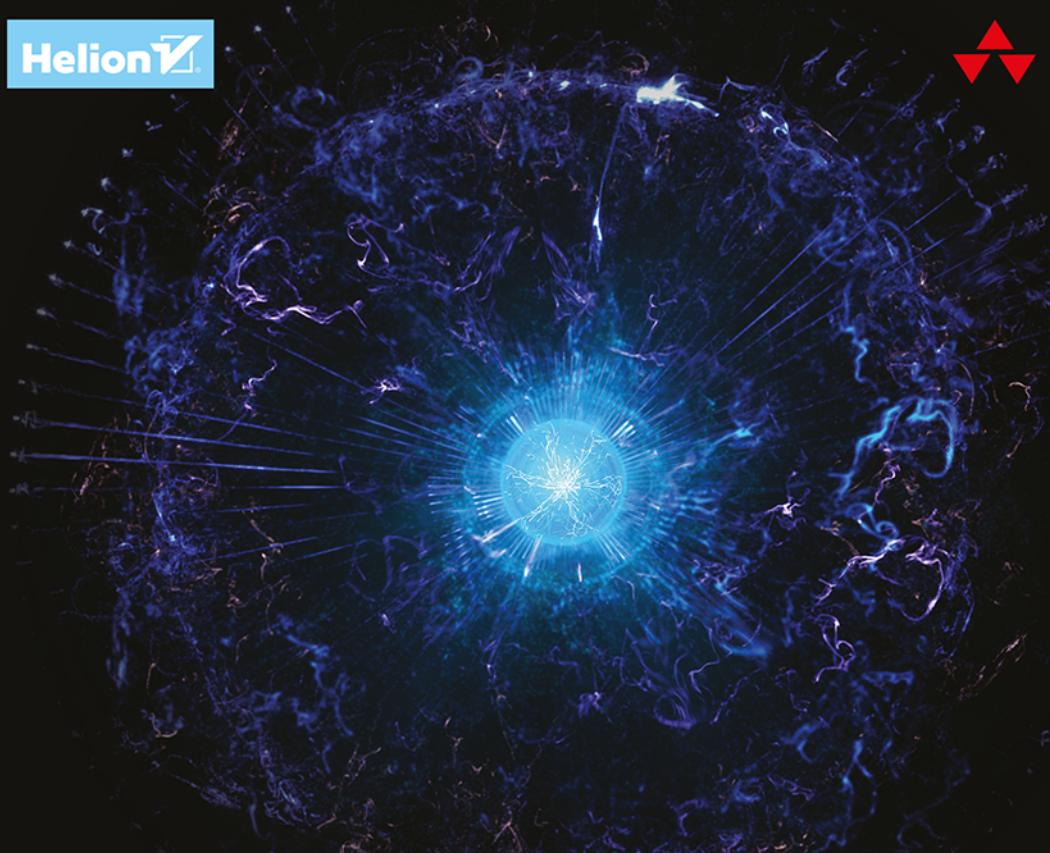




SPECYFIKACJA WYMAGAŃ OPROGRAMOWANIA

Kluczowe praktyki analizy biznesowej



KARL WIEGERS | CANDASE HOKANSON

Przedmowa Joy Beatty

Tytuł oryginału: Software Requirements Essentials: Core Practices for Successful Business Analysis

Tłumaczenie: Łukasz Wójcicki

ISBN: 978-83-289-0436-1

Authorized translation from the English language edition, entitled SOFTWARE REQUIREMENT ESSENTIALS: Core Practices for Successful Business Analysis 1st Edition by KARL WIEGERS; CANDASE HOKANSON published by Pearson Education, Inc, publishing as Addison-Wesley Professional, Copyright © 2023 Karl Wieggers and Seilevel Partners, LP.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

POLISH language edition published by Helion S.A., Copyright © 2024.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

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.

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

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

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

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)

Printed in Poland.

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

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

Spis treści

O autorach	13
Podziękowania	15
Wstęp	17
Rozdział 1. Podstawy wymagań oprogramowania	19
Określenie wymagań	20
Dobre praktyki inżynierii wymagań	23
Kto wykonuje wszystkie te czynności?	25
Kilka powracających tematów	26
Życie i czasy wymagań	27
Rozpoczęcie	28
Rozdział 2. Podstawy	29
Praktyka nr 1. „Zrozumienie problemu przed wypracowaniem rozwiązania”	30
Problemy biznesowe	30
Wywołanie prawdziwych problemów	31
Utrzymywanie uwagi na problemie biznesowym	33
Powiązane praktyki	33
Kolejne kroki	34
Praktyka nr 2. „Określenie celów biznesowych”	34
Wymagania biznesowe	34
Cele biznesowe	36
Miary sukcesu	38
Wizja produktu	39
Powiązane praktyki	40
Kolejne kroki	40

Praktyka nr 3. „Określenie granic rozwiązania”	40
Precyzowanie koncepcji rozwiązania	41
Ustalenie kontekstu	42
Poszerzanie ekosystemu	43
Stosowanie granic rozwiązania	45
Powiązane praktyki	46
Następne kroki	46
Praktyka nr 4. „Identyfikacja i charakterystyka interesariuszy”	47
Poszukiwanie interesariuszy	47
Interesariusze, klienci i klasy użytkowników	50
Charakterystyka interesariuszy	51
Powiązane praktyki	52
Następne kroki	52
Praktyka nr 5. „Identyfikacja osób podejmujących decyzje”	53
Kto podejmuje decyzję?	54
Jak podejmują decyzje?	55
Co dzieje się po podjęciu decyzji?	56
Powiązane praktyki	56
Kolejne kroki	57
Rozdział 3. Pozyskiwanie wymagań	58
Praktyka nr 6. „Zrozumienie tego, co użytkownicy muszą zrobić za pomocą rozwiązania”	59
Skupienie na użytkowaniu	60
Pozyskiwanie wymagań użytkownika	61
Anatomia przypadku użycia	63
Zastosowanie informacji o wymaganiach użytkownika	63
Powiązane praktyki	65
Następne kroki	65
Praktyka nr 7. „Identyfikacja zdarzeń i odpowiedzi”	66
Rodzaje zdarzeń	66
Specyfikowanie wydarzeń	68
Powiązane praktyki	71
Następne kroki	71
Praktyka nr 8. „Ocena koncepcji danych i relacji”	71
Zrozumienie obiektów danych i ich relacji	72
Udoskonalanie zrozumienia danych	74
Szczegóły danych decydują o sukcesie	76
Znajdź wymagania dotyczące danych tam, gdzie są ukryte	78
Powiązane praktyki	79
Następne kroki	79

Praktyka nr 9. „Pobieranie i ocena atrybutów jakościowych”	79
Gromadzenie atrybutów jakości	80
Konsekwencje atrybutów jakości	81
Kompromisy jakościowe	82
Określanie cech jakościowych	82
Powiązane praktyki	84
Następne kroki	84
Rozdział 4. Analiza wymagań	85
Praktyka nr 10. „Analiza wymagań i zestawów wymagań”	86
Analiza indywidualnych wymagań	86
Analiza zestawów wymagań	90
Powiązane praktyki	92
Następne kroki	92
Praktyka nr 11. „Tworzenie modeli wymagań”	93
Wybór odpowiednich modeli	94
Używanie modeli dla lepszego zrozumienia	96
Modelowanie iteracyjne	98
Powiązane praktyki	99
Następne kroki	99
Praktyka nr 12. „Tworzenie i ocena prototypów”	100
Powody tworzenia prototypów	100
Jak tworzyć prototypy	102
Przyszłość prototypu	104
Powiązane praktyki	104
Następne kroki	105
Praktyka nr 13. „Priorytetyzacja wymagań”	105
Wyzwanie związane z priorytetyzacją	105
Czynniki wpływające na priorytet	106
Techniki priorytetyzacji	107
Porównanie parami dla priorytetyzacji cech jakościowych	108
Metody analitycznej priorytetyzacji	109
Powiązane praktyki	111
Kolejne kroki	111
Rozdział 5. Specyfikacja wymagań	112
Praktyka nr 14. „Zapisywanie wymagań w spójny sposób”	113
Niektóre powszechne wzorce dotyczące wymagań	114
Poziomy abstrakcji	115
Atrybuty wymagań	117
Wymagania niefunkcjonalne	118
Powiązane praktyki	119
Następne kroki	119

Praktyka nr 15. „Organizowanie wymagań w strukturalny sposób”	119
Szablony wymagań	120
Specyfikacja wymagań oprogramowania	121
Narzędzia służące do zarządzania wymaganiami	123
Powiązane praktyki	124
Następne kroki	124
Praktyka nr 16. „Identyfikowanie i dokumentowanie zasad biznesowych”	124
Definicja reguł biznesowych	125
Odkrywanie reguł biznesowych	126
Dokumentowanie reguł biznesowych	127
Zastosowanie reguł biznesowych	128
Powiązane praktyki	129
Następne kroki	129
Praktyka nr 17. „Tworzenie słownika pojęć”	130
Synchronizacja komunikacji	130
Powiązane praktyki	132
Następne kroki	132
Rozdział 6. Walidacja wymagań	133
Praktyka nr 18. „Przegląd i testowanie wymagań”	134
Przeglądy wymagań	134
Testowanie wymagań	135
Kryteria akceptacji	136
Testowanie modeli analizy	138
Skuteczne testowanie wymagań	139
Przenoszenie jakości na pierwszy plan	140
Powiązane praktyki	141
Następne kroki	141
Rozdział 7. Zarządzanie wymaganiami	142
Praktyka nr 19. „Ustanawianie baz wymagań i zarządzanie nimi”	143
Określenie bazy wymagań	143
Dwie strategie ustalania bazy	144
Identyfikowanie wymagań uwzględnianych w bazie	145
Uzgodnienie bazy	145
Zarządzanie wieloma bazami i zmianami w nich	147
Powiązane praktyki	148
Następne kroki	149
Praktyka nr 20. „Skuteczne zarządzanie zmianami w wymaganiach”	149
Przewidywanie zmian w wymaganiach	150
Definiowanie procesu kontroli zmian	151
Ocenianie wpływu zmiany	152

Po podjęciu decyzji	154
W poszukiwaniu mniejszej ilości zmian	154
Powiązane praktyki	155
Następne kroki	155
Dodatek	157
Podsumowanie praktyk	157
Podstawy	157
Uzyskiwanie wymagań	157
Analiza wymagań	157
Specyfikacja wymagań	158
Walidacja wymagań	158
Zarządzanie wymaganiami	158
Bibliografia	159

Podstawy wymagań oprogramowania

Wiele lat temu zaczynałem (Karl) tworzenie nowego programu, opierając się tylko na początkowym pomysle. Spędzałem czas na kodowaniu, uruchamianiu i naprawianiu. Robiłem bałagan w kodzie źródłowym, próbując osiągnąć odpowiednie rezultaty. Ostatecznie zdałem sobie sprawę, że źródłem mojego problemu był pośpiech w kodowaniu bez wyznaczonego celu — kodowanie jest zabawne! Te frustrujące doświadczenia nauczyły mnie, jak ważne jest przemyślenie pewnych wymagań — celów, zadań, elementów danych i innych — zanim zrobi się cokolwiek jeszcze. Po dostosowaniu mojego procesu, tak aby najpierw zrozumieć moje wymagania, nigdy więcej już nie czułem, że projekt oprogramowania wymyka mi się spod kontroli.

Wszystkie projekty mają wymagania. Niektóre zespoły rozpoczynają od precyzyjnie zdefiniowanych celów biznesowych, inne otrzymują bogaty opis pożądanых możliwości i cech, a jeszcze inne zaczynają jedynie od mglistej koncepcji nowego produktu. Bez względu na punkt wyjścia wszyscy uczestnicy projektu muszą w końcu osiągnąć wspólne zrozumienie tego, co zespół ma dostarczyć.

Niektórzy uczestnicy projektu nie są zbyt zainteresowani wymaganiami. Niektórzy menedżerowie twierdzą, że są zbyt zajęci, aby angażować się w dyskusje dotyczące wymagań. Ale potem, gdy produkt już rozwinie się do momentu, w którym większe zmiany oznaczają kosztowną przeróbkę, formułują oczekiwania. Niektórzy specjaliści uważają, że czas poświęcony na badanie i dokumentowanie wymagań odwraca ich uwagę od prawdziwej pracy nad tworzeniem kodu. Jednak dobre zestawienie wymagań pozwala odpowiedzieć na kilka ważnych i uniwersalnych pytań.

- Dlaczego nad tym pracujemy?
- Kogo próbujemy zadowolić?
- Co próbujemy zbudować?

- Które funkcje implementujemy jako pierwsze? Które jako następne? A których może nigdy nie zaimplementujemy?
- Jak możemy stwierdzić, czy nasze rozwiązanie¹ jest wystarczająco dobre?
- Kiedy wiemy, że jesteśmy gotowi?

Ta książka opisuje dwadzieścia najważniejszych praktyk, które pomagają zespołom programistycznym stworzyć zestaw wymagań stanowiących podstawę kolejnych prac rozwojowych. Te praktyki mają szerokie zastosowanie niezależnie od rodzaju tworzonego przez zespół produktu lub podejścia do rozwoju. Niektóre zespoły programistyczne pracują nie nad oddzielnymi projektami rozwojowymi, ale nad istniejącymi produktami, które wymagają ciągłych modyfikacji i nowej funkcjonalności. Osoby odpowiedzialne za opracowanie wymagań dla zespołów produktowych takich jak te znajdują w tej książce praktyki równie przydatne w swojej pracy.

Terminologia dotycząca wymagań różni się między tradycyjnymi projektami (opartymi na planie lub przewidywalnymi) a projektami zwinnymi (ang. *agile*) (opartymi na zmianach lub adaptacyjnych). Bez względu na używaną terminologię programistom nadal potrzebne są te same informacje, aby poprawnie zbudować odpowiednie rozwiązanie [Wiegers i Beatty, b.d.]. Niektóre zespoły będą realizować pewne praktyki iteracyjnie, dostarczając wartość w małych kawałkach. Inne mogą wykonać większość prac związanych z wymaganiami we wczesnej fazie projektu, ponieważ problem jest dobrze zrozumiany. Start-up, który stara się ocenić dopasowanie swojego produktu do rynku, skupi się na badaniu pomysłów i podejść, a nie na próbie ułożenia szczegółowej specyfikacji. Niezależnie od tego, jak planujesz swoje cykle rozwojowe, dobre wykonanie tych dwudziestu praktyk może stanowić różnicę między dostarczeniem rozwiązania, które zadowala klientów, a stworzeniem takiego, które tego nie robi.

Określenie wymagań

Teraz, gdy kilka razy użyliśmy słowa „wymaganie”, powinniśmy je zdefiniować. Zespół programistyczny musi radzić sobie z różnorodną wiedzą związaną z wymaganiami, a bez wspólnego zrozumienia tych wymagań łatwo będzie o pomyłki.

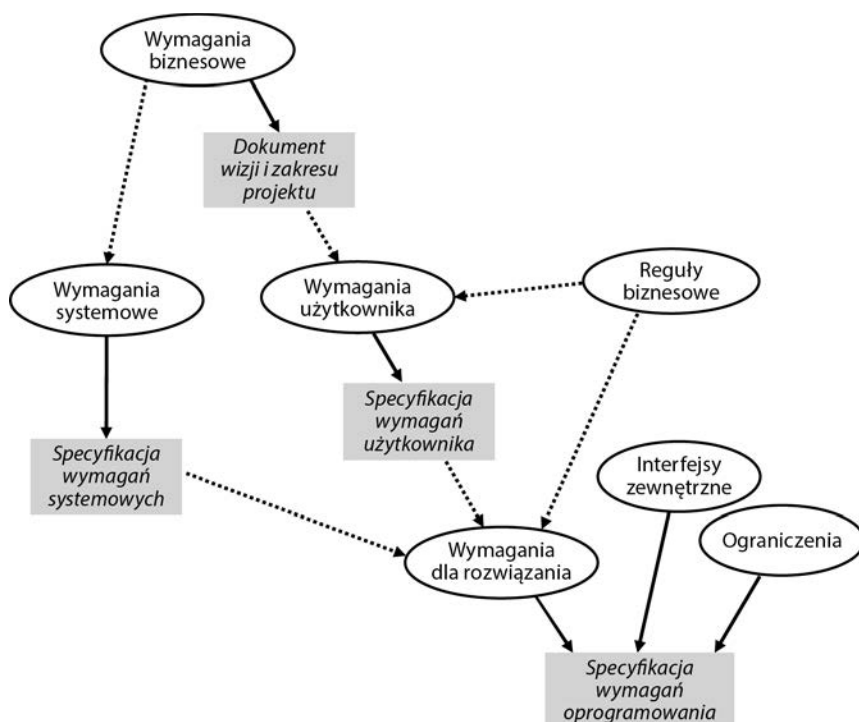
To przydatne, choć niepełne, określenie wymagań jest autorstwa Iana Sommerville’a i Pete’a Sawyera [1997]:

Wymagania to... specyfikacja tego, co powinno zostać zaimplementowane. Są to opisy dotyczące tego, jak system powinien się zachowywać lub jaką ma mieć cechę lub atrybut. Mogą stanowić ograniczenie procesu rozwojowego systemu.

¹ **Projekt** to inicjatywa, która jest uruchamiana w celu stworzenia rozwiązania dla jednego lub więcej problemów biznesowych albo wykorzystania okazji biznesowej. **Rozwiązanie** obejmuje tworzenie lub modyfikację jednego lub więcej produktów, które mogą obejmować systemy oprogramowania, operacje manualne i procesy biznesowe. W tej książce terminy „**produkt**”, „**system**” i „**aplikacja**” są używane zamiennie, aby odnosić się do tego, co Twój zespół buduje.

Ta definicja wskazuje, że wymagania obejmują wiele rodzajów informacji. Jedynym brakującym aspektem w tej definicji jest pojęcie wymagania jako wyrażenia potrzeb klienta, które stanowi rzeczywisty punkt wyjścia do wszystkich dyskusji dotyczących wymagań.

Współcześnie w powszechnym użyciu istnieje kilka schematów i modeli klasyfikacji służących do opisu różnych rodzajów informacji dotyczących wymagań [Robertson i Robertson 2013, Wiegers i Beatty 2013, IIBA 2015]. Zazwyczaj zgadzają się one co do zasad, ale różnią się nieco w terminologii. W tej książce będziemy korzystać z modelu przedstawionego na rysunku 1.1.



Rysunek 1.1. Połączenia między różnymi rodzajami informacji dotyczących wymagań a kontenerami, w których są przechowywane. Linie ciągłe oznaczają „są przechowywane w”. Linie przerywane oznaczają „są źródłem” lub „wpływają”

Ten model przedstawia różne kategorie informacji dotyczących wymagań (owale) oraz kontenery, w których można przechowywać te informacje (prostokąty). W celu uproszczenia w tej książce będziemy odnosić się do tych kontenerów jako dokumentów. Mogą to być arkusze kalkulacyjne, bazy danych, narzędzia do zarządzania wymaganiami, narzędzia do śledzenia problemów, może to być wiki lub ściana pokryta karteczkami samoprzylepnymi — cokolwiek działa dla Twojego zespołu. Sam kontener jest mniej istotny niż informacje, które przechowuje, oraz sposób, w jaki je wybierasz, by móc je rejestrować, organizować i o nich komunikować.

Modele takie jak ten na rysunku 1.1 ilustrują istnienie wielu rodzajów informacji o wymaganiach. W tej książce wykorzystano definicje przedstawione w tabeli 1.1, które są szeroko akceptowane w dziedzinach inżynierii wymagań i analizy biznesowej. Należy zauważyć, że wymagania rozwiązania obejmują wymagania funkcjonalne, niefunkcjonalne i dotyczące

danych [IIBA 2015]. Przykłady tych różnych elementów znajdziesz w późniejszych rozdziałach. W tej książce używa się zbiorczego terminu „wymagania” na odniesienie się do wszystkich tych rodzajów informacji, niezależnie od tego, czy lokalna terminologia skupia się na cechach, przypadkach użycia, historiach użytkowników czy czymkolwiek innym.

Tabela 1.1. Definicje kilku rodzajów informacji o wymaganiach

Rodzaj informacji	Definicja
Wymagania biznesowe	Informacja, która opisuje dlaczego organizacja podjęła się projektu, określa cele biznesowe, definiuje wizję produktu i zawiera inne informacje dotyczące ustalania kierunku. (Zobacz praktykę nr 2, „Określenie celów biznesowych”)
Reguła biznesowa	Dyrektywa, która definiuje lub ogranicza działania w ramach operacji organizacji. Polityka, regulacja, prawo lub standard, które prowadzą do wynikających z nich wymagań rozwiązania, które go egzekwują lub z nim są zgodne. (Zobacz praktykę nr 16, „Identyfikowanie i dokumentowanie zasad biznesowych”)
Ograniczenia	Ograniczenie nałożone na wymagania, projekt lub implementację.
Wymagania danych	Definicja obiektu danych lub elementu, którym system musi manipulować, jego budowy, atrybutów, relacji między obiektami danych oraz ich formatu wejściowego i wyjściowego. (Zobacz praktykę nr 8, „Ocena koncepcji danych i relacji”)
Wymagania dla interfejsów zewnętrznych	Opis połączenia między tworzonym rozwiązaniem a innymi elementami otaczającego go świata, w tym użytkownikami, innymi systemami oprogramowania, urządzeniami sprzętowymi i sieciami.
Wymaganie funkcjonalne	Opis pewnego zachowania, jakie produkt będzie wykazywał w określonych okolicznościach.
Wymaganie niefunkcjonalne	Najczęściej odnosi się do tego, co jest również znane jako wymaganie dotyczące cech jakościowych . Cechy jakościowe opisują różne jakości, obsługę lub parametry wydajności rozwiązania. (Zobacz praktykę nr 9, „Pobieranie i ocena atrybutów jakościowych”)
Wymaganie dotyczące rozwiązania	Opis możliwości lub cechy, które produkt w trakcie tworzenia musi posiadać, aby zaspokoić określone wymagania użytkowników i przyczynić się do osiągnięcia celów biznesowych projektu. Wymagania dotyczące rozwiązania obejmują wymagania funkcjonalne, niefunkcjonalne, dotyczące danych oraz operacje manualne.
Wymagania systemowe	Opis głównej możliwości lub cechy złożonego systemu, który składa się z wielu podsystemów, często obejmujących zarówno elementy sprzętowe, jak i oprogramowanie. Wymagania systemowe stanowią podstawę pochodnych wymagań dotyczących rozwiązania oprogramowania.
Wymagania użytkownika	Opis zadania lub celu, który użytkownik chce osiągnąć za pomocą rozwiązania. Międzynarodowy Instytut Analizy Biznesowej (IIBA) uogólnia tę kategorię jako „wymagania klientów”, ale w rzeczywistości wszystkie wymagania pochodzą od pewnych klientów [IIBA 2015]. Tutaj szczególnie odnosimy się do rzeczy, których użytkownik musi dokonać, i oczekiwań specyficznych dla użytkownika, które rozwiązanie musi spełnić. (Zobacz praktykę nr 6, „Zrozumienie tego, co użytkownicy muszą zrobić za pomocą rozwiązania”)

To, że strzałki na rysunku 1.1, prowadzące od „Wymagań biznesowych” w dół, do „Specyfikacji wymagań oprogramowania”, są w jednej linii, nie jest przypadkiem. Deweloperzy nie implementują bezpośrednio wymagań biznesowych ani wymagań użytkowników. Implementują wymagania funkcjonalne, w tym te wynikające z innych kategorii informacji o wymaganiach. Celem jest zaimplementowanie odpowiedniego zestawu funkcjonalności, które umożliwiają użytkownikom wykonywanie zadań i zaspokajają ich oczekiwania dotyczące jakości, co (mam nadzieję) przyczynia się do osiągnięcia wymagań biznesowych projektu, z uwzględnieniem narzuconych ograniczeń. Ten „odpowiedni zestaw” wymagań funkcjonalnych wynika z dobrze zrozumianych wymagań biznesowych i użytkowników.

Nie każde wymaganie będzie idealnie pasować do jednej z kategorii przedstawionych w tabeli 1.1. Dyskusja na temat dokładnego nazewnictwa konkretnej deklaracji nie jest ważna. Istotne jest, aby zespół rozpoznawał potrzebę, analizował ją, rejestrował w odpowiednim formacie i miejscu oraz budował wszystko, co konieczne do jej zaspokojenia.

Dobre praktyki inżynierii wymagań

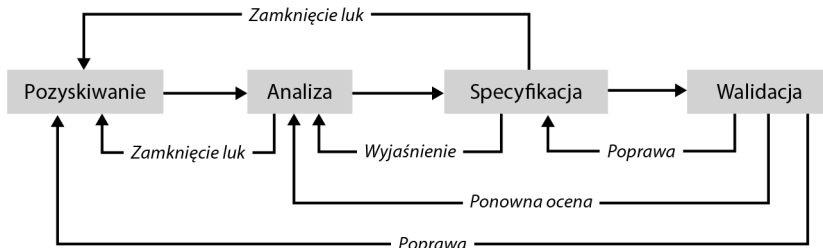
Dziedzina inżynierii wymagań jest podzielona na rozwijanie ich i zarządzanie nimi. Rozwój wymagań obejmuje działania, które zespół wykonuje, aby zidentyfikować, zrozumieć i komunikować wiedzę na ich temat. Zarządzanie wymaganiami dotyczy odpowiedniej opieki po ich uzyskaniu. Działania związane z zarządzaniem wymaganiami obejmują obsługę nieuchronnych zmian, śledzenie wersji wymagań i ich statusu w czasie oraz śledzenie poszczególnych wymagań w odniesieniu do powiązanych wymagań, komponentów projektowych, kodu, testów i innych elementów.

Rozwój wymagań jest dodatkowo podzielony na cztery poddomeny:

Pozyskiwanie	Działania mające na celu zbieranie, odkrywanie wymagań. Czasami jest nazywane zbieraniem wymagań, ale pozyskiwanie jest czymś więcej niż procesem kolekcjonowania.
Analiza	Działania mające na celu ocenę wymagań pod względem ich szczegółowości, wartości, powiązań, wykonalności i innych właściwości, aby osiągnąć dostatecznie precyzyjne ich zrozumienie i wdrożyć wymagania z minimalnym ryzykiem.
Specyfikacja	Działania mające na celu przedstawienie wiedzy na temat wymagań w odpowiednich i trwałych formach, aby można je było przekazać innym osobom.
Walidacja	Działania mające na celu ocenę, w jakim stopniu wymagania zaspokoją potrzebę klienta.

Te cztery zestawy działań nie są po prostu wykonywane w liniowej, jednokrotnie przebiegającej sekwencji. Jak ilustruje rysunek 1.2, są one splecione i powtarzane, aż do momentu, gdy określony zestaw wymagań jest dostatecznie dobrze zrozumiany, tak że zespół deweloperski może zbudować i zweryfikować tę część rozwiązania. Rozwój wymagań jest procesem stopniowym i iteracyjnym, koniecznym, choć może być frustrujący dla jego uczestników.

Eksplorowanie wymagań to działanie, które redukuje niepewność i poprawia efektywność. Proces może wydawać się powolny, ale myślenie o wymaganiach na końcu oszczędza czas.



Rysunek 1.2. Pozyskiwanie, analiza, specyfikacja i walidacja wymagań są wykonywane stopniowo, iteracyjnie i często równolegle

Każda z poddomen inżynierii wymagań obejmuje wiele odrębnych praktyk. O tym właśnie traktuje ta książka. Opisuje ona dwadzieścia podstawowych praktyk, które szczególnie przyczyniają się do sukcesu, praktycznie w każdym projekcie. Niezależnie od tego, czy kierujesz działaniami dotyczącymi wymagań, bierzesz w nich udział, czy polegasz na nich w swojej pracy, będziesz bardziej skuteczny, jeśli zastosujesz te podstawowe praktyki. Kilka z nich odwołuje się do szablonów, narzędzi arkuszy kalkulacyjnych, list kontrolnych i innych narzędzi pomocniczych, które można pobrać ze strony internetowej <http://www.informit.com> powiązanej z tą książką. Praktyki podzieliliśmy według poddomen inżynierii wymagań, cztery dotyczą rozwijania wymagań, a jedna dotyczy zarządzania wymaganiami. Rozdział 3. obejmuje pozyskiwanie wymagań, w rozdziale 4. opisano praktyki analizy, w rozdziale 5. zajęto się specyfikacją wymagań, a w rozdziale 6. omówiono kluczowe praktyki walidacji. Najważniejsze praktyki zarządzania wymaganiami przedstawiono w rozdziale 7.

Każdy opis działania przedstawia wiele praktycznych technik, identyfikuje powiązane praktyki i sugeruje kilka następnych kroków, które pomogą Ci natychmiast wdrożyć je w życie. Opisy praktyk są stosunkowo krótkie, dlatego przedstawiliśmy wiele odniesień do innych źródeł, w których można uzyskać bardziej szczegółowe informacje.

Niektóre praktyki w rozdziale na temat pozyskiwania wymagań dotyczą również powiązanych działań dotyczących analizy i specyfikacji dla tematów takich jak atrybuty jakości i dane. To grupowanie podkreśla wewnętrzne powiązanie tych poddomen wymagań. Nie jest to czyste rozdzielanie.

Może zauważyłeś, że pominęliśmy rozdział 2. Omówiono w nim pięć dodatkowych działań związanych z wymaganiami, które to działania w każdym projekcie powinno się przeprowadzić, aby stworzyć solidne podstawy do osiągnięcia pomyślnego rezultatu. Przeprowadzenie tych działań na wczesnym etapie pozwoli zebrać wszystkich interesariuszy wokół wspólnych celów, zamiast wracać i zajmować się nimi później, gdy zespół napotka problemy.

Ten zestaw praktyk nie stanowi procesu wymagań, który pasuje do każdego przypadku. Podczas tworzenia oprogramowania osoba odpowiedzialna za pracę z wymaganiami powinna współpracować z innymi liderami, aby zdecydować, które podejścia do wymagań będą najbardziej skuteczne. Czynniki, które warto wziąć pod uwagę, są charakter i rozmiar projektu, doświadczenie zespołu w podobnych produktach, dostęp zespołu do interesariuszy, konkretne obszary ryzyka związane z wymaganiami, ograniczenia i kultury organizacyjne

[IIBA 2015]. Wybierz te praktyki, które uważasz za najbardziej wartościowe w pracy, i dostosuj ich opisy z tej książki i innych źródeł, aby jak najlepiej spełniały Twoje konkretne potrzeby.

W Dodatku znajduje się lista wszystkich dwudziestu praktyk, które omawiamy. Nie są to jedyne dostępne techniki wymagań. Istnieje wiele obszernych książek, w których opisano dziesiątki praktyk dotyczących inżynierii wymagań i analizy biznesowej. Oto kilka najbardziej przydatnych źródeł:

- Wiegers Karl, Beatty Joy, *Software Requirements, 3rd Edition*, Microsoft Press, 2013.
- Robertson Suzanne, Robertson James, *Mastering the Requirements Process: Getting Requirements Right, 3rd Edition*, Addison-Wesley, 2013.
- Leffingwell Dean, *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*, Addison-Wesley, 2011.
- Blais Steven P., *Business Analysis: Best Practices for Success*, John Wiley & Sons, Inc., 2012.
- Paul Debra, Cadle James, *Business Analysis, 4th Edition*, BCS, The Chartered Institute for IT, 2020.
- *A Guide to the Business Analysis Body of Knowledge (BABOK Guide), 3rd Edition*, International Institute of Business Analysis, 2015.
- *Business Analysis for Practitioners: A Practice Guide*, Project Management Institute, Inc., 2015.
- *The PMI Guide to Business Analysis*, Project Management Institute, Inc., 2017.

Wspieramy Cię, wskazując książki, które zawierają więcej informacji na temat omawianych tu tematów oraz umożliwią Ci zapoznanie się z innymi praktykami, które mogą być pomocne. Profesjonalista w dziedzinie wymagań musi zgromadzić bogaty zestaw narzędzi i technik oraz zdobyć doświadczenie, aby wiedzieć, które narzędzie jest najlepsze w danej sytuacji.

W niektórych książkach lub w niektórych ramach rozwoju może być zalecane porzucenie pewnych ustalonych praktyk i zastąpienie ich innymi. To złe rady. Powinienesz dodawać nowe praktyki do swojego zestawu narzędziowego, odrzucając starsze tylko wtedy, gdy możesz je zastąpić tymi, których lepsze działanie we wszystkich sytuacjach jest udowodnione. Jeśli coś działa dobrze, to dlaczego to porzucać?

Kto wykonuje wszystkie te czynności?

Osobę odpowiedzialną za opracowanie wymagań w projekcie oprogramowania i zarządzanie nimi historycznie nazywano **analitykiem wymagań**, **analitykiem systemów**, **analitykiem systemów biznesowych** lub po prostu **analitykiem**. W dużych projektach, zwłaszcza tych budujących systemy składające się zarówno z komponentów sprzętowych, jak i oprogramowania, mogą istnieć inżynierowie wymagań, którzy pełnią tę funkcję. Organizacje tworzące komercyjne produkty oprogramowania korzystają z **menedżerów produktu**, którzy wypełniają luki między marketingiem a zespołem deweloperskim. Zespoły Agile często mają **właściciela produktu** (ang. *product owner*), który definiuje prowadzące do rozwiązania wymagania oraz inne elementy pracy, zwane zbiorczo elementami backlogu produktu, i zarządza nimi.

W ostatnich latach termin „analityk biznesowy” w dużej mierze zastąpił te historyczne tytuły zawodowe. W tej książce używany jest termin „**analityk biznesowy**” (BA — ang. *business*

analyst) na określenie osoby odpowiedzialnej za wymagania w zespole deweloperskim. W wielu organizacjach rola BA wykracza poza obszar wymagań, ale skupimy się tutaj na jego działaniach dotyczących jedynie wymagań.

Należy zauważyć, że termin „**analitik biznesowy**” odnosi się do roli, a niekoniecznie do tytułu zawodowego. Nawet jeśli zespół nie ma oficjalnego BA, to nadal ktoś musi analizować, określać, sprawdzać wymagania i zarządzać nimi. Praca ta może być podzielona między kilka osób, być może włączając w to menedżera projektu, lidera zapewnienia jakości i deweloperów. Kiedy członek zespołu zajmuje się zawodowo czymś innym, to jeśli wykonuje tego rodzaju pracę, działa jako BA.

Ponieważ dziedzina wymagań jest krytyczna i złożona, nierealne jest oczekiwanie, że każdy członek zespołu będzie w stanie pełnić rolę BA bez odpowiedniego przygotowania. Kompetentny BA wnosi do procesu określony zestaw wiedzy, doświadczenia, cech osobowości oraz umiejętności, w tym tych wymienionych w tabeli 1.2 [Wiegers i Beatty 2013]. Jeśli pracujesz w tej roli, oceniaj swoje zdolności w każdej z tych kategorii, a następnie pracuj nad poprawą tych umiejętności, które są słabsze od pozostałych.

Tabela 1.2. *Niektóre cenne umiejętności i cechy analityka biznesowego*

Śluchanie	Pisanie
Przeprowadzanie wywiadów i zadawanie pytań	Modelowanie
Ułatwianie (ang. <i>facilitation</i>)	Elastyczność na skali abstrakcji
Komunikacja niewerbalna	Organizowanie informacji i działań
Myślenie analityczne	Obsługa interakcji międzyludzkich
Myślenie systemowe	Przywództwo
Szybkie myślenie	Kreatywność
Obserwacja	Ciekawość

W ostatnich latach wiele organizacji dostrzegło, jaką ogromną wartość mogą wносить analitycy biznesowi i inżynierowie wymagań. Te organizacje opracowały odpowiednie obszary wiedzy i certyfikaty zawodowe, którymi osoby pracujące w tych dziedzinach mogą się posługiwać. Do takich organizacji należą:

- Międzynarodowy Instytut Analizy Biznesowej (IIBA), <http://www.iiba.org>,
- Międzynarodowa Rada Inżynierii Wymagań (IREB), <http://www.ireb.org>,
- Project Management Institute (PMI), <http://www.pmi.org>.

Wiedza zgromadzona przez te organizacje jest bogatym źródłem informacji na temat wielu procesów, technik i narzędzi dotyczących wymagań, które przyczyniają się do uzyskania sukcesu.

Kilka powracających tematów

W tej książce pojawiają się pewne powtarzające się tematy. Miej je na uwadze, wybierając praktyki do stosowania w swoich projektach i dostosowując je do każdej sytuacji.

- **Rozwój wymagań wymaga podejścia inkrementalnego i iteracyjnego.** Bardzo mało prawdopodobne jest, że ktokolwiek wymyśli wszystkie wymagania przed rozpoczęciem prac rozwojowych i że pozostaną one niezmienione. Ludzie uzyskują więcej informacji, mają nowe pomysły, przypominają sobie rzeczy, które przeoczyli, zmieniają zdanie i muszą dostosować się do zmieniających się realiów biznesowych i technicznych.
- Bez względu na to, w jaki sposób decydujesz się przedstawiać wiedzę o wymaganiach, **celem wszystkich działań specyfikacyjnych jest jasna i skuteczna komunikacja.** Artefakty tworzone przez BA mają wielu odbiorców. Ci odbiorcy mogą mieć różne preferencje dotyczące formy prezentacji informacji i stopnia szczegółowości. Twórz dokumentację wymagań, biorąc pod uwagę zróżnicowane grupy odbiorców.
- **Inżynieria wymagań to proces oparty na współpracy.** Wymagania dotyczą wszystkich interesariuszy. Wiele osób może wносить wkład w wymagania, wiele osób pracuje na ich podstawie i równie wiele korzysta z ostatecznego rozwiązania. Zaangażowanie klienta jest potężnym czynnikiem przyczyniającym się do osiągnięcia sukcesu. BA musi współpracować z osobami, które potrafią dokładnie przedstawić potrzeby różnych grup interesariuszy. Większość decyzji dotyczących wymagań wymaga udziału wielu uczestników o różnych, czasem sprzecznych, interesach i priorytetach.
- **Zmiany się zdarzają.** Praca nad rozwiązaniem to pogoń za poruszającym się celem. Potrzeby biznesowe, technologie, rynki, przepisy i użytkownicy się zmieniają. BA musi być na bieżąco z ewoluującymi potrzebami i upewnić się, że zmiany są jasno zrozumiane, zostały zarejestrowane i przekazane tym, których dotyczą.
- Skutecznym sposobem zwiększenia produktywności rozwoju jest minimalizacja konieczności przeróbek, które zespół musi wykonać. Dlatego warto dążyć do **umieszczenia działań związanych z jakością na początku cyklu rozwojowego**, czyli wcześniej niż później. Lepsze wymagania przynoszą korzyści w postaci mniejszej ilości przeróbek w późniejszych fazach rozwoju lub po dostarczeniu.
- **Wykorzystaj podejście oparte na zarządzaniu ryzykiem** do decydowania, jakie praktyki dotyczące wymagań należy zastosować, kiedy je wykonać, kiedy zakończyć oraz jak szczegółowo je opisać. Na przykład ryzyko nieporozumienia i marnowania wysiłku jest większe, gdy rozwój jest zlecony firmom zewnętrznym lub gdy zespoły pracują w odległości, niż gdy uczestnicy pracują blisko siebie. Dlatego wymagania dla takich projektów muszą być bardziej precyzyjne i szczegółowe niż w sytuacji, gdy deweloperzy mogą szybko uzyskać odpowiedzi od osób znajdujących się wokół nich.

Życie i czasy wymagań

Ani rozwój wymagań, ani działania związane z zarządzaniem wymaganiami nie kończą się, z chwilą gdy początkowy zespół projektowy dostarcza rozwiązanie. Trwają one przez cały okres funkcjonowania produktu na rynku, w miarę jego rozwoju — w ramach ciągłego cyklu ulepszeń i konserwacji. Wraz z nadejściem zgłoszeń zmian ktoś musi wydobyć odpowiadające im szczegóły wymagań i ocenić ich wpływ na obecne rozwiązanie. Następnie należy udokumentować nowe lub zmienione wymagania, zweryfikować je, śledzić status ich

implementacji, śledzić zależności od innych elementów systemu itd. BA powinien szukać istniejących elementów związanych z wymaganiami z innych projektów, które mogą być ponownie wykorzystane. Czasami można tworzyć wyniki, które mają potencjał ponownego wykorzystania w innych obszarach organizacji. Słowniki, reguły biznesowe, opisy procesów, katalogi interesariuszy, modele danych, wymagania dotyczące bezpieczeństwa i podobne elementy mogą mieć zastosowanie w wielu sytuacjach. Po zainwestowaniu w tworzenie tych artefaktów organizacja powinna je zorganizować, aby umożliwić ich ponowne wykorzystanie i szukać możliwości dalszego wykorzystania tej inwestycji [Wieggers i Beatty 2013].

Rozpoczęcie

Ta książka zawiera wiele informacji i zaleca się w niej wiele praktyk i technik. Część z nich z pewnością już wykonujesz, inne mogą być dla Ciebie nowe. Mamy dwie rady dotyczące rozpoczęcia praktyk.

1. Nie martw się, jeśli nie wykonujesz jeszcze wszystkich tych działań w swoich projektach.
2. Nie próbuj robić wszystkiego naraz.

Podczas czytania zidentyfikuj te praktyki, które według Ciebie najbardziej przyczynią się do zwiększenia wartości w Twoim projekcie. Szukaj okazji, aby je wypróbować, i sytuacji, w których mogą przynieść lepsze wyniki. Zrozum, że krzywa uczenia spowolni Cię nieco, gdy będziesz próbował zrozumieć, jak nowe metody mogą działać dla Ciebie i Twoich kolegów. Korzystaj z podanych przez nas odwołań, aby dowiedzieć się więcej o tych praktykach, które Cię zainteresują. Z czasem nowe sposoby pracy staną się częścią Twojego zestawu narzędzi BA — i będziesz osiągał lepsze rezultaty.

Niezależnie od tego, czy nazywasz to analizą biznesową, czy inżynierią wymagań, jest to trudna, ale niezwykle istotna funkcja. Podstawowe praktyki opisane w tej książce dają Ci solidne narzędzia do pewnego i skutecznego wykonywania tej kluczowej działalności.

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 

To oczywiste, że przed rozpoczęciem wdrożenia oprogramowania należy określić wymagania biznesowe. Jednak sam proces określania wymagań już taki oczywisty nie jest, a błędy popełnione na tym etapie bywają kosztowne i trudne do naprawienia w późniejszych stadiach rozwoju oprogramowania. Aby tego uniknąć i doprowadzić do wdrożenia projektu przynoszącego maksymalne zyski, konieczne jest potraktowanie etapu określania wymagań jako procesu, który składa się z konkretnych czynności.

Oto zwięzłe i równocześnie praktyczne kompendium, w którym znajdziesz 20 praktyk udanego planowania, analizy, specyfikacji, walidacji i zarządzania wymaganiami. Praktyki te są odpowiednie dla projektów zarządzanych zarówno w tradycyjny, jak i zwinny sposób, niezależnie od branży. Sprawiają, że zespół lepiej zrozumie problemy biznesowe, zaangażuje właściwych uczestników, poprawi komunikację, by wdrożyć najbardziej wartościową funkcjonalność we właściwej kolejności, a później dostosować oprogramowanie do zmian i wzrostu. Dzięki metodom opisanym w tej książce zespół może wypracować wspólne zrozumienie kluczowych koncepcji, terminologii, technik i racjonalnych podstaw, aby efektywniej współpracować nad każdym projektem.

Dr KARL WIEGERS jest głównym konsultantem w firmie Process Impact, wcześniej był badaczem w firmie Kodak. Był też członkiem rady redakcyjnej magazynu „IEEE Software”, a także redaktorem współpracującym w magazynie „Software Development”.

CANDASE HOKANSON jest architektem biznesowym i certyfikowanym praktykiem PMI Agile w firmie ArgonDigital. Od ponad 10 lat zajmuje się analizą i wymaganiami biznesowymi, w tym zakresie prowadzi szkolenia i udziela konsultacji.

Ta jedna książka zwięzłe przedstawia więcej przydatnych informacji niż dziesięć książek razem wziętych!

Mike Cohn, autor książki *User Stories Applied*
i współzałożyciel Scrum Alliance

Helion	KOD KORZYŚCI Sięgnij po więcej!	
 helion.pl	ISBN 978-83-289-0436-1	
 HELION SA ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl		
Cena: 67,00 zł		

 **Pearson**
Addison-Wesley