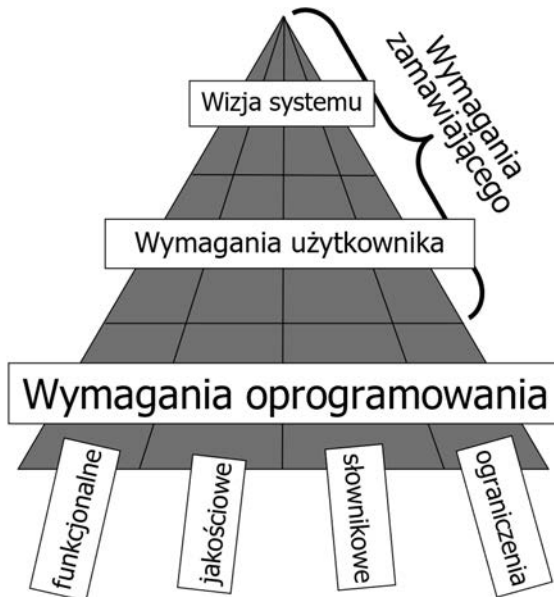


RYSUNEK 1.2. Różne reprezentacje wymagań

1.3. Klasyfikacja wymagań

Jak wskazują badania sondażowe, najczęściej stosowanym narzędziem do specyfikowania wymagań jest zwykły procesor tekstu. Bardzo często formą specyfikacji wymagań jest monolityczny dokument tekstowy, w którym nie są w jednoznaczny sposób wyodrębnione konkretne wymagania. Brakuje określenia zależności między wymaganiami, ich atrybutów czy historii dokonanych zmian. Ogólnie — brakuje systematycznej struktury zgodnej z dobrymi praktykami inżynierii wymagań. Korzystanie z takiego dokumentu podczas analizy i projektowania, a także implementacji i testów, jest niezmiernie trudne. Opisy różnych aspektów systemu są w nim przemieszane, a kolejne modyfikacje powodują dezaktualizację niejawnie związanych ze sobą części specyfikacji. Ponadto niedostateczny formalizm zapisu wymagań sprawia, że cechy systemu są opisane w sposób niejasny, niejednoznaczny i pozbawiony precyzji.

Dla powodzenia projektu konstrukcji oprogramowania kluczem jest zatem przedstawienie zgromadzonych wymagań w systematyczny sposób, zapewniający spełnienie omówionych wyżej kryteriów jakościowych. Aby opis przyszłego systemu był użyteczny dla różnych grup uczestniczących w projekcie, niezbędne jest systematyczne podejście do procesów związanych z tworzeniem specyfikacji wymagań. Procesy te obejmują zbieranie, organizację, dokumentowanie, modyfikację, śledzenie zmian oraz wykorzystanie wymagań. Efektywność tych działań uwarunkowana jest zapewnieniem właściwej formy i struktury specyfikacji wymagań. Dobre praktyki inżynierii wymagań wskazują na potrzebę jednoznacznego podziału specyfikacji zarówno pod kątem poziomu abstrakcji (szczegółowości) wymagań, jak i ich rodzaju. Ponadto bardzo istotne jest precyzyjne określenie powiązań między wymaganiami różnego poziomu i rodzaju. Dobrze zbudowana specyfikacja wymagań powinna mieć strukturę **piramidy**, przedstawionej na rysunku 1.3. Szczyt takiej piramidy stanowi ogólny opis systemu — jego wizja. Elementy wizji systemu stanowią punkt wyjścia do sformułowania wymagań użytkownika określających zakres budowanego systemu. Podstawą piramidy i jednocześnie jej największym fragmentem są wymagania oprogramowania, wyrażające wszystkie szczegółowe potrzeby zamawiającego. Na wszystkich poziomach piramidy formułujemy wymagania czterech rodzajów: wymagania funkcjonalne, jakościowe, słownikowe oraz ograniczenia środowiskowe i techniczne. Poniżej omówimy wszystkie te elementy piramidy wymagań.



RYSUNEK 1.3. Organizacja specyfikacji wymagań

Należy tutaj podkreślić, że specyfikacja wymagań nie musi (a często — nie powinna) być tworzona jako jeden gruby dokument na początku całego procesu wytwarzania oprogramowania. W metodykach iteracyjnych (zarówno zwinnych, jak i sformalizowanych) wymagania są zbierane i formułowane przez cały okres projektu.

Wymagania mogą podlegać zmianom, zatem specyfikacja wymagań powinna być produktem „żyjącym” przez cały czas projektu. W wielu projektach konieczne jest jednak sformułowanie formalnej umowy między zamawiającym a wykonawcą systemu. Kluczowym elementem takiej umowy jest **specyfikacja wymagań zamawiającego** (np. w zamówieniach publicznych spełniająca rolę specyfikacji istotnych warunków zamówienia). Z punktu widzenia struktury wymagań taka specyfikacja powinna objąć dwie górne warstwy piramidy (patrz rysunek 1.3).

- Wizja systemu pełni rolę określenia ogólnych cech systemu w ścisłym powiązaniu z potrzebami biznesowymi zamawiającego.
- Wymagania użytkownika pełnią rolę specyfikacji potrzebnej do określenia rozmiaru systemu i nakładu pracy potrzebnego na jego zbudowanie.

Zamawiający, w ramach wymagań zamawiającego, przedstawia swoje rzeczywiste potrzeby związane ze wspieraną przez system dziedziną swojej działalności. Jednocześnie określa zakres systemu na tyle jednoznacznie, żeby możliwe było dokonanie wyceny budowanego systemu przez wykonawcę. Zwróćmy uwagę na to, że do określenia zakresu systemu nie są konieczne wszystkie szczegółowe wymagania (np. kolor i położenie elementów ekranowych czy kolejność nawigacji między oknami). Dlatego też umieszczenie szczegółowych wymagań oprogramowania w specyfikacji wymagań zamawiającego należy traktować jako błąd. Takie wymagania są często zmieniane wielokrotnie w trakcie projektu i nakład pracy związany z ich stworzeniem na samym początku projektu będzie zazwyczaj stracony. W dalszej części omawiamy poszczególne składniki piramidy wymagań. Więcej szczegółów zostanie przedstawionych w kolejnych rozdziałach.

Wizja systemu jest formułowana na wysokim poziomie ogólności i nie powinna mieć charakteru wiążącego w kontaktach między zamawiającym a wykonawcą systemu. Celem jej jest sformułowanie intencji przyświecających rozwojowi systemu i wyszczególnienie problemów odkrytych w modelu biznesu zamawiającego. Kompletna wizja systemu powinna zawierać:

- opis problemu biznesowego, który jest podstawą do powstania nowego systemu; oznacza to m.in. identyfikację głównych obszarów biznesu zamawiającego, które powinny uzyskać wsparcie w wyniku zbudowania nowego systemu;
- przedstawienie interesariuszy powstającego systemu, czyli osób bezpośrednio i pośrednio zainteresowanych powstaniem systemu; są to m.in.: kluczowi użytkownicy, ich reprezentanci oraz kadra kierownicza, zamawiający system, osoby finansujące powstanie systemu lub w inny sposób związane z nim w aspekcie ekonomicznym, personel zaangażowany w tworzenie systemu;
- ogólne cechy i funkcje systemu oraz słownik pojęć biznesowych;
- najważniejsze ograniczenia środowiskowe i techniczne przyszłego systemu;
- opcjonalnie — motto projektu, czyli krótki „slogan” oddający główny cel powstania produktu informatycznego.

Wymagania użytkownika definiują system w sposób na tyle szczegółowy, aby potencjalnie stanowić mniej lub bardziej formalną podstawę do kontroli postępów prac, w tym — zawarcia kontraktu na wykonanie systemu. Typowo na wymagania użytkownika składają się wymagania funkcjonalne wyrażone przez uporządkowaną

strukturę przypadków użycia systemu lub historii użytkownika, definicje aktorów (ról użytkowników), słownik pojęć wykorzystanych podczas formułowania wymagań, istotne wymagania jakościowe oraz ograniczenia środowiskowe przyszłego systemu.

Na poziomie wymagań użytkownika tworzonych jest wiele dziesiątków lub nawet tworzone są setki wymagań różnego rodzaju. Dlatego też zazwyczaj konieczne jest wyraźne wydzielenie struktury i podział na pakiety wymagań. Konieczne jest także wyraźne rozróżnienie między wymaganiami funkcjonalnymi i jakościowymi. Wymagania funkcjonalne określamy, stosując notacje charakterystyczne dla wybranych metodyk wytwórczych. Najczęściej stosowanymi notacjami są przypadki użycia systemu (język UML) oraz historie użytkownika. Z uwagi na liczbę wymagań funkcjonalnych konieczne jest zazwyczaj podzielenie ich na pakiety, np. ze względu na użytkowników lub obszary funkcjonalne. Dla wymagań jakościowych musimy na tym poziomie wymagań sformułować precyzyjne metryki. Umożliwi to — po zawarciu kontraktu — odpowiednią kontrolę spełnienia tych wymagań.

W trakcie prac nad systemem wymagania oprogramowania podlegają dokładnej specyfikacji — uszczegółowieniu. Odpowiada za to najbardziej szczegółowy poziom piramidy wymagań, czyli **wymagania oprogramowania**. Precyzja określenia wymagań oprogramowania powinna zapewniać jednoznaczność implementacji systemu przez zespół deweloperski. Na ich podstawie zespół powinien móc zaprojektować i zakodować wszystkie komponenty systemu. Oznacza to, że w ramach specyfikowania wymagań oprogramowania konieczne jest szczegółowe ustalenie z zamawiającym scenariuszy zachowania systemu, wyglądu interfejsu użytkownika oraz struktury przetwarzanych przez system danych wraz z opisem algorytmów przetwarzania danych. Często konieczne jest również doprecyzowanie cech jakościowych systemu, np. poprzez dokładne określenie czasów odpowiedzi dla poszczególnych interakcji użytkownika z systemem.

Wymagania oprogramowania bazują na wymaganiach „odkrytych” i zapisanych jako wymagania zamawiającego. Szczegółowa struktura tych wymagań zależy od rodzaju wymagania bazowego. Na przykład w przypadku wymagań funkcjonalnych najczęstszą notacją są szczegółowe scenariusze interakcji między jednostkami na zewnątrz systemu (aktorami) a systemem. W ten sposób powinny być systematycznie opisane wszystkie realizowane przypadki użycia lub historie użytkownika. Podobnie słownik dziedziny sformułowany na poziomie wymagań użytkownika powinien ulec przekształceniu w model danych, który bardzo szczegółowo określa klasy danych, atrybuty i ich typy, a także rodzaje relacji.

Na wszystkich poziomach piramidy wymagań powinien być wyraźnie zachowany podział między różnymi rodzajami wymagań (patrz rysunek 1.3). Rodzaje wymagań to „kolumny” piramidy, gdyż na kolejnych poziomach stanowią jednolitą całość, opisywaną w sposób coraz bardziej szczegółowy. Najczęściej spotykanym podziałem wymagań jest podział na wymagania funkcjonalne i jakościowe. Czasami jako rodzaj wymagań jakościowych wymienia się ograniczenia środowiskowe i techniczne, ale tutaj traktujemy je jako osobny rodzaj wymagań. Bardzo istotny jest również słownik (wymagania słownikowe), który spaja całość specyfikacji poprzez zastosowanie wspólnej terminologii.

Wymagania funkcjonalne określają sposób zachowania się systemu w interakcji z jednostkami na zewnątrz tego systemu — użytkownikami (ludźmi) oraz systemami zewnętrznymi. W ramach specyfikacji wymagań funkcjonalnych powinniśmy odpowiedzieć na podstawowe pytanie: „co system ma robić?”. W szczególności specyfikacja powinna określać m.in. usługi dostarczane przez system, sposób reakcji na komunikaty dochodzące spoza systemu, sposób zachowania w określonych sytuacjach i przy aktualnym stanie systemu. Zestaw wymagań funkcjonalnych określa zakres funkcjonalności konieczny do zrealizowania przez zespół budujący system oprogramowania. Dobrą praktyką jest również określenie funkcjonalności, które nie będą realizowane lub zostaną zrealizowane w kolejnych projektach.

Podstawowym sposobem wyrażania wymagań funkcjonalnych jest podział całej funkcjonalności systemu na jednostki o dobrze określonym celu dla użytkowników systemu. Takimi jednostkami mogą być przypadki użycia lub historie użytkownika, formułowane zazwyczaj na poziomie wymagań użytkownika. Na poziomie wizji systemu w zupełności wystarczające jest stosowanie prostych zdań oznajmujących bez zachowania konkretnej notacji. Na poziomie wymagań oprogramowania szczególności interakcji systemu możemy opisać scenariuszami oraz scenopisami. Wszystkie te notacje omówione są w kolejnych rozdziałach.

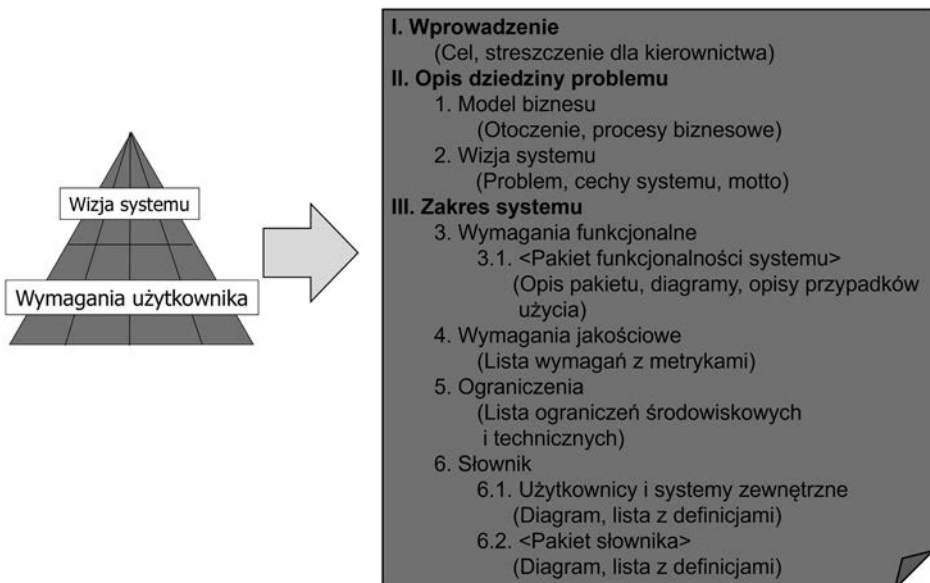
Wymagania jakościowe inaczej nazywamy wymaganiami pozafunkcjonalnymi. Stanowią one charakterystyki systemu określające sposób oceny jego działania. O ile wymagania funkcjonalne definiują to, „co system powinien robić”, o tyle wymagania jakościowe definiują to, „jak system powinien coś robić”. System może realizować wszystkie wymagania funkcjonalne, lecz nadal nie spełniać oczekiwań zamawiającego. Może np. działać zbyt wolno, być zawodny lub nie spełniać kryteriów bezpieczeństwa. Wymagania jakościowe dzielimy na różne kategorie, określające różne charakterystyki jakościowe budowanego systemu.

Bardzo istotne dla określenia wymagań jakościowych jest przedstawienie sposobu ich weryfikacji (testowania). Musimy zapewnić mierzalność wymagań jakościowych, czyli zdefiniować sposób oceny spełnienia wymagań zgodnie z możliwie najbardziej obiektywnymi kryteriami. W tym celu konieczne jest zdefiniowanie metryk wymagań, które opiszą sposób pomiaru oraz oczekiwane wyniki. Warto też zwrócić uwagę na to, że wymagania jakościowe „przecinają” wymagania funkcjonalne. Jedno wymaganie jakościowe może mieć wpływ na sposób (i jednocześnie — koszt) realizacji wielu wymagań funkcjonalnych.

Ograniczenia środowiskowe i techniczne określają warunki narzucone budowanemu systemowi przez jego otoczenie oraz konieczne do zastosowania technologii. Ograniczenia nie są zatem wymaganiami w sensie potrzeb zamawiającego. Stanowią one wymagania w sensie narzuconych przez zamawiającego warunków, w jakich będzie tworzony i uruchamiany system. Te warunki mogą dotyczyć środowiska, w jakim będzie pracował system (np. wyposażenia stanowisk pracy, warunków pogodowych), technologii wymaganych podczas budowy systemu (np. posiadanych licencji na system bazy danych) lub jego działania (np. posiadanego przez zamawiającego sprzętu).

Wymagania słownikowe definiują zakres pojęć i danych, które są używane w ramach dziedziny problemu obejmującej budowany system. Słownik powinien zawierać na bieżąco aktualizowane definicje wszystkich pojęć używanych w ramach wymagań funkcjonalnych, jakościowych oraz ograniczeń. Definicje te wykonywane są na różnym poziomie szczegółowości. Na poziomie wizji często wystarczy krótkie 1 – 2-zdaniowe wyjaśnienie znaczenia danego pojęcia. Na poziomie wymagań użytkownika konieczne jest określenie co najmniej najważniejszych atrybutów pojęć oraz wyspecyfikowanie relacji między pojęciami. Wymagania oprogramowania wprowadzają konieczność zdefiniowania wszystkich szczegółów dotyczących struktur danych, czyli stworzenia szczegółowego modelu danych. Do formułowania wymagań słownikowych możemy w najprostszym przypadku użyć formuły słownika językowego — listy pojęć z ich definicjami. Często jednak o wiele korzystniejsze jest zastosowanie notacji graficznej, np. diagramów klas języka UML. Takie notacje zapewniają większą precyzję i lepsze zrozumienie przedstawianej dziedziny problemu.

Zbiór wszystkich wymagań stanowi specyfikację wymagań, która może być wyrażona w sposób mniej lub bardziej sformalizowany. Często wymagane jest stworzenie formalnego dokumentu, który będzie stanowił załącznik do umowy między zamawiającym a wykonawcą. Przykładową strukturę takiego dokumentu **specyfikacji wymagań zamawiającego** przedstawia rysunek 1.4. Struktura ta jest zgodna ze strukturą dwóch górnych warstw piramidy. Jeśli jest to wymagane, dokument rozpoczyna się od wprowadzenia, które pokrótce opisuje motywację powstania, cel oraz strukturę. Kolejna część zawiera opis dziedziny problemu, na który składają się model biznesu oraz wizja systemu. Ostatnią częścią jest zakres systemu, który dzielimy w sposób wyraźny na cztery omawiane wyżej rodzaje wymagań. Z uwagi na liczbę wymagań może być konieczny ich podział na pakiety. Dotyczy to szczególnie wymagań funkcjonalnych i słownikowych.



RYСУNEK 1.4. Przykładowa struktura dokumentu specyfikacji wymagań zamawiającego

Warunki projektu mogą również narzucać tworzenie dokumentów wymagań już po zawarciu kontraktu. Struktura takich dokumentów może być rozwinięciem struktury specyfikacji wymagań zamawiającego, przedstawionej na rysunku 1.4. Jak wiemy, wymagania zamawiającego są uszczegóławiane w ramach specyfikowania wymagań oprogramowania. Dlatego też dokument **specyfikacji wymagań oprogramowania** może zawierać uszczegółowienie elementów opisujących zakres systemu. W rezultacie rozdział 3. naszej przykładowej struktury specyfikacji wymagań będzie rozszerzony o szczegółowe scenariusze przypadków użycia (lub historii użytkownika). W rozdziale 6. możemy dodać nowy pakiet, przedstawiający słownik elementów interfejsu użytkownika. Rozdziały 4. i 5. mogą być uzupełnione o kolejne wymagania jakościowe i ograniczenia, które np. wynikają z analizy scenariuszy i scenopisów.

Na koniec warto jeszcze raz podkreślić, że sposób tworzenia specyfikacji wymagań silnie zależy od warunków w projekcie, a szczególnie — metodyki wytwórczej. Różne organizacje posiadają różne wzorce specyfikacji wymagań. Specyfikacje mogą być tworzone w oparciu o różnego rodzaju narzędzia dedykowane do zarządzania wymaganiami czy też narzędzia modelowania (np. dla języka UML). W projektach prowadzonych metodykami zwinnymi zazwyczaj nie tworzy się dokumentów w formie przedstawionej powyżej. Należy jednak pamiętać, że niezależnie od sposobu prowadzenia projektu pominięcie niektórych wymagań może skutkować istotnymi problemami lub wręcz niepowodzeniami projektu. Dlatego też warto jest traktować wzorce specyfikacji wymagań jako podstawę zapewnienia ich kompletności i spójności.

1.4. Jakość specyfikacji wymagań

Dla powodzenia przedsięwzięcia informatycznego istotne jest nie tylko zgromadzenie wymagań określających rozwijany system, ale także zapewnienie odpowiedniej jakości tychże wymagań. Aby opis przyszłego systemu przyjął satysfakcjonującą i użyteczną formę, niezbędne jest systematyczne podejście do procesów związanych z zadaniami zarządzania specyfikacją wymagań. Zadania te obejmują zbieranie, organizację, dokumentowanie, modyfikacje, śledzenie zmian oraz wykorzystanie wymagań.

Czynności zarządzania wymaganiami nie mogą być prowadzone efektywnie, gdy specyfikacja wymagań — istotny produkt powstający w procesie wytwarzania oprogramowania — jest tworzona i utrzymywana w nieodpowiedni sposób. Często spotykaną formą specyfikacji wymagań jest monolityczny dokument tekstowy, w którym ciężko wyodrębnić nawet konkretne wymagania, nie wspominając już o zależnościach między nimi, ich atrybutach czy śladach wykonanych zmian. Korzystanie z takiego dokumentu, niezależnie, czy to podczas analizy, implementacji, czy testów, jest niezmiernie trudne — opisy różnych aspektów systemu są w nim przemieszane, kolejne modyfikacje powodują dezaktualizację niejawnie związanych ze sobą części specyfikacji, a niedostateczny formalizm zapisu wymagań sprawia, że charakterystyka systemu jest tworzona w sposób niejasny, niejednoznaczny i pozbawiony precyzji.

Należy podkreślić, że jakość wymagań jest kluczowym elementem powodzenia każdego projektu konstrukcji systemu oprogramowania. Źle sformułowane wymagania są bardzo częstą przyczyną niepowodzeń. Ogólnie można powiedzieć, że dobre sformułowanie wymagań polega na dobrym odzwierciedleniu rzeczywistych potrzeb zamawiającego. Dobrej jakości wymagania powinny charakteryzować się w szczególności kilkoma podstawowymi cechami (patrz rysunek 1.5).



RYСУNEK 1.5. Cechy dobrej jakości specyfikacji wymagań

Wymagania powinny być **kompletne**. Oznacza to, że powinny obejmować cały zakres potrzeb zamawiającego opisany w niezbędnych szczegółach. Aby zapewnić kompletność wymagań, należy nieustannie zadawać sobie oraz zamawiającemu pytanie, czy czegoś nie pominęliśmy. Można tutaj pomagać sobie stosowaniem odpowiednich technik odkrywania wymagań, jak burze mózgów lub warsztaty interaktywne. Kompletność specyfikacji wymagań oznacza również uwzględnienie wszystkich obszarów wymagań, omówionych w poprzedniej sekcji. Przykładowo, sprawdzając kompletność specyfikacji wymagań powinniśmy sprawdzić m.in.:

- Czy specyfikacja wymagań funkcjonalnych zawiera wszystkie istotne scenariusze dla wszystkich przypadków użycia (przed przystąpieniem do ich implementacji)?
- Czy zostały uwzględnione wszystkie obszary wymagań jakościowych zgodnie z odpowiednią normą ISO/IEC?
- Czy wszystkie pojęcia w modelu słownika dziedziny zawierają atrybuty umożliwiające zaprojektowanie na ich podstawie systemu?
- Czy uwzględniliśmy wszystkie grupy użytkowników systemu oraz współpracujące z systemem systemy zewnętrzne?

Drugą istotną cechą wymagań jest ich **jednoznaczność i poprawność**. Takie wymagania powinny definiować jeden możliwy zakres systemu i nie pozostawiać pola do różnych interpretacji. Aby zapewnić jednoznaczność i poprawność, należy stale upewniać się, że wymagania są przejrzyste dla zamawiającego, ale również — że są zrozumiałe dla deweloperów. Należy stale dążyć do uzyskania informacji zwrotnej, czy dobrze rozumiemy to, co przekazuje nam zamawiający. W cyklu iteracyjnym dobrym testem na przejrzystość wymagań jest zrealizowanie pierwszego przyrostu systemu. Jeśli klient potwierdzi, że spełnia on określony wymaganiami zakres

w stopniu dobrym, to mamy właściwe potwierdzenie jednoznaczności wymagań. Ostateczne potwierdzenie poprawności wymagań następuje po odbiorach końcowych, kiedy klient akceptuje system, przyznając, że dobrze spełnia on jego wymagania. Aby zapewnić jednoznaczność i poprawność wymagań, powinniśmy sobie zadawać m.in. następujące pytania:

- Czy system rzeczywiście powinien się tak zachowywać?
- Czy nazwy przypadków użycia określają istotny i jednoznaczny cel dla ich użytkowników?
- Czy treść scenariuszy jest sformułowana za pomocą prostych, jednoznacznych zdań oznajmujących?
- Czy dane przetwarzane (przechowywane) przez system są dobrze i precyzyjnie określone?
- Czy wiadomo, co (dane, wykresy, grafika) ma być wyświetlane na poszczególnych ekranach?
- Czy wymagania dobrze realizują procesy biznesowe zdefiniowane przez klienta?
- Czy wszystkie pojęcia w słowniku dziedziny są wystarczająco jasno zdefiniowane?

Wymagania powinny być również **spójne**, czyli niesprzeczne. Eliminacja sprzeczności polega na ciągłym wzajemnym porównywaniu wymagań ze sobą, szczególnie wymagań różnych rodzajów. Sprzeczności często prowadzą do problemów podczas realizacji systemu. Mogą one powstać w wyniku tego, że specyfikacja jest tworzona przez zespół i brak w nim wewnętrznego porozumienia (w tym, w zespole reprezentującym zamawiającego). Dobrym sposobem na eliminację sprzeczności jest ciągła aktualizacja wspólnego słownika. Jednoznaczne słownictwo pozwala uniknąć nieporozumień i zapewnić spójność różnych obszarów specyfikacji wymagań. Ważne jest szczególnie zwrócenie uwagi na wymagania jakościowe oraz ograniczenia techniczne. Niektóre ograniczenia (np. konieczność stosowania określonego sprzętu) mogą w ewidentny sposób stać w sprzeczności z wymaganiami jakościowymi (np. wydajnością przetwarzania danych). Spójność wymagań można weryfikować, odpowiadając m.in. na następujące pytania:

- Czy każde pojęcie użyte w specyfikacji wymagań funkcjonalnych i jakościowych jest zdefiniowane w słowniku?
- Czy w słowniku nie ma synonimów (pojęć o takiej samej definicji, ale różnych nazwach) lub homonimów (pojęć o takich samych nazwach, ale różnych definicjach)?
- Czy orzeczenia w zdaniach scenariuszy są powiązane z definicjami pojęć w słowniku?
- Czy alternatywne scenariusze mają poprawne rozgałęzienia (opisane np. diagramami czynności)?

Kolejna cecha dobrej specyfikacji wymagań to **możliwość zarządzania i śledzenia**. Wymagania powinny być podzielone na dobrze określone jednostki, które mogą sterować procesem wytwórczym. Powinny one być na tyle małe, aby można było je przydzielać do wykonania w stosunkowo krótkich odcinkach czasu (np. w ciągu jednej iteracji/w ciągu jednego sprintu). Jednocześnie poszczególne wymagania

powinny mieć przypisane odpowiednie atrybuty pozwalające na efektywne przydzielanie ich do wykonania w trakcie projektu. Takie atrybuty mogą obejmować m.in. unikatowy identyfikator, priorytet dla użytkownika, poziom trudności technicznej, osobę odpowiedzialną oraz wersję. Łatwość zarządzania to również łatwość śledzenia produktów wynikających z wymagań. Wymagania na różnych poziomach powinny z siebie w jednoznaczny sposób wynikać. To samo dotyczy również produktów projektowych oraz kodu. Każda zmiana wymagania powinna w jednoznaczny sposób prowadzić do innych produktów (a ostatecznie — kodu), które wymagają zmiany. Aby zweryfikować możliwość zarządzania wymaganiami, warto zadać sobie m.in. następujące pytania:

- Czy wszystkie wymagania mają nadane unikatowe identyfikatory?
- Czy wszystkie wymagania mają określoną informację o wymaganiach źródłowych (ślad)?
- Czy każdy scenariusz dotyczy konkretnego przypadku użycia bądź historii użytkownika?
- Czy wymagania funkcjonalne są podzielone na jednostki możliwe do zrealizowania w jednej iteracji?
- Czy wymagania posiadają jednolite atrybuty (np. priorytet, trudność, stabilność, osoby odpowiedzialne)?

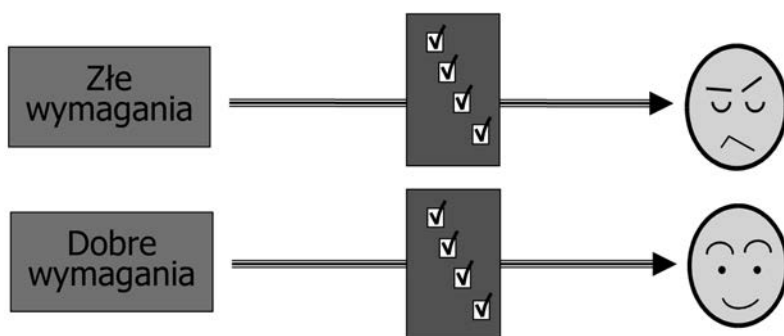
Bardzo istotną, można powiedzieć — kluczową cechą wymagań jest ich **testowalność** (mierzalność). Wymaganie, którego nie jesteśmy w stanie przetestować, jest nic niewarte. Stosunkowo prosto jest sformułować testy dla wymagań funkcjonalnych. Zazwyczaj polegają one na opracowaniu odpowiednich scenariuszy testowych. Sprawa jest trudniejsza w przypadku wymagań jakościowych. Tutaj konieczna jest indywidualizacja podejścia do testowania takich wymagań w zależności od ich szczegółowego rodzaju. Niezbędne jest opracowanie odpowiednich metryk, które pozwolą na jednoznaczne potwierdzenie spełnienia takich wymagań. O metrykach mówimy w sposób bardziej szczegółowy w następnych rozdziałach. Weryfikując możliwość testowania wymagań, warto zadawać sobie m.in. następujące pytania:

- Jak dowiedzimy, że dane wymaganie jest spełnione?
- Czy wymagania funkcjonalne (np. przypadki użycia) posiadają procedury (przypadki) testowe, pozwalające testerom na sformułowanie i przeprowadzenie sprawdzenia poprawności działania systemu?
- Czy słownik dziedziny posiada załączone dane testowe weryfikujące poprawność wprowadzania, przetwarzania i uzyskiwania wyników wyjściowych systemu?
- Czy wymagania jakościowe posiadają miary (np. precyzyjne liczby), które mogą być sprawdzone w sposób obiektywny przez odpowiednie procedury testowe?

Ważne jest również, aby specyfikacja wymagań była **podatna na zmiany**. Należy cały czas pamiętać, że zmiany wymagań są nieodłącznym elementem każdego praktycznie projektu konstrukcji oprogramowania. Należy zatem upewnić się, że specyfikacja umożliwia precyzyjne zidentyfikowanie miejsc, gdzie należy dokonać zmian w momencie zgłoszenia ich przez zamawiającego. Podane w niniejszym podręczniku techniki i notacje zapewniają uzyskanie takiej właśnie cechy specyfikacji wymagań. Przykładowo, przypadki użycia stanowią dobrze określone jednostki wymagań

funkcjonalnych, definiujące jednoznaczny cel dla użytkownika systemu. W momencie, kiedy cele się zmieniają, łatwo jest zatem zidentyfikować te przypadki użycia, które powinny ulec modyfikacji.

Warto na koniec rozważań o jakości wymagań zauważyć, że spełnienie tak określonych wymagań oznacza zadowolonego zamawiającego. Warto też zauważyć, że często system spełnia wszystkie wymagania, a klient (zamawiający) nie jest z niego zadowolony. Jest to najczęściej właśnie wynikiem niekompletnych, mało precyzyjnych, niejednoznacznych czy niemierzalnych wymagań. Taką sytuację ilustruje rysunek 1.6. Wymagania dobrej i złej jakości podlegają zawsze walidacji, czyli sprawdzeniu ich spełnienia. W obydwu przypadkach odpowiednie testy wykazują, że wymagania (zgodnie z ich specyfikacją) zostały spełnione. W przypadku złej jakości wymagań (np. niejednoznacznych czy niekompletnych) po formalnie pomyślnym zakończeniu testów klient nie wyraża zadowolenia i zgłasza wiele uwag, poprawek czy zastrzeżeń do działania systemu.



RYСУNEK 1.6. Konsekwencje źle sformułowanych wymagań

Aby sformułować dobrej jakości wymagania, musimy uzyskać dostęp do odpowiednich źródeł informacji. Można tutaj wyróżnić źródła osobowe (ludzi) oraz źródła nieosobowe (np. dokumenty). W szczególności warto wziąć pod uwagę następujące źródła wymagań:

- istniejące procesy — opisy procesów biznesowych obowiązujące w danej organizacji;
- docelowe procesy — specyfikacja zmodyfikowanych procesów biznesowych, biorących m.in. pod uwagę zastosowanie nowego systemu;
- podręczniki użytkownika — specyfikacja opisująca funkcjonalność istniejących systemów;
- zamawiający — osoby decydujące o dokonaniu zmian w organizacji i budowie nowego (lub rozbudowanego) systemu;
- użytkownicy — osoby, które będą bezpośrednio pracować z systemem.

Ważne jest, aby źródłem wymagań były osoby (zamawiający, użytkownicy) dobrze umocowane i reprezentatywne. W szczególności mogą to być kluczowi użytkownicy (np. kierownicy zespołów, doświadczeni konsultanci wewnętrzni, mentorzy) oraz kluczowi sponsorzy projektu (kierownicy wysokiego szczebla, którym zależy na powodzeniu projektu).

Pisząc wymagania, należy pamiętać o tym, że będą z nich korzystały bardzo różnorodne grupy osób. Możemy tu wyróżnić osoby zarówno po stronie zamawiającego (np. firmy zamawiającej), jak i po stronie wykonawcy (np. firmy wykonującej system). Rolą osób specyfikujących wymagania jest pogodzenie oczekiwań tak różnorodnych grup w stosunku do specyfikacji. Warto zauważyć, że dobrą praktyką byłoby, aby osoby formułujące specyfikację wymagań nie były zależne ani od zamawiającego, ani od wykonawcy. Dzięki temu specyfikacja nie byłaby obciążona określonym skrzywieniem spowodowanym punktem widzenia specyfikującego wymagania. Niestety jest to stosunkowo rzadko spotykana sytuacja. Niezależnie jednak od tego należy brać pod uwagę różnorodność odbiorców.

Zadania

Zadanie 1

Sformułuj kilka wymagań w swobodnym języku naturalnym. Zastanów się, na jakim poziomie można umieścić takie wymagania (wizja, wymaganie użytkownika, wymaganie oprogramowania). Spróbuj zidentyfikować rodzaje tych wymagań (funkcjonalne, jakościowe, słownikowe, ograniczenie).

Zadanie 2

Podaj kilka wymagań, które mogą być ze sobą sprzeczne. Opisz, na czym polega sprzeczność.

Zadanie 3

Zapisz jedno wymaganie funkcjonalne i jedno wymaganie jakościowe. Napisz, w jaki sposób można takie wymagania przetestować.

Słownik pojęć

Dyscyplina wytwarzania oprogramowania

Jeden z obszarów w ramach procesu wytwarzania oprogramowania, obejmujący określony, spójny zestaw czynności i produktów pracy.

Jednoznaczność wymagań

Cecha wymagań oznaczająca, że definiują one tylko jeden możliwy zakres systemu i nie pozostawiają pola do różnych interpretacji.

Kompletność wymagań

Cecha wymagań oznaczająca, że obejmują one cały zakres potrzeb zamawiającego wraz z opisem niezbędnych szczegółów.

Możliwość śledzenia wymagania

Cecha wymagań pozwalająca na jednoznaczne ustalenie, z których wymagań na określonych poziomach piramidy wymagań wynikają wymagania na innych poziomach oraz inne produkty procesu wytwarzania oprogramowania.

Ograniczenia środowiskowe i techniczne

Część specyfikacji wymagań określająca warunki narzucone budowanemu systemowi przez jego otoczenie oraz konieczne do zastosowania technologie.

Opis środowiska (biznesu)

Opis fragmentu świata, który jest obszarem działań danej organizacji (biznesowej) wraz z jej otoczeniem. Zawiera zwykle słownik pojęć (biznesowych) reprezentujący strukturę środowiska (biznesowego) oraz składniki procesów (biznesowych).

Specyfikacja wymagań

Dokument lub model przedstawiający zebrane wymagania dla rozwijanego systemu w uporządkowanej formie.

Spójność wymagań

Cecha wymagań oznaczająca, że nie występują pośród nich wymagania wzajemnie sprzeczne.

Testowalność wymagania

Cecha wymagania oznaczająca, że jest możliwe zmierzenie, czy dane wymaganie jest spełnione. Szczególnie istotna cecha dla wymagań jakościowych.

Wizja systemu

Warstwa specyfikacji wymagań opisująca na wysokim poziomie ogólności intencje przyświecające rozwojowi systemu.

Wymagania funkcjonalne

Wymagania określające sposób zachowania się systemu w interakcji z jednostkami na zewnątrz tego systemu — użytkownikami (ludźmi) oraz systemami zewnętrznymi.

Wymagania jakościowe

Wymagania określające miary oceny działania systemu (jego jakości).

Wymagania oprogramowania

Warstwa specyfikacji wymagań dostarczająca najbardziej szczegółowych opisów działania i miar jakości systemu. Powinna ona zawierać wszystkie potrzebne informacje do zaprojektowania i zakodowania (implementacji) wszystkich komponentów systemu.

Wymagania słownikowe

Część specyfikacji wymagań definiująca pojęcia i dane, które wchodzą w skład dziedziny problemu dotyczącej budowanego systemu.

Wymagania użytkownika

Warstwa specyfikacji wymagań definiująca system na poziomie umożliwiającym określenie zakresu prac niezbędnych do jego realizacji. Wymagania użytkownika mogą stanowić mniej lub bardziej formalną podstawę do kontroli postępów prac.

Wymagania zamawiającego

Część specyfikacji wymagań, która może posłużyć stworzeniu formalnej umowy między zamawiającym a wykonawcą systemu. Składają się na nią wizja systemu i wymagania użytkownika.

Wymaganie

Własność produktu końcowego, którą musi on posiadać, aby spełnić oczekiwania (potrzeby) zamawiającego.

Co trzeba zapamiętać

Czym zajmuje się inżynieria wymagań?

Inżynieria wymagań dostarcza informacji o kształcie budowanego systemu z punktu widzenia klienta (np. użytkowników). W ramach inżynierii wymagań konieczne jest uwzględnienie takich czynników, jak: środowisko, w którym system będzie funkcjonował, zadania, które system będzie realizował, czy też sposób, w jaki system będzie funkcjonował. Główne produkty wykorzystywane i tworzone w ramach dyscypliny wymagań to opis środowiska (np. środowiska biznesowego) i specyfikacja wymagań. Inżynieria wymagań oprogramowania obejmuje działania polegające na zbieraniu, formułowaniu, modyfikacji i aktualizacji specyfikacji wymagań, która jest podstawą efektywnego kosztowo wytworzenia systemu oprogramowania, z wykorzystaniem wiedzy naukowej oraz technicznej. W procesie inżynierii wymagań można wyróżnić sześć kroków: 1) ustalenie wymagań, 2) analizowanie i negocjowanie wymagań, 3) specyfikowanie wymagań, 4) modelowanie obserwowalnego działania systemu, 5) zatwierdzenie wymagań, 6) zarządzanie wymaganiami.

Wymagania, ich odbiorcy i reprezentacje

Wymagania odpowiadają na najistotniejsze w całym procesie budowy oprogramowania pytanie: „jaki system mamy zbudować?”. W dobrze zorganizowanym procesie wytwórczym wymagania sterują konstrukcją systemu. Wszystkie działania związane z projektowaniem, implementacją oraz wdrożeniem systemu są podporządkowane spełnieniu wymagań. Z wymagań korzystają osoby zarówno po stronie zamawiającego (np. firma zamawiająca), jak i po stronie wykonawcy (np. firma wykonująca system). Konieczne jest zatem zastosowanie notacji, które byłyby zrozumiałe dla określonych uczestników procesu wytwarzania oprogramowania. Notacje wykorzystywane w dyscyplinie wymagań można podzielić na tekstowe i graficzne. Najbardziej podstawową notacją jest wykorzystanie swobodnego języka naturalnego. Powstały też różnego rodzaju notacje wykorzystujące ograniczony język naturalny, np. historie użytkownika oraz tekstowe scenariusze przypadków użycia. Notacje graficzne często wykorzystują język UML, np. diagramy przypadków użycia czy diagramy klas.

Klasyfikacja wymagań

Dobrze zbudowana specyfikacja wymagań powinna mieć strukturę piramidy. Szczyt takiej piramidy stanowi ogólny opis systemu — jego wizja. Elementy wizji systemu stanowią punkt wyjścia do sformułowania wymagań użytkownika określających zakres budowanego systemu. Podstawą piramidy i jednocześnie jej największym fragmentem są wymagania oprogramowania, wyrażające wszystkie szczegółowe potrzeby zamawiającego. Na wszystkich poziomach piramidy formułujemy wymagania czterech rodzajów: wymagania funkcjonalne, jakościowe, słownikowe oraz ograniczenia środowiskowe i techniczne. Zbiór wszystkich wymagań stanowi specyfikację

wymagań, która może być wyrażona w sposób mniej lub bardziej sformalizowany. Często wymagane jest stworzenie formalnego dokumentu, który będzie stanowił załącznik do umowy między zamawiającym a wykonawcą.

Jakość specyfikacji wymagań

Jakość wymagań jest kluczowym elementem powodzenia każdego projektu konstrukcji systemu oprogramowania. Ogólnie można powiedzieć, że dobre sformułowanie wymagań polega na dobrym odzwierciedleniu rzeczywistych potrzeb zamawiającego. W szczególności dobrej jakości wymagania powinny charakteryzować się kilkoma podstawowymi cechami. Wymagania powinny być kompletne, tzn. powinny obejmować cały zakres potrzeb zamawiającego opisany w niezbędnych szczegółach. Powinny być również jednoznaczne i poprawne, tzn. powinny definiować jeden możliwy zakres systemu i nie pozostawiać pola do różnych interpretacji. Wymagania powinny być również spójne, czyli niesprzeczne. Eliminacja sprzeczności polega na ciągłym wzajemnym porównywaniu wymagań ze sobą, szczególnie wymagań różnych rodzajów. Kolejną cechą dobrej specyfikacji wymagań jest możliwość zarządzania i śledzenia. Wymagania powinny być podzielone na dobrze określone jednostki, które mogą sterować procesem wytwórczym. Bardzo istotną, można powiedzieć — kluczową cechą wymagań jest ich testowalność (mierzalność). Wymaganie, którego nie jesteśmy w stanie przetestować, jest nic niewarte. Ważne jest również, aby specyfikacja wymagań była podatna na zmiany. Należy cały czas pamiętać, że zmiany wymagań są nieodłącznym elementem każdego praktycznie projektu konstrukcji oprogramowania.

Rozwiązania zadań

Rozwiązanie zadania 1

Rozwiązanie dotyczy dziedziny biblioteki w kontekście planowanej budowy systemu informatycznego.

Wymaganie 1. System powinien umożliwiać zarządzanie zasobami bibliotecznymi poprzez umożliwienie dodawania i usuwania książek oraz prowadzenie stanu magazynowego.

Jest to wymaganie funkcjonalne na poziomie wizji systemu.

Wymaganie 2. System powinien zapewniać bezpieczeństwo danych czytelników i księgozbioru poprzez stosowanie odpowiednich zabezpieczeń informatycznych.

Jest to wymaganie jakościowe na poziomie wizji systemu.

Wymaganie 3. Ekran E-25-1 musi się wyświetlać w maksimum 0,2 sekundy po naciśnięciu przycisku *Pobierz* na formularzu F-17-1.

Jest to wymaganie jakościowe (wydajność) na poziomie wymagań oprogramowania.

Wymaganie 4. System musi umożliwiać pracę na istniejących terminalach bibliotecznych, wyposażonych w system operacyjny Windows 7 oraz posiadających następujące parametry techniczne: procesor Intel Core i3 IV generacji, częstotliwość taktowania, ... itd.

Jest to ograniczenie techniczne (konieczność zastosowania określonego sprzętu i oprogramowania) na poziomie wymagań użytkownika.

Wymaganie 5. Jako użytkownik systemu chcę móc przeglądać księgozbiór. Przeglądanie księgozbioru polega na wyświetleniu listy książek zgodnych z zadanymi kryteriami wyszukiwania. Listę książek można sortować oraz zaznaczać książki, dla których będą możliwe do wykonania działania szczegółowe.

Jest to wymaganie funkcjonalne na poziomie wymagań użytkownika.

Wymaganie 6. Kryteria wyszukiwania składają się z nazwiska autora, nazwy książki oraz roku wydania.

Jest to wymaganie słownikowe (definicja pojęcia) na poziomie wymagań użytkownika.

Rozwiązanie zadania 2

Wymaganie 1. Wyświetlenie każdego z formularzy z grupy F-ABC-1 będzie trwało nie dłużej niż 0,25 sekundy.

Jest to wymaganie jakościowe wydajności.

Wymaganie 2. System musi działać zgodnie z zadanymi wymaganiami wydajności na terminalach zainstalowanych obecnie w czytelnicy głównej, o następującej specyfikacji technicznej: system operacyjny Windows XP, procesor Intel Pentium 5, ... itd.

Jest to ograniczenie techniczne.

Sprzeczność między wymaganiami 1 i 2 polega na tym, że otwarcie dowolnego okienka na terminalach określonych w wymaganium 2 zawsze trwa dłużej niż czas określony w wymaganium 1. Jest to niezależne od wybranej architektury i implementacji systemu.

Rozwiązanie zadania 3

Wymaganie 1. Dodanie nowej książki do księgozbioru. Po wybraniu opcji dodania nowej książki system pokazuje formularz nowej książki. Po wprowadzeniu danych książki są one zapamiętywane przez system.

Jest to wymaganie funkcjonalne. Wymaganie to można przetestować poprzez wykonanie m.in. następującego scenariusza: 1) pokazanie księgozbioru (brak danej książki), 2) dodanie nowej książki do księgozbioru (zakończone sukcesem), 3) pokazanie księgozbioru (nowa książka jest w księgozbiórce).

Wymaganie 2. Sposób działania interfejsu użytkownika będzie akceptowany przez ponad 90% użytkowników.

Jest to wymaganie jakościowe (używalność). Wymaganie to można przetestować poprzez wykonanie następującego eksperymentu: 1) wybranie grupy 50 niezależnych osób, które nie miały do tej pory styczności z systemem, 2) umożliwienie każdej osobie wykonania 10 zadanych przypadków użycia systemu, 3) przeprowadzenie ankiety satysfakcji, 4) obliczenie średniej ocen satysfakcji. Jeśli średnia ocen dla ponad 90% osób będzie wyższa niż 75/100, to należy uznać wymaganie za spełnione.

PROGRAM PARTNERSKI

— GRUPY HELION —

- 
1. ZAREJESTRUJ SIĘ
 2. PREZENTUJ KSIĄŻKI
 3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion

A może zostaniesz inżynierem wymagań?

Inżynieria wymagań jest jedną z podstawowych dyscyplin inżynierii oprogramowania. Cały proces wytwarzania oprogramowania powinien być oparty właśnie na niej. W ramach inżynierii wymagań korzystamy z wiedzy naukowej, technicznej i doświadczenia do zdefiniowania problemu, a także do zarządzania zakresem budowanego systemu oprogramowania. Bez znajomości reguł dotyczących zbierania, specyfikowania i analizy wymagań bardzo trudno jest zbudować system oprogramowania spełniający rzeczywiste potrzeby jego odbiorców.

Ta książka, poświęcona stricte temu zagadnieniu, jest przeznaczona dla wszystkich osób zaangażowanych w formułowanie oraz wykorzystywanie zmieniających się wymagań podczas tworzenia oprogramowania — między innymi analityków systemowych i biznesowych, właścicieli produktu, architektów oprogramowania, kierowników projektów, przedstawicieli zamawiającego, deweloperów. Krok po kroku, od strony teoretycznej i praktycznej, omawia wszystkie kwestie, które inżynier wymagań powinien poznać i opanować, w tym:

- Sposób organizacji i jakość wymagań
- Wymagania w rozmaitych ujęciach (na przykład biznesowym lub z perspektywy użytkownika)
- Wymagania funkcjonalne, jakościowe i ograniczenia
- Wymagania słownikowe (na przykład modele dziedziny i modele danych)
- Wizja systemu, wymagania użytkownika, wymagania oprogramowania
- Pisanie scenariuszy dla wymagań funkcjonalnych
- Tworzenie scenopisów i prototypowanie

| | |
|--|---|
|  | KOD KORZYŚCI <i>Sięgnij po więcej!</i> ▶  |
|  helion.pl | ISBN 978-83-289-1268-7  9 788328 912687 |
|  HELION S.A. ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl | Cena: 69,00 zł |