

ZARZĄDZANIE PROJEKTAMI ZE SCRUMEM

Twórz produkty, które
pokochają klienci

ROMAN PICHLER



Twoja przepustka do nowoczesnego zarządzania projektami!

Tytuł oryginału: Agile Product Management with Scrum: Creating Products that Customers Love

Tłumaczenie: Katarzyna Żarnowska

ISBN: 978-83-246-8526-4

Authorized translation from the English language edition, entitled: AGILE PRODUCT MANAGEMENT WITH SCRUM: CREATING PRODUCTS THAT CUSTOMERS LOVE; ISBN 0321605780; by Roman Pichler; published by Pearson Education, Inc, publishing as Addison Wesley. Copyright © 2010 by Roman Pichler.

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 © 2014.

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

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/zaprs>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

Spis treści

Słowo wstępne — Jeff Sutherland	15
Słowo wstępne — Brett Queener	17
Przedmowa	21
Podziękowania	25
O autorze	27
 Rozdział 1. Rola właściciela produktu	 29
Właściciel produktu	30
Pożądane cechy właściciela produktu	32
Wizjoner i wykonawca	32
Lider i gracz zespołowy	32
Komunikator i negocjator	34
Inspirujący i zaangażowany	34
Dostępny i wykwalifikowany	35
Praca z zespołem	36
Współpraca z mistrzem młyna	38
Praca z klientami, użytkownikami oraz innymi interesariuszami	39
Skalowanie roli właściciela produktu	41
Główny właściciel produktu	41
Hierarchia właścicieli produktu	42
Wybór odpowiednich właścicieli produktu	45

Powszechne błędy	46
Właściciel produktu o zbyt małych uprawnieniach	46
Przepracowany właściciel produktu	47
Właściciel produktu o podzielonych obowiązkach	48
Odległy właściciel produktu	48
Zastępczy właściciel produktu	49
Komitet właścicieli produktu	50
Spostrzeżenia	50
Rozdział 2. Tworzenie wizji produktu	53
Wizja produktu	54
Pożądane cechy wizji	55
Wspólna i jednocząca	56
Obszerna i angażująca	56
Krótka i zwięzła	57
Minimalna wersja produktu nadająca się do wypuszczenia na rynek	58
Prostota	61
Brzytwa Ockhama	62
Mniej znaczy więcej	62
Prosty interfejs użytkownika	63
Potrzeby klientów i atrybuty produktu	64
Narodziny wizji	66
Wykorzystanie projektów osobistych	66
Wykorzystanie metodologii Scrum	67
Techniki tworzenia wizji	68
Prototypy i makiety	68
Persony i scenariusze	69
Opakowania poglądowe i recenzje w prasie branżowej	70
Model Kano	71
Wizja i mapa produktu	72
Minimalna wersja produktu i warianty produktu	73
Powszechne błędy	75
Brak określonej wizji	75
Tworzenie wizji profetycznych	75
Paraliż analityczny	76
Przekonanie o tym, co jest najlepsze dla klientów	76
Duże jest piękne	77
Spostrzeżenia	78

Rozdział 3. Praca z rejestrem produktu	79
Cechy rejestru produktu	80
Wystarczająco szczegółowy	80
Szacunkowy	81
Nowo powstający	81
Zawierający priorytety	81
Porządkowanie rejestru produktu	81
Odkrywanie i opisywanie elementów	83
Odkrywanie elementów	84
Opisywanie elementów	85
Ustalanie struktury rejestru	86
Ustalanie priorytetów rejestru produktu	87
Wartość	88
Wiedza, niepewność i ryzyko	89
Zdatność do wypuszczenia na rynek	90
Zależności	91
Przygotowanie do planowania sprintu	92
Wybór celu sprintu	92
Przygotowanie wystarczającej liczby elementów dokładnie na czas	93
Dekompozycja elementów	95
Zapewnianie klarowności, możliwości testowania wykonalności	97
Dostosowywanie wielkości elementów	98
Punkty	98
Poker planistyczny	99
Postępowanie w przypadku wymagań нефункциональных	102
Opisywanie wymagań нефункциональных	102
Zarządzanie wymaganiami нефункциональными	103
Skalowanie rejestru produktu	104
Wykorzystuj jeden rejestr produktu	104
Działania porządkowe na szeroką skalę	105
Uwzględnienie odrębnych spojrzeń na rejestr	105
Powszechne błędy	106
Ukryte specyfikacje wymagań	106
Lista życzeń do Świętego Mikołaja	107
Określanie wymagań	107

Zaniedbywanie porządkowania	107
Uzupełnianie rejestrów	108
Spostrzeżenia	108
Rozdział 4. Planowanie wydania	111
Czas, koszt i funkcjonalność	112
Zamrożona jakość	115
Wczesne i częste wydania	115
Cykle kwartalne	118
Szybkość	119
Wykres spalania wydania	120
Wykres spalania	120
Belka spalania	122
Plan wydań	124
Prognozowanie szybkości	126
Tworzenie planu wydania	127
Planowanie wydań w dużych projektach	128
Wspólna linia bazowa dla szacunków	129
Planowanie przyszłościowe	129
Systematyzacja	130
Powszechne błędy	131
Brak wykresu spalania lub planu	131
Właściciel produktu na siedzeniu pasażera	132
Rozbudowane wydania	132
Kompromisy związane z jakością	132
Spostrzeżenia	133
Rozdział 5. Współpraca w trakcie spotkań planujących sprint	135
Planowanie sprintu	136
Definicja pojęcia „gotowe”	137
Codzienne zebrania scrumowe	138
Rejestr sprintu i wykres spalania	139
Przegląd sprintu	140
Retrospekcja sprintu	142
Zebrania w trakcie większych projektów	143
Wspólne planowanie sprintu	143
Codzienne zebranie scrumowe dla wszystkich zespołów	144
Wspólne przeglądy sprintu	144
Wspólna retrospekcja sprintu	145

Powszechne błędy	145
Znikający właściciel produktu	146
Pasywny właściciel produktu	146
Zmienne tempo pracy	147
Zasłona dymna	147
Raportowanie elementów wykresu spalania	148
Spostrzeżenia	148
Rozdział 6. Przejście do roli właściciela produktu	151
Bycie doskonałym właścicielem produktu	152
Poznaj siebie	152
Rozwijaj się	152
Zdobądź trenera	154
Upewnij się, że sponsoring pochodzi z właściwego poziomu	154
Jeszcze nie skończyłeś	155
Tworzenie doskonałych właścicieli produktu	155
Doceń wagę roli	156
Wybierz odpowiednich właścicieli produktu	156
Upoważniaj i wspieraj właścicieli produktu	157
Wspieraj wdrażanie roli właściciela produktu	158
Spostrzeżenia	159
Źródła	161
Skorowidz	167

Rozdział 3

Praca z rejestrem produktu

Niewiele jest w Scrumie artefaktów równie popularnych co rejestr produktu. Jest tego powód: rejestr produktu to niezmiernie prosty twór — lista pozostałych do wykonania elementów podzielona na priorytety. Elementy mogą zawierać poznanie potrzeb klientów lub różnych opcji technicznych, opis wymagań funkcjonalnych i niefunkcjonalnych, opis pracy niezbędnej do wypuszczenia produktu na rynek, a także rzeczy takie jak zainstalowanie środowiska lub naprawa defektów. Rejestr produktu zastępuje tradycyjne artefakty dotyczące wymagań, takie jak specyfikacje rynku i produktów. Właściciel produktu jest odpowiedzialny za zarządzanie rejestrem produktu, mistrz młyna, zespół oraz interesariusze również wnoszą swój wkład. Wszyscy razem odkrywają funkcjonalności produktu.

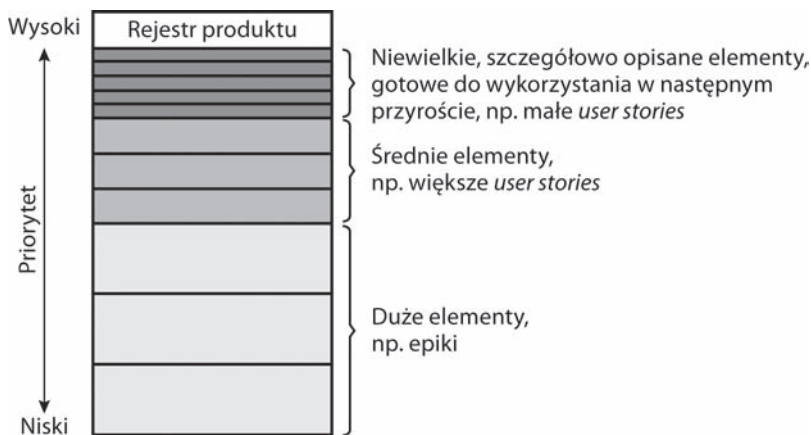
Ten rozdział opisuje rejestr produktu, a także techniki jego efektywnego porządkowania. Dodatkowo przyjrzymy się kilku skomplikowanym zastosowaniom rejestru, w tym obchodzeniu się z wymaganiami niefunkcjonalnymi oraz skalowaniu rejestru produktu dla dużych projektów.

Cechy rejestru produktu

Rejestr produktu ma cztery cechy: jest wystarczająco szczegółowy, szacunkowy, nowo powstający oraz zawiera priorytety (ang. *Detailed appropriately, Estimated, Emergent, Prioritized* — DEEP, czyli dogłębny¹). Przyjrzyj się im dokładniej.

Wystarczająco szczegółowy

Elementy rejestru produktu są wystarczająco szczegółowe, jak pokazano na rysunku 3.1. Elementy mające wyższy priorytet opisane są bardziej szczegółowo niż te o niższym priorytecie. „Im niższy priorytet, tym mniej szczegółów, aż do momentu, w którym ledwo można dany element zrozumieć” — piszą Schwaber i Beedle [2002, 33]. Kierowanie się tymi wytycznymi pozwala na utrzymanie zwięzłości rejestru i upewnienie się, że elementy, nad którymi zespoły będą prawdopodobnie pracować w następnym sprincie, są na to przygotowane. W konsekwencji wymagania są odkrywane, rozkładane na czynniki i dopracowywane w trakcie całego projektu.



Rysunek 3.1. Priorytety nadane elementom rejestru determinują poziom detali

¹ Akronim DEEP zawdzięczam Mike’owi Cohnowi.

Szacunkowy

Elementy rejestru produktów są szacunkowe. Są one ogólnikowe i często wyrażane za pomocą punktów lub dni idealnych. Poznanie rozmiarów elementu pomaga w nadaniu mu odpowiedniego priorytetu oraz zaplanowaniu wydania. Szczegółowe szacunki na poziomie zadań tworzone są w trakcie planowania sprintu; zadania i dotyczące ich szacunki zapisywane są w rejestrze sprintu.

Nowo powstający

Rejestr produktu ma naturalną jakość. Ewoluuje, a jego zawartość często się zmienia. Odkrywane są nowe elementy, które dodaje się do rejestru na podstawie opinii zwrotnych klientów i użytkowników. Istniejące elementy są regularnie modyfikowane, dopracowywane, usuwane lub zmienia się im priorytet.

Zawierający priorytety

Wszystkie elementy rejestru produktów mają priorytety. Te najważniejsze, o najwyższym priorytecie, implementowane są jako pierwsze. Można je znaleźć na szczycie rejestru produktu, jak pokazano na rysunku 3.1. Każdy element, który zostanie wykonany, jest usuwany z rejestru.

Porządkowanie rejestru produktu

Rejestr produktu, o który nie dba się wystarczająco, zarasta jak zaniebany ogród. Rejestr potrzebuje regularnej uwagi i troski, trzeba nim ostrożnie zarządzać, czyli porządkować go. Porządkowanie rejestru produktu to proces ciągły, na który składają się poniższe kroki. Pamiętaj, że nie jest konieczne przeprowadzanie ich w podanej kolejności.

- Nowe elementy są odkrywane i opisywane, a istniejące zmieniane lub usuwane.

- Poszczególnym elementom rejestru produktu nadaje się priorytet. Najważniejsze elementy znajdują się na górze.
- Elementy o wysokim priorytecie przygotowywane są na nadchodzące planowanie sprintu; rozkłada się je na mniejsze elementy i dopracowuje.
- Zespół skaluje elementy rejestru produktu. Dodawanie nowych elementów do rejestru produktu, zmiana istniejących oraz poprawa szacunków sprawiają, że skalowanie staje się niezbędne.

Mimo że to właściciel produktu jest odpowiedzialny za upewnienie się, iż rejestr produktu znajduje się w dobrym stanie, porządkowanie to działanie wspólne. Elementy rejestru są odkrywane, opisywane, rozkładane na czynniki, dopracowywane i oceniane pod względem priorytetu przez całe zespoły scrumowe — proces Scrum zakłada przeznaczenie do 10% czasu zespołu na aktywności związane z porządkowaniem [Schwaber, 2007]; interesariusze są angażowani w miarę potrzeb. Wymagania nie są już przekazywane zespołowi, ich autorami są członkowie zespołu. Właściciel produktu, mistrz młyna oraz zespół angażują się w bezpośrednie rozmowy, zamiast komunikować się za pośrednictwem dokumentów.

Wspólne porządkowanie rejestru produktu jest radosne i efektywne. Pomaga stworzyć dialog pomiędzy członkami zespołu scrumowego oraz pomiędzy zespołem a interesariuszami. Usuwa podział między pracownikami „biznesowymi” a „technicznymi”, eliminując zbędne przekazywanie obowiązków. Zwiększa jasność wymagań, równoważy kolektywną wiedzę i kreatywność zespołu scrumowego, tworząc poczucie więzi i wspólnej własności.

Niektóre zespoły lubią przeprowadzać porządkowanie po codziennym zebraniu scrumowym, innym wystarczą sesje tygodniowe lub dłuższe warsztaty pod koniec każdego sprintu. Porządkowanie odbywa się również w trakcie przeglądu sprintu, kiedy zespół scrumowy i interesariusze omawiają następne kroki; identyfikowane są nowe

elementy rejestru, a stare usuwane. Upewnij się, że proces porządkowy jest ustalony tak, aby wszelkie aktywności przeprowadzane były w sposób solidny, na przykład podczas warsztatów porządkowych na początku każdego tygodnia. Dobrze uporządkowany rejestr jest warunkiem udanego zebrania planującego sprint.

Do wspierania tych czynności przyda się doskonałe narzędzie: papierowe karty. Są tanie i łatwe w użyciu. Wspomagają współpracę — każdy może wziąć jedną i spisać swój pomysł. Można je też układać na stole lub ścianie, sprawdzając spójność i kompletność. Karty i elektroniczne narzędzia do obsługi rejestru produktu, takie jak na przykład arkusze kalkulacyjne, wzajemnie się uzupełniają. Przed warsztatem porządkowym wydrukuj istniejące wymagania na kartach, a po spotkaniu przenieś wszystkie informacje z kart z powrotem do systemu.

Przyjrzyj się teraz bliżej czterem krokom procesu porządkowania, zaczynając od odkrywania i opisywania elementów rejestru produktu.

Odkrywanie i opisywanie elementów

Odkrywanie i opisywanie elementów rejestru produktu jest procesem ciągłym. Jeżeli jesteś przyzwyczajony do wcześniejszego tworzenia złożonych i szczegółowych specyfikacji wymagań, pamiętaj, że Scrum zachęca do zupełnie innego podejścia. Wymagania nie są już zamrożone na wczesnym etapie, ale odkrywane i opisywane w trakcie całego projektu. W miarę poprawiania się Twojego rozumienia potrzeb klientów oraz ich zaspokajania istniejące wymagania będą się zmieniać lub staną się niepotrzebne. Na ich miejsce pojawią się nowe. W Scrumie odkrywanie produktu nie odbywa się więc wyłącznie na wczesnym etapie rozwoju, lecz pokrywa się z całym projektem. Wielu menedżerów produktu przechodzących do roli właściciela produktu za duże wyzwanie uważa zakaz spisywania wszystkich wymagań na początku projektu, nawet gdyby było to możliwe.

Odkrywanie elementów

Odkrywanie elementów rejestru produktu rozpoczyna się wraz z uzupełnieniem rejestru. Najlepiej, kiedy zespół scrumowy i, jeśli to konieczne, interesariusze pracują nad tym wspólnie, wymyślając elementy niezbędne do powołania produktu do życia. Mogą przy tym korzystać z idei produktu, jego wizji czy też mapy produktu. W trakcie uzupełniania rejestru produktu unikaj błędów, próbując wypisać wszystkie możliwe elementy. Podczas pracy nad rejestrem skupiaj się na **minimalnej** liczbie funkcjonalności, niezbędnej, by powołać produkt do życia. Należy dążyć do prostoty, jak już pisałem w rozdziale 2. W miarę postępu projektu pojawiać się będzie coraz więcej pomysłów, a rejestr rozrośnie się na podstawie opinii zwrotnych klientów i użytkowników. Tworzenie zbyt długich i złożonych elementów rejestru sprawia, że trudno jest utrzymać koncentrację i ustalić priorytety. Idea lub wizja produktu może Ci w tym pomóc. Skup się jedynie na tym, co jest niezbędne, nie przejmując się resztą. Oprzyj się pokusie zbyt szybkiego dodawania wielu szczegółów. Detale dodaje się stopniowo, biorąc pod uwagę priorytety. Elementy o niskim priorytecie są duże i ogólne. Zmienia się to dopiero wraz ze zmianą priorytetu (lub dlatego, że elementy o wyższym priorytecie zostały już wykonane). Wymagania нефunkcjonalne, które reprezentują właściwości całego produktu, są wyjątkiem od tej reguły. Powinny być one wyszczególnione jak najwcześniej, co wyjaśnię w dalszej części tego rozdziału.

Kiedy wstępny rejestr produktu jest już gotowy, pojawia się wiele możliwości odkrywania nowych elementów. Ujawniają się one w trakcie warsztatów porządkowych, kiedy zespół scrumowy ustala priorytety i rozkłada elementy rejestru produktu na mniejsze części. Ujawniają się również w trakcie przeglądów sprintu, kiedy interesariusze wyrażają swoje opinie, oraz kiedy klienci i użytkownicy komentują nowe przyrosty produktu.

Za każdym razem, kiedy nowe wymaganie wpisywane jest do rejestru, upewnij się, że wynikające z niego wymagania klienta są zrozumiałe. Zapytaj, dlaczego jest ono konieczne oraz jakie korzyści przy-

niesie klientom. Nie kopiuj wymagań bezmyślnie do rejestru produktu, ponieważ w ten sposób tworzy się niespójna i niemożliwa do zarządzania lista pobożnych życzeń. Uważaj istniejące wymagania za podejrzane i potraktuj je jako odpowiedzialność, a nie kapitał. Wymaganie opisuje po prostu funkcjonalność produktu, która na pewnym etapie została uznana za konieczną. W miarę jak zmieniają się rynki i technologie, a zespół scrumowy zyskuje wiedzę na temat spełniania potrzeb klientów, zmieniają się również wymagania, do tego stopnia, że niektóre z nich stają się zbędne.

Opisywanie elementów

Scrum nie określa sposobu opisywania elementów rejestru produktu, ale ja preferuję pracę z *user stories* [Cohn, 2004]. Jak sugeruje nazwa, *user story*, czyli historia użytkownika, opowiada o kliencie lub użytkowniku korzystającym z produktu. Zawiera imię, krótki opis oraz kryteria akceptacji czy warunki, które muszą być spełnione, by historia była kompletna. *User story* może być ogólne lub szczegółowe; ogólne historie zwane są **epikami**. *User stories* są łatwe do napisania, dekompozycji i dopracowania. Do opisanía wymagań możesz oczywiście wykorzystać również inne techniki. Jeśli jednak korzystasz z *user stories*, nie powinienes czuć się zobligowany do opisanía każdego elementu rejestru jako historii. Przykładowo: wymagania dotyczące użyteczności najlepiej obrazują prototypy lub szkice.

Praca z rejestrem produktu nie oznacza, że zespół scrumowy nie może tworzyć innych pomocnych artefaktów, w tym streszczeń *user stories*, historii modelujących przepływ pracy, diagramów ilustrujących zasady biznesowe, arkuszy kalkulacyjnych przedstawiających złożone obliczenia, szkiców interfejsu użytkownika, scenopisów, diagramów nawigacyjnych oraz prototypów interfejsu użytkownika. Te artefakty nie zastąpią rejestru produktu, ale pomogą opracować i wyjaśnić jego zawartość. Powinny być proste. Wykorzystuj tylko te artefakty, które pomogą zespołowi scrumowemu w przybliżeniu się do gotowego produktu.

Ustalanie struktury rejestru

Rejestry produktu dużo zyskują dzięki pogrupowaniu powiązanych elementów w motywy. Motywy pełnią funkcję symbolu zastępczego produktu, tworzą strukturę rejestru, pomagają w ustalaniu priorytetów i ułatwiają dostęp do informacji. Przykładowymi motywami dla telefonu komórkowego będą: e-mail, kalendarz, komunikacja głosowa i organizier. Najważniejszą zasadą jest przypisanie na samym początku do każdego motywu od dwóch do pięciu ogólnych wymagań. Zapewni to wystarczającą ilość informacji, by zrozumieć, co należy zrobić, aby powołać produkt do życia, bez konieczności zawierania zbyt dużej liczby specyfikacji w rejestrze. Motywy tworzą w rejestrze produktu hierarchię, która na tym etapie zawiera zarówno indywidualne elementy, jak i grupy. Przydatne może być również dalsze rozróżnianie w rejestrze elementów ogólnych, takich jak epiki, oraz szczegółowych, jak *user stories*, co pokazałem w tabeli 3.1.

Tabela 3.1. Przykładowy rejestr produktu

Motyw	Element ogólny	Element szczegółowy	Wysilek
E-mail	Stworzenie e-maila	Jako przedsiębiorca chcę mieć możliwość określenia tematu e-maila	1

Motyw w tabeli 3.1 zawiera elementy ogólne. Z czasem będą one rozłożone na bardziej szczegółowe. W miarę dokonywania przez zespół prognoz dotyczących elementów rozmiar tych ostatnich jest zapisywany. Zauważ, że strukturę w tabeli 3.1 można wdrożyć niezależnie od narzędzia wykorzystywanego do tworzenia rejestru produktu. Wystarczy odpowiednio rozłożyć papierowe karty na tablicy lub ścianie biura.

Ustalanie priorytetów rejestru produktu

Nigdy nie zapomnę dnia, w którym zasugerowałem menedżerze produktu związanego z ochroną zdrowia, żeby ustaliła priorytety dla stosu elementów leżących przed nią. Popatrzyła na mnie szeroko otwartymi oczyma i powiedziała: „Nie mogę. Wszystkie mają wysoki priorytet”.

Ustalanie priorytetów wymaga podjęcia decyzji dotyczących wagi poszczególnych elementów. Jeżeli wszystko ma wysoki priorytet, wszystko jest równie ważne. Oznacza to, że tak naprawdę nic nie jest priorytetem, więc istnieje nikkie prawdopodobieństwo, że klient dostanie to, czego naprawdę potrzebuje. Za ustalenie priorytetów dla każdego elementu rejestru produktu odpowiedzialny jest właściciel produktu. Tak jak i inne czynności związane z porządkowaniem, ustalanie priorytetów najlepiej przeprowadzić przy udziale całego zespołu scrumowego. Wyrównuje to kolektywny poziom wiedzy zespołu, a także tworzy więzi.

Ustalane priorytety kierują pracą zespołu, skupiając uwagę jego członków na najważniejszych aspektach. Powodują także stopniowe zamrażanie zawartości rejestru. Jak już wspomniałem, elementy rejestru mają ilość szczegółów odpowiednią dla ich priorytetu. To sprawia, że proces jest elastyczny i pozwala na opóźnianie decyzji dotyczących elementów o niższym priorytecie, dając zespołowi scrumowemu więcej czasu na ocenę opcji, zebranie opinii klientów oraz zdobycie dodatkowej wiedzy. W ten sposób podejmowane są lepsze decyzje i tworzone lepsze produkty².

Ponieważ indywidualne elementy rejestru produktu mogą być niewielkie i trudno będzie określić ich priorytet, najlepiej, jeśli najpierw przyznasz priorytety motywom. Następnie ustal priorytety wewnątrz motywów lub, jeśli to konieczne, pomiędzy nimi. W dalszym ciągu tej części rozdziału opiszę następujące czynniki wpływające na

² Opóźnianie decyzji do ostatniego momentu nazywa się również „ostatnim odpowiedzialnym momentem” [Poppendieck i Poppendieck, 2003].

ustalanie priorytetów rejestru produktu: wartość, wiedzę, niepewność i ryzyko, zdolność do wypuszczenia na rynek oraz zależności.

Wartość

Wartość jest powszechnym czynnikiem wykorzystywanym przy ustalaniu priorytetów. Z pewnością chcesz dostarczyć najbardziej wartościowe elementy na początku. Ale co sprawia, że element rejestru produktu staje się wartościowy? Odpowiedź jest prosta. Element jest wartościowy, jeżeli jest niezbędny w procesie powoływania produktu do życia. Jeżeli taki nie jest, wówczas okazuje się zbędny; zostaje on wyłączony z prac nad bieżącą wersją wydania lub produktu. Zespół scrumowy może zmienić priorytet elementu, umieszczając go na samym dole rejestru produktu lub — co jest najlepszym rozwiązaniem — usuwając go. Drugi sposób zapewnia zwężłość rejestru oraz lepszą koncentrację zespołu scrumowego. Jeżeli element jest ważny dla przyszłych wersji, pojawi się ponownie.

Przed dodaniem elementu do wydania zdecyduj, czy produkt będzie spełniał swoje zadanie bez niego. Jest to pomocne w tworzeniu prostego produktu, który zawiera minimalną liczbę niezbędnych funkcjonalności, jak pokazałem w rozdziale 2. Na przykład Apple wypuściła na rynek pierwszą i drugą generację iPhone'ów bez funkcjonalności umożliwiającej kopiowanie i wklejanie, co nie wpłynęło negatywnie na sukces produktu. Jeżeli element rzeczywiście jest niezbędny, sprawdź, czy istnieje alternatywa, która zapewni te same korzyści przy mniejszym nakładzie wysiłku i czasu lub zredukuje koszt jednostkowy. Mimo że brzmi to jak oczywistość, zespoły są często ograniczone ukrytymi założeniami i nie zawsze dostrzegają wszystkie opcje.

Nie rozpatruj tylko nowych wymagań. Zbadaj również te istniejące. Alternatywy pojawiają się często po tym, jak zespół scrumowy dowie się więcej na temat potrzeb klientów oraz tworzonego rozwiązania. Należy upraszczać, kroić i utrzymywać porządek — niczym ogrodnik usuwający chwasty i przycinający krzewy.

Jeżeli jest jakaś wątpliwość, trzeba usunąć wymaganie i wypuścić wydanie bez niego — tak jak zrobiła firma Google, wypuszczając Google News, aplikację agregującą wiadomości z całego świata. Zespół deweloperów nie mógł dojść do porozumienia, czy filtrowanie wiadomości powinno odbywać się według daty, czy też lokalizacji. Firma zdecydowała, że w wydaniu nie znajdzie się żadna z wymienionych właściwości. Krótco po wypuszczeniu produktu na rynek zaczęły napływać pytania dotyczące nowych funkcjonalności. Trzysta osób prosiło o filtrowanie według daty, a tylko trzy według lokalizacji — co było jasną wskazówką dotyczącą tego, która funkcjonalność powinna zostać rozwinięta jako pierwsza. Gdyby Google wypuścił produkt z oboma właściwościami, wydanie pochłonęłoby więcej czasu i pieniędzy. Uzyskanie opinii zwrotnej dotyczącej tego, która właściwość jest ważniejsza, byłoby trudniejsze. Celowo wypuszczając okrojoną wersję produktu, Google szybko przekonał się, co należy zrobić.

Wiedza, niepewność i ryzyko

„Ryzyko jest niezbędną cechą innowacji produktu. Każda decyzja dotycząca projektu: czy to warunkowa, czy bezwarunkowa pociąga za sobą ryzyko” — piszą Preston G. Smith i Guy M. Merritt [2002, 4]. Ryzyko jest wobec tego częścią rozwoju oprogramowania; żaden powstający produkt nie jest od niego wolny. Z ryzykiem związana jest niepewność. Im więcej niepewności, tym bardziej ryzykowny jest projekt. Niepewność z kolei spowodowana jest brakiem wiedzy. Im mniej wiesz na temat tego, co chcesz stworzyć i jak to zrobić, tym bardziej niepewnie się czujesz. Wiedza, niepewność i ryzyko są zatem połączone.

Ponieważ ryzyko i niepewność wpływają na sukces produktu, niepewne i ryzykowne elementy powinny mieć wysoki priorytet. Przyspiesza to zdobywanie nowej wiedzy, niweluje niepewność i redukuje ryzyko. Jeżeli zespół scrumowy nie jest pewien niektórych aspektów zawartych w projekcie interfejsu użytkownika, odpowiednie opcje powinny zostać zbadane i przetestowane poprzez zebranie opinii klientów

i użytkowników. Jeżeli zespół nie wie, czy powinna zostać zastosowana warstwa dostępu do bazy danych firmy zewnętrznej, wymagania dotyczące transakcji bazodanowych powinny zostać zaimplementowane na wczesnym etapie, tak aby mogły być poddane ocenie różne opcje. Pamiętaj, że ryzyko może się również kryć w infrastrukturze i środowisku, szczególnie w niestworzonym procesie budowy lub niezlokalizowanym zespole scrumowym.

Walka z niepewnymi, ryzykownymi elementami na wczesnym etapie może spowodować wczesną porażkę. Wczesna porażka pozwala zespołowi scrumowemu na zmianę kursu, kiedy jest jeszcze taka szansa. Można wtedy zmodyfikować architekturę i technologię albo zmienić skład zespołu. Takie podejście bywa trudne do zaakceptowania dla osób i organizacji przyzwyczajonych do tradycyjnych procesów, w których problemy i przeszkody pojawiają się na późnym etapie i traktowane są jako złe wieści, a nie okazja do nauki i wprowadzania zmian.

Zdatność do wypuszczenia na rynek

Wczesne i częste wypuszczanie oprogramowania to doskonały sposób na rozwijanie go tak, by stało się produktem, który pokochają klienci. Powiem o tym w rozdziale 4. Jest to również efektywny sposób na ograniczanie ryzyka. Jeżeli zespół scrumowy jest niepewny tego, czy i jak dana właściwość powinna zostać zaimplementowana, wczesne wydania mogą odpowiedzieć na te pytania, tak jak w przypadku omawianej wcześniej aplikacji Google News.

Możliwość wypuszczania przyrostów produktu wcześniej i często powinna wpłynąć na ustalanie priorytetów w rejestrze produktu. Każde wydanie musi zapewniać funkcjonalności, które są użyteczne dla klientów i użytkowników oraz generują pożądane opinie. Pamiętaj, że zazwyczaj nie jest konieczna pełna implementacja motywu; częściowe wdrożenie bywa dla wczesnych wydań wystarczające.

Zależności

Zależności są stałym punktem rejestru produktu, bez względu na to, czy je lubisz, czy też nie. Wymagania funkcjonalne na przykład często zależą od innych wymagań funkcjonalnych, a nawet нефunkcjonalnych. Jeżeli kilka zespołów pracuje razem, zależności między nimi mogą wpłynąć na ustalanie priorytetów, co omówię bardziej szczegółowo w rozdziale 4. Zależności ograniczają wolność ustalania priorytetów w rejestrze produktu oraz wpływają na ocenę ilości pracy; element, od którego zależą wysiłki innych, musi zostać wdrożony jako pierwszy. Powinieneś w związku z tym postarać się usunąć zależności, kiedy to możliwe.

Łączenie kilku zależnych elementów w jeden większy, a także rozdzielanie elementów to dwie powszechne techniki radzenia sobie z zależnymi *user stories* [Cohn, 2004, 17]. Przyjrzyj się dwóm przykładowym historiom: „Jako użytkownik chcę mieć możliwość pisania wiadomości tekstowych” oraz „Jako użytkownik chcę mieć możliwość pisania e-maili”. Obie historie są zależne, ponieważ wymagają możliwości przetwarzania tekstu. Jeżeli jako pierwszą wdrożysz historię dotyczącą pisania wiadomości, wysiłek włożony w historię dotyczącą e-maili będzie zredukowany i odwrotnie. Pierwszym wyjściem jest połączenie ich w jedną większą historię. Nie jest to zbyt dobry pomysł, ponieważ powstała historia będzie bardzo rozbudowana. Drugim wyjściem jest podzielenie wymagań w inny sposób. Jeżeli wspólna funkcjonalność („Jako użytkownik chcę mieć możliwość pisania”) zostanie zapisana osobno, te dwie historie nie będą już od siebie zależne. W ten sposób na dotyczące ich prognozy nie ma już wpływu kolejność wykonywania.

Przygotowanie do planowania sprintu

Przed każdym spotkaniem planującym sprint muszą zostać przygotowane elementy rejestru produktu, nad którymi zespół będzie pracował w następnej kolejności. Przygotowania zaczyna się od wybrania celu sprintu.

Wybór celu sprintu

Cel sprintu podsumowuje oczekiwany wynik sprintu. Powinien przybliżyć zespół scrumowy o krok w stronę wydania udanego produktu. Właściciel produktu w projekcie, nad którym pracowałem, wybrał dla pierwszego sprintu następujący cel: „Wysokie drzewa mają głębokie korzenie”. Cel ten bardzo ładnie opisywał zamierzenia sprintu: stworzenie fundamentów dla całego projektu. Dobry cel sprintu jest szeroki, ale realistyczny. Powinien zostawić trochę miejsca dla zespołu na manewry i być aktualny nawet, kiedy zespół nie zobowiąże się do wykonania wszystkich głównych elementów rejestru produktu. Tak jak w przypadku aktywności porządkowych, zespół powinien brać udział w tworzeniu celu. Gwarantuje to jasność przekazu i zaangażowanie pracowników.

Cele sprintu przynoszą korzyści z kilku powodów.

- Tworzą wspólną płaszczyznę pomiędzy właścicielem produktu, mistrzem młyna oraz zespołem: wszyscy pracują, aby osiągnąć jeden cel.
- Minimalizują zmiany, ograniczając typ wymagań, nad którym zespół będzie pracował w danym sprincie. Można to zrobić, wybierając elementy z tego samego motywu. Ułatwia to bliską współpracę i pomaga w osiągnięciu określonej szybkości.
- Ułatwiają komunikację z interesariuszami dotyczącą tego, nad czym pracuje zespół.

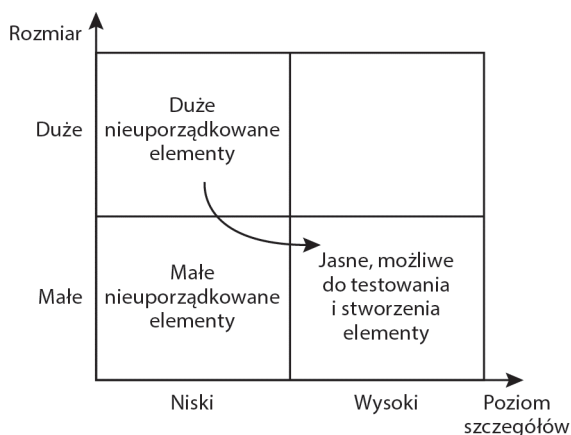
Pamiętaj, że wybranie celu sprintu może prowadzić do zmian w priorytetach rejestru produktu, do których należy usuwanie i dodawanie elementów na szczycie listy. Być może będziesz zmuszony do dokonania wyboru pomiędzy spójnym celem sprintu a sprawną pracą nad projektem. Kiedy cel zostanie już ustalony, wszystkie stosowne elementy powinny się znaleźć na szczycie rejestru produktu.

Przygotowanie wystarczającej liczby elementów dokładnie na czas

Kiedy cel sprintu zostanie wybrany, należy przygotować wystarczającą liczbę elementów dokładnie na czas³. Temat dużych projektów, które wymagają wybiegania w przyszłość, omówię w dalszej części tego rozdziału. Czynności porządkowe w pierwszym sprincie polegają na skupieniu się na elementach niezbędnych w drugim sprincie, a te w drugim na elementach niezbędnych w trzecim i tak dalej. Podejście to ma kilka korzyści: minimalizuje ilość czasu i pieniędzy, które należy zainwestować w opisywanie elementów rejestru produktu, oraz utrzymuje spis szczegółowych elementów na odpowiednim poziomie — zapewnianie większej ilości informacji niż to konieczne jest marnotrawstwem. Dodając szczegóły tylko do elementów, które najprawdopodobniej zostaną wybrane w następnym sprincie, pozwalasz na rozwój rejestru produktu.

Przygotowanie elementów do zebrania planującego sprint wymaga rozłożenia większych elementów na mniejsze czynniki do momentu, w którym są one wystarczająco niewielkie, by zmieścić się w sprincie. Należy również uporządkować elementy tak, by były jasne, dostępne i nadające się do testów. Rysunek 3.2 właściwie ilustruje ten proces. Pamiętaj, że dekompozycja elementów może zająć kilka sprintów, o czym zaraz powiem.

³ Terminy „wystarczająca ilość” i „dokładnie na czas” zostały po raz pierwszy użyte przez Cohna [2008] w dyskusji na temat czynności porządkowych.



Rysunek 3.2. Dekompozycja i porządkowanie elementów rejestru produktu

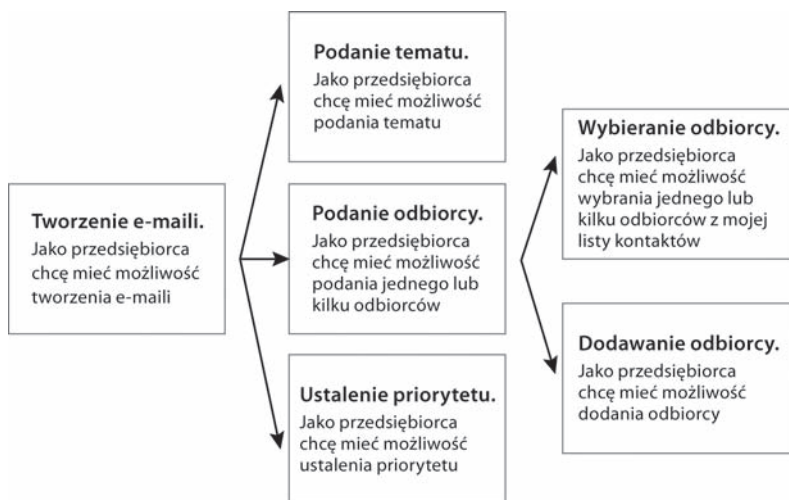
To, ile elementów trzeba przygotować, zależy od szybkości zespołu oraz oczekiwanej ilości szczegółów. Im większa jest szybkość zespołu, tym więcej elementów musi zostać przygotowanych. Pomocne może być dołączenie kilku dodatkowych elementów, by dać zespołowi możliwość wyboru. Przydadzą się one również w przypadku, gdy prace nad sprintem postępują szybciej niż przewidywano. Lubię pracować nad niewielkimi projektami, które mogą zostać ukończone w ciągu kilku dni, bez względu na długość sprintu. W ten sposób usprawnione jest śledzenie postępu zespołu w trakcie sprintu, a także samoorganizacja: postęp zespołu opiera się nie tylko na pozostałych zadaniach, ale także na liczbie zaimplementowanych funkcjonalności, które zostały przetestowane i udokumentowane. Niewielkie wymagania minimalizują ilość bieżącej pracy oraz ryzyko otrzymania na koniec sprintu elementów częściowych i wadliwych. Dodatkowo niewielkie elementy ułatwiają składanie realistycznych zobowiązań. Te większe mogą zawierać tak wiele zadań, że zespół nie będzie w stanie ich zidentyfikować.

Dekompozycja elementów

Dekompozycja elementów rejestru produktu oznacza zmniejszanie ich gabarytów do momentu, w którym będą mieściły się w jednym sprincie. Ten proces, zwany także „progresywną dekompozycją wymagań” [Reinertsen, 1997], może trwać dłużej niż jeden sprint. Możesz być zmuszony do rozpoczęcia dekompozycji elementu rejestru produktu kilka sprintów wcześniej, szczególnie jeżeli dany element jest duży i skomplikowany. Pozwala to na zebranie opinii klientów, użytkowników i innych interesariuszy przed ustalaniem szczegółów elementu. Przyjrzyj się progresywnej dekompozycji *user stories*.

Jak pokazano na rysunku 3.3, zespół scrumowy umieścił w rejestrze produktu epik „Tworzenie e-maili”. Ponieważ jest on zbyt duży i niejasny, by mógł zostać dostarczony w jednym sprincie, należy rozbić go na kilka ogólnych *user stories*. Historia „Podanie odbiorcy” jest następnie rozebrana na dwie kolejne ogólne *user stories*. Dopiero wtedy są one gotowe, by włączyć je do sprintu. Epik jest przykładem historii złożonej, która ma więcej niż jeden cel [Cohn, 2004, 24 – 25]. Aby ją zdekomponować, należy wprowadzić osobną historię dla każdego celu. „Tworzenie e-maili” zostaje więc rozbite na „Podanie tematu”, „Podanie odbiorcy” oraz „Ustalanie priorytetów”.

Istnieją inne *user stories*, które również muszą zostać zdekomponowane, w tym historie złożone oraz historie z rozbudowanymi kryteriami. Historia złożona to taka, która jest zbyt duża, by mogła być dostarczona w jednym sprincie, z powodu jej nieodłącznej niepewności lub zbyt dużej liczby funkcjonalności [Cohn, 2004, 25 – 26]. Jeżeli niepewność jest zbyt duża, wprowadź do rejestru jeden lub kilka elementów, które rozwiązują tę niepewność, tworząc odpowiednią bazę wiedzy, na przykład: „Zbadanie JavaServer Faces jako technologii interfejsu użytkownika”. Jeżeli historia opisuje zbyt dużą liczbę funkcjonalności, należy je podzielić, by umożliwić przyrostową dostawę funkcjonalności. Ta technika zwana jest również „krojeniem ciasta” [Cohn, 2004, 76]. Historia „Walidacja użytkownika” może zostać rozłożona na historie „Walidacja nazwy użytkownika” oraz „Walidacja hasła”.



Rysunek 3.3. Dekompozycja user stories

Historie często wyglądają dobrze, dopóki nie rozważy się kryteriów akceptowalności. Jeżeli jest ich zbyt wiele — więcej niż dziesięć — lub jeżeli wymagania ukryte są w kryteriach, musisz zmienić kompozycję historii. Oto przykład: „Jako użytkownik chcę mieć możliwość kasowania wiadomości tekstowych”. Kryterium akceptacji mówi: „Mogę wybrać jakąkolwiek wiadomość. Mogę usunąć tekst wiadomości. Mogę zapisać zmodyfikowaną wiadomość”. Drugi warunek jest zbędny, a pozostałe, zamiast specyfikacji kryteriów akceptacji, wprowadzają nowe wymagania. Ta historia powinna być podzielona na trzy: na historię odnoszącą się do kasowania wiadomości tekstowych, historię odnoszącą się do edytowania wiadomości tekstowych oraz historię dotyczącą zapisywania zmodyfikowanych wiadomości.

Zapewnianie klarowności, możliwości testowania i wykonalności

Kiedy element jest już wystarczająco mały, musisz upewnić się, że jest klarowny, możliwy do przetestowania i wykonalny⁴. Wymaganie jest **jasne**, jeżeli wszyscy członkowie zespołu scrumowego jednakowo rozumieją jego semantykę. Wspólne opisywanie wymagań oraz wyrażanie elementów rejestru produktu w prosty i zwięzły sposób wspomaga przejrzystość. Element **umożliwia testowanie**, jeżeli istnieje efektywny sposób sprawdzenia w trakcie bieżącego sprintu, czy wymaganie zostało spełnione. Historie muszą mieć kryteria akceptowalności, które pozwolą sprawdzić możliwości testowania. Element jest **możliwy do zrealizowania**, jeżeli da się go ukończyć w trakcie jednego sprintu, zgodnie z definicją pojęcia „gotowe”. Definicja tego pojęcia omówiona jest w rozdziale 5. Takie kryterium osiągniesz, rozważając zależności od innych elementów, zarówno funkcjonalne, jak i niefunkcjonalne. Jeżeli historia jest ograniczona wymaganiami interfejsu użytkownika, musi być jasne, jak ma wyglądać docelowy przyrost produktu. W innym przypadku zespół powinien przyrzeć się wymaganiom interfejsu przed implementacją historii. Jeżeli eksploracja elementu wymaga dużego nakładu sił, powinno się ją przeprowadzić w osobnym sprincie, na przykład z wykorzystaniem jednorazowego prototypu, który pomoże w projektowaniu interfejsu.

⁴ Bill Wake sugeruje, że historie powinny być niezależne, negocjowalne, wartościowe, umożliwiające szacunki i testowanie, a także niewielkie [Wake, 2003]. Kryteria te nazwane zostały INVEST. Zależności oraz wartości zostaną omówione w rozdziale dotyczącym ustalania priorytetów, razem z metodami określania szacunków. Negocjowanie odnosi się do możliwości dostosowania *user stories*. Historia jest obietnicą konwersacji, jak twierdzi Ron Jeffries, a nie twardym wymogiem.

Dostosowywanie wielkości elementów

Szacowanie elementów rejestru produktu pozwala na ogólne określenie ich rozmiaru i ilości pracy niezbędnej do włożenia w ich przygotowanie. Jest to pomocne z dwóch powodów: pomaga w ustalaniu priorytetów oraz pozwala śledzić i prognozować postęp projektu. Zauważ, że w procesie Scrum istnieją dwa odrębne szacunki: ogólne w rejestrze produktu, wskazujące przybliżony rozmiar elementu, a także szczegółowe w rejestrze sprintu, opisujące rozmiar zadania, zwykle podawane w godzinach. W tej części rozdziału omówię dostosowywanie rozmiaru elementów w rejestrze produktu. Elementy rejestru produktu są szacowane, kiedy odkrywa się nowe lub modyfikuje istniejące oraz kiedy zespół orientuje się, że rozmiar elementu się zmienił. Potrzebujesz miary, która jest szybka i prosta w użyciu. Moją ulubioną są punkty⁵.

Punkty

Punkty to miara ogólna i relatywna, dotycząca włożonego wysiłku i rozmiaru danego elementu⁶. Element wart jeden punkt jest o połowę mniejszy niż ten wart dwa punkty. Element oceniony na trzy punkty wymaga tyle samo pracy co element wart jeden punkt i element wart dwa punkty łącznie. Miara relatywna wykorzystuje to, że rozmiar jest również relatywny; semantyka dużego i małego zależy od punktu odniesienia. Moja myszka komputerowa jest mała w porównaniu do mojego laptopa, ale duża w porównaniu do pamięci USB, która leży obok. Popularny zakres wykorzystywanych punktów pokazuje tabela 3.2.

⁵ Więcej szczegółów oraz obszerną dyskusję na temat technik określania szacunków znajdziesz u Cohna [2005].

⁶ Czas mierzony jest osobno, za pomocą szybkości, co omówię w rozdziale 4.

Tabela 3.2. Popularne zakresy punktów

Punkty	Rozmiar koszulki	
0	gratis, element został już stworzony	
1	XS	bardzo mały
2	S	mały
3	M	średni
5	L	duży
8	XL	bardzo duży
13	XXL	bardzo bardzo duży
20	XXXL	olbrzymi

Nielinearna sekwencja z tabeli 3.2 przyspiesza proces podejmowania decyzji przez zespół. Zapobiega długim dyskusjom na temat odpowiednich wartości, które mogą pojawić się w przypadku sekwencji linearnych. Zespół może rozszerzyć zakres pokazany w tabeli 3.2, dodając 40 i 100 do największych wartości, pod warunkiem że szacunki relatywne będą prawidłowe. Bez względu na wybrany zakres zespół powinien trzymać się sekwencji i czuć się z nią komfortowo. Ponieważ punkty są relatywne i arbitralne, nie można ich porównywać pomiędzy zespołami, chyba że wszystkie zespoły uzgodnią wspólny zakres i semantykę.

Poker planistyczny

Punkty są świetne, ale same w sobie nie wystarczą. Potrzebujesz techniki, która umożliwi zespołowi efektywne szacowanie. Taką techniką jest poker planistyczny [Cohn, 2005, 56 – 59]. Polega on na tym, że każdy członek zespołu dostaje talię kart zawierającą wszystkie, uzgodnione wcześniej, wartości punktów. Gdybyś wykorzystywał zakres z tabeli 3.2, talia zawierałaby osiem kart, a każda z kart — jeden z punktów zakresu. Po rozdaniu uczestnikom kart rozpoczyna się szacowanie.

Jeżeli zespół wyznacza wielkość elementów rejestru po raz pierwszy, wskazane jest określenie znaczenia wartości zakresu. Aby to zrobić, wiele zespołów jako pierwszy wybiera element, co do którego niewielkich rozmiarów nikt nie ma wątpliwości. Innym rozwiązaniem jest wybranie najmniejszego, największego oraz średniego elementu i oszacowanie ich po kolei. Jeżeli jednak zespół jest zaznajomiony z zakresem, może zacząć od elementu mającego najwyższy priorytet i stopniowo schodzić w dół rejestru.

Zanim zespół dokona szacunków, właściciel produktu musi objaśnić element wszystkim członkom zespołu, którzy następnie omawiają kolejne kroki niezbędne do wdrożenia produktu zgodnie z definicją pojęcia „gotowe”. Po dyskusji każdy członek zespołu dokonuje własnej oceny wielkości elementu, bez sugerowania się tym, kto będzie go rozwijał. Przydziału zadań dokonuje się podczas odpowiedniego codziennego zebrania zespołu scrumowego. Każdy członek zespołu wybiera kartę zawierającą odpowiednie szacunki i kładzie ją obrazkiem w dół na stole. Kiedy już wszyscy wyłożyli swoje karty, należy je jednocześnie odwrócić obrazkami ku górze. Jeżeli szacunki różnią się między sobą, dwóch członków zespołu, których wartości były najbardziej zróżnicowane, argumentuje swój wybór. Następnie przeprowadzana jest kolejna runda. Wszystkie karty wracają do talii, członkowie zespołu po raz kolejny wybierają kartę, która najlepiej odpowiada ich szacunkowi. Nie musi to być ta sama karta co za pierwszym razem. Gra jest kontynuowana, dopóki szacunki się nie pokryją. Zasadą podejmowania decyzji jest konsensus, wszyscy członkowie zespołu powinni czuć się swobodnie. Gdy tylko zespół oszacuje co najmniej dwa elementy, nowe szacunki powinny zostać porównane z istniejącymi, aby upewnić się, że relatywny rozmiar jest odpowiedni. Można to uczynić, grupując elementy w zależności od rozmiaru.

Szacowanie wymagań niefunkcjonalnych

Dla wymagań niefunkcjonalnych, które dotyczą wszystkich wymagań funkcjonalnych (na przykład *user experience* lub wydajność), nie tworzy się osobnych szacunków. Należy je włączyć do definicji pojęcia „gotowe”. Jeżeli jednak do stworzenia wymagania niefunkcjonalnego potrzebna będzie konkretna ilość pracy, tak jak w przypadku odkrywania różnych opcji projektów interfejsu użytkownika lub refaktoringu architektonicznego, odpowiednie elementy powinny być umieszczone w rejestrze produktu, a ich wielkość określona przez zespół. Włączanie wymagań niefunkcjonalnych do definicji pojęcia „gotowe” nie oznacza, że nie wymagają one żadnej pracy. Wręcz przeciwnie: definicja pojęcia „gotowe” wpływa na szacunki zespołu.

Aby uzyskać w miarę dokładne szacunki, muszą być spełnione trzy warunki: zespół musi wiedzieć, ile pracy trzeba włożyć w dostarczenie danego elementu, członkowie zespołu muszą być w stanie określić zależności od innych elementów, a także musi być dostępna definicja pojęcia „gotowe”. Jeżeli zespół nie jest w stanie oszacować danego elementu, należy dodać go do rejestru, co spowoduje przyrost wiedzy, na przykład: „Stworzenie prototypu lub makiety pomagającej w odkrywaniu opcji interfejsu użytkownika”.

Tylko członkowie zespołu, którzy tworzą przyrosty produktu, powinni mieć możliwość dokonywania szacunków elementów rejestru produktu. Właściciel produktu i mistrz młyna nie powinni dokonywać szacunków ani też wpływać na nie (chyba że są członkami zespołu lub zespół poprosi ich o radę). Właściciel produktu musi jednak być obecny na zebraniu. Wiele elementów rejestru produktu będzie miało formę szkicu, a właściciel produktu będzie musiał je objaśnić.

Szybkie szacunki

Jeżeli zespół ma zbyt mało czasu na poker planistyczny, można wykorzystać następującą technikę szacunkową. Należy podzielić jedną ścianę pokoju zebrań na kilka części, a każdą z nich opatrzyć etykietą zawierającą liczbę z zakresu punktów. Wydrukuj elementy rejestru produktu na kartach i rozłóż je na stole. Poproś każdego członka zespołu o wybranie jednej karty, dokonanie szacunku, a następnie przytwierdzenie karty w odpowiednim miejscu na ścianie, upewniając się, że

wszystkie karty w danej grupie wyobrażają elementy o jednakowych rozmiarach. Jeżeli ktokolwiek zauważy niepasującą kartę, powinien ją przenieść do właściwej grupy. Dzięki temu procesowi można szybko stworzyć szacunki dla wielu elementów rejestru, wkładając w to minimalny wysiłek. Główną wadą tego procesu jest to, że zespół nie dyskutuje na temat rozmiaru elementów, w związku z tym jakość szacunków będzie niższa niż w przypadku pokera planistycznego.

Postępowanie w przypadku wymagań niefunkcjonalnych

Wymagania niefunkcjonalne (nazywane również wymaganiami operacyjnymi, cechami systemu i ograniczeniami) to brzydkie kaczątko rozwoju oprogramowania. Są często zaniedbywane, mimo że opisują ważne właściwości, takie jak: wydajność, solidność, skalowalność, użyteczność, a także wymagania techniczne oraz te związane ze zgodnością (na przykład wspieranie protokołu lub możliwość uzyskania certyfikatu). Mają one wpływ na projekt interfejsu użytkownika, architekturę, wybór technologii oraz na ostateczny koszt produktu i długość jego życia. W tej części rozdziału opowiem o opisywaniu niefunkcjonalnych wymagań w Scrumie i zarządzaniu nimi.

Opisywanie wymagań niefunkcjonalnych

Wymagania niefunkcjonalne mogą być wyrażone w postaci ograniczeń [Newkirk i Martin, 2001, 16 – 18]. Możesz na przykład opisać wydajność w sposób przedstawiony na rysunku 3.4.

Wymagania dotyczące *user experience* najłatwiej zapisuje się w postaci szkiców, scenopisów, diagramów nawigacyjnych interfejsu użytkownika i prototypów. Moje doświadczenie sugeruje, że zespoły wolą tego rodzaju artefakty niż wytyczne dotyczące interfejsu użytkownika przedstawione w formie graficznej.



Rysunek 3.4. Wymaganie niefunkcjonalne sformułowane jako ograniczenie

Zarządzanie wymaganiami niefunkcjonalnymi

W trakcie zarządzania wymaganiami niefunkcjonalnymi najlepiej jest wprowadzić rozróżnienie na wymagania globalne i lokalne. Te pierwsze zwykle odnoszą się do wszystkich wymagań funkcjonalnych i tworzą małą grupę. Przykładem może być ograniczenie wydajnościowe pokazane na rysunku 3.4. Globalne wymagania niefunkcjonalne powinny być jak najwcześniej szczegółowo opisane — w trakcie tworzenia wizji lub dodawania nowych elementów do rejestru produktu. Odkrywanie i dopracowywanie ich na zbyt późnym etapie może mieć negatywny wpływ na podejmowanie wyborów i odniesienie przez produkt sukcesu. Globalne wymagania niefunkcjonalne mogą być zapisane w specjalnym miejscu rejestru produktu, jak przedstawiono w tabeli 3.3.

Tabela 3.3. Przykładowy rejestr produktu z wymaganiami niefunkcjonalnymi

Wymagania funkcjonalne				Wymagania niefunkcjonalne
Motyw	Wymagania ogólne	Wymagania szczegółowe	Wysiłek	
E-mail	Tworzenie e-maila	Jako przedsiębiorca chcę mieć możliwość podania tematu e-maila	1	Produkt musi odpowiedzieć na zapytanie w czasie poniżej jednej sekundy

Warto również umieścić globalne wymagania niefunkcjonalne w definicji pojęcia „gotowe”. W konsekwencji każdy przyrost produktu musi spełniać te same wymagania.

Z drugiej strony, lokalne wymagania niefunkcjonalne dotyczą jedynie specyficznych wymagań funkcjonalnych, na przykład konkretnych wymagań wydajnościowych niezbędnych do odzyskiwania informacji. Jeżeli wymagania niefunkcjonalne wyrażone są za pomocą ograniczeń, możesz dołączyć je do historii, jak sugerują James Newkirk i Robert C. Martin [2001] oraz Cohn [2004]. Może to być rozwiązane za pomocą adnotacji.

Skalowanie rejestru produktu

Duże projekty niosą nowe wyzwania. Jednym z nich jest skalowanie rejestru produktu. Aby odnieść sukces, stwórz jeden rejestr produktu, zastosuj działania porządkowe na szeroką skalę, a także weź pod uwagę specyficzny punkt widzenia zespołu.

Wykorzystaj jeden rejestr produktu

W trakcie pracy nad dużym projektem Scrum upewnij się, że istnieje tylko jeden rejestr produktu zawierający wszystkie elementy niezbędne, by powołać produkt do życia. Unikaj rejestrów specyficznych dla zespołu lub komponentu, które rozkładają wymagania produktu na podsystemy lub komponenty. Generują one znaczące straty, ponieważ wszystkie składowe pochodzą z rejestru produktu; muszą być one również porządkowane i synchronizowane. Postaraj się przydzielać zespołom zadania wprost z rejestru produktu, dając priorytet zespołom tworzącym właściwości przed zespołami tworzącymi komponenty, jak opisałem w rozdziale 1. Darin Fisher, jeden z inżynierów pracujących przy projekcie przeglądarki Chrome, przedstawia kroki, jakie podjął Google, by umożliwić projektowi pracę z jednego rejestru produktu: „Jeśli chodzi o wymagania, wiele decyzji było podejmowanych

w trakcie zebrań z burzą mózgów, tak żeby zespół mógł omówić właściwości. Mieliśmy również otwartą listę mailingową, poprzez którą ludzie mogli dzielić się interesującymi pomysłami. Staraliśmy się skupić na właściwościach i ich minimalistycznej formie. Następnie udostępniliśmy listę całemu zespołowi, a pracownicy sami wybierali to, nad czym chcą pracować”⁷.

Działania porządkowe na szeroką skalę

Elementy rejestru produktu w dużych projektach Scrum są wciąż rozkładane na mniejsze i udoskonalane dokładnie na czas. Ale skala porządkowania się zmienia. Zamiast skupiać się na następnym sprincie, przygotowując rejestr produktu w dużym projekcie, należy myśleć o dwóch lub trzech sprintach naprzód, o czym powiem w rozdziale 4. W konsekwencji szczegółowy rejestr produktu w dużych projektach Scrum jest większy niż ten w mniejszych.

Uwzględnienie odrębnych spojrzeń na rejestr

Duże zwinne projekty, które zawierają wiele zespołów tworzących właściwości, mogą wiele zyskać, wykorzystując odrębne spojrzenia na rejestr produktu [Cohn, 2009, 330 – 331]. Każde spojrzenie pokazuje zestaw rejestru produktu. Jeżeli zespół tworzący właściwości pracuje w kolejnych sprintach nad motywem organizera, spojrzenie zespołu na rejestr składa się z odpowiednich zestawów. Spojrzenia mogą powodować konflikty pomiędzy kilkoma właścicielami produktu oraz zespołami pracującymi nad tym samym rejestrem produktu.

⁷ Wywiad z Darinem Fisherem przeprowadzony przez Colleen Frye na stronie *SearchSoftwareQuality.com*, data dostępu: 1 października 2008 roku.

Powszechne błędy

Mimo że rejestr produktu jest bardzo prostym narzędziem, korzystanie z niego może być trudne. Pamiętaj o występujących powszechnie błędach: specyfikacji wymagań wyglądającej jak rejestr, liście życzeń do Świętego Mikołaja, spychaniu wymagań na zespół, zaniedbywaniu porządkowania rejestru oraz tworzeniu dla jednego zespołu kilku rejestrów.

Ukryte specyfikacje wymagań

Specyfikacje wymagań udające rejestr produktu są jak diabeł w przebraniu: ładne, schludne i perfekcyjne. Bywają kuszące, ponieważ odwołują się do naszych pragnień poznania wszystkich wymagań od razu. Taka sytuacja ma jednak ciemną stronę. Rejestr produktu, który ma zbyt wiele szczegółów i jest zbyt złożony, nie wspiera spontanicznego powstawania wymagań. Nie postrzega wymagań jako płynnych i przelotnych, a raczej widzi je jako ustalone i ostateczne; zamraża wszystkie decyzje dotyczące zaspokajania potrzeb klientów na wczesnym etapie projektu.

Specyfikacja wymagań przebrana za rejestr produktu jest objawem niezdrowego związku pomiędzy właścicielem produktu a zespołem. Jeżeli natkniesz się na taki rejestr, dowiedz się, czy wizja produktu jest dostępna. Jeżeli tak, opracuj na jej podstawie nowy rejestr produktu i pozbądź się ukrytych specyfikacji wymagań. Jeżeli wizja produktu nie istnieje, stwórz ją. Możesz oczywiście brnąć dalej, siłować się z rejestrem, wydobywać motywy, przepisywać elementy jako *user stories* oraz zmagać się z przypisywaniem priorytetów poszczególnym elementom. Istnieje jednak marne prawdopodobieństwo, że uda Ci się stworzyć dobry produkt.

Lista życzeń do Świętego Mikołaja

Rejestr produktu, który przypomina dziecięcą listę życzeń do Świętego Mikołaja, zawiera wszystko i nic. Taki rejestr nie jest już listą niezrealizowanych prac — jest to baza wymagań. Na takiej liście bardzo trudno jest ustalić priorytety, ogranicza ona również możliwości rozwojowe produktu na podstawie opinii użytkowników i klientów, ponieważ zostało zidentyfikowanych zbyt dużo funkcjonalności. Aby określić, które elementy są niezbędne dla stworzenia dobrego produktu, użyj idei produktu lub jego wizji. Resztę wyrzuć.

Określanie wymagań

Czasem właściciel produktu sam tworzy rejestr produktu, dostarczając zespołowi gotowy produkt w trakcie zebrania planującego sprint. Takie podejście wzmacnia stary podział na „nich” i „nas”: właściciel produktu z jednej strony, zespół z drugiej. W ten sposób marnują się wiedza, doświadczenie i kreatywność zespołu, a planowanie sprintu staje się trudniejsze. Upewnij się, że właściciel produktu zawsze angażuje zespół scrumowy w prace porządkowe. Zaplanuj w każdym sprincie jedno lub kilka spotkań warsztatowych służących do uporządkowania rejestru, zaproś na nie cały zespół scrumowy i przypomnij poszczególnym członkom o zarezerwowaniu na te czynności czasu w każdym sprincie. Zawsze pamiętaj o jednej z mantr *Manifestu Agile*, dotyczącej współpracy: „Ludzie biznesu i deweloperzy muszą ze sobą blisko współpracować codziennie w trakcie trwania całego projektu” [Beck i in., 2001].

Zaniedbywanie porządkowania

Większość zebrań planujących sprint, w których brałem udział, była zabawna. Te, które nie były, miały problem z zaniedbanym rejestrem produktu. Kiedy rejestr nie zostanie uporządkowany przed spotkaniem, właściciel produktu i zespół starają się wykonać te czynności naprędce, co pochłania cenny czas i powoduje powstawanie słabych wymagań i zobowiązań bez przekonania. Dodatkowo pod koniec zebrania

wszyscy są wykończeni. Pamiętaj, że jeśli rejestr produktu nie został prawidłowo uporządkowany, następny sprint nie powinien się rozpocząć. Powinien zostać zawieszony do czasu poprawnego przygotowania rejestru.

Uzupełnianie rejestrów

Mój klient miał kiedyś pięciu właścicieli produktu pracujących z jednym zespołem. Ponieważ każdy z nich chciał, by jak największa ilość pracy została zrobiona jak najszybciej, zespół został poproszony o wykorzystywanie wszystkich pięciu rejestrów w każdym sprincie. To trochę uspokoiło właścicieli produktu; wiedzieli oni, że ich wymagania będą uwzględnione w pracy. Sprawilo to również, że byli bardzo nieusatysfakcjonowani: prace znacznie się przeciągały. Zajmowanie się kilkoma produktami jednocześnie może wyglądać dobrze — każdy jest zajęty, toczy się praca nad każdym elementem. Ale nie ma szybkiego postępu. Zespół nie ma spójnego celu sprintu i traci cenny czas na manewrowanie pomiędzy zadaniami.

Jeżeli Twój zespół musi pracować nad kilkoma rejestrami, upewnij się, że każdy sprint poświęcony jest tylko jednemu produktowi. Byłoby jeszcze lepiej, gdyby zespół pracował nad jednym produktem przez kilka sprintów, tak by wydanie mogło nastąpić szybciej. Dopiero później należy przejść do następnego produktu. To podejście wymaga ustalenia priorytetów produktów i określenia procesów zarządzania portfolio. Sprawa mojego klienta trafiła w końcu do dyrektora generalnego firmy, który chciał mieć wszystko zrobione „na wczoraj” i trudno mu było ustalać priorytety, jakie prowadziłyby właścicieli produktu.

Spostrzeżenia

Uwierz w swoją kreatywność i pozwól, by elementy rejestru produktu pojawiały się spontanicznie. Rejestr powinien być prosty i spójny. Skoncentruj się na elementach, które są niezbędne do powstania

produktu. Bądź odważny i pozbywaj się elementów niepotrzebnych. W zastosowaniu koncepcji omówionych w tym rozdziale pomogą Ci odpowiedzi na poniższe pytania.

W jaki sposób w Twojej firmie odkrywa się i opisuje wymagania?

Czy Twój projekt ma cechy DEEP?

W jaki sposób porządkujesz swój rejestr produktu?

Co należałoby zrobić, by wspólnie odkrywać i opisywać wymagania w każdym sprincie?

Jak sobie radzisz z wymaganiami нефunkcjonalnymi? Kiedy i w jaki sposób się je definiuje?

Skorowidz

A

Apple, 55, 59, 62, 74, 88, 114
atrybuty produktu, 64

B

błędy, 46
Brzytwa Ockhama, 62
budżet, 112

C

Chief Product Owner, Patrz
główny właściciel produktu
Chrome, 104
Chucka Close, 135
cykle kwartalne, 118
czas, 112

D

Darin Fisher, 104
data premiery wydania, 112
dekompozycja elementów, 95
dekompozycja epików, 96

E

efektywna wizja produktu, 54
epiki, 85, 141
Experticy, 58, 59

F

FDA, 118
funkcjonalność, 112

G

główny właściciel produktu, 41
Google, 60, 63, 66, 89, 116, 118

H

hierarchia właścicieli produktu,
42, 43

I

interfejs użytkownika, 63

J

John Maeda, 62

K

Ken Schwaber, 30
klarowność, 97
komitet
właścicieli produktu, 50
koszt, 112

M

makiety, 68
mapa produktu, 72
marketer produktu, 40
menedżer
produktu, 31
projektu, 31, 40
metodologia Scrum, 15
minimalna wersja produktu,
58, 59, 73
mistrz młyna, 36, 82

mnożniki dla szybkości, 126

mobile.de, 37

Model Kano, 71

funkcje wydajnościowe, 71

funkcjonalności

uszczęśliwiające, 71

podstawowe funkcje, 71

możliwość testowania, 97

O

odległy właściciel produktu, 48

Office, 73

ogólne historie, Patrz epiki

określanie wymagań, 107

opakowanie pogładowe, 70

P

paraliż analityczny, 76

pasywny właściciel produktu, 146

persony, 69

plan wydania, 124, 127

planowanie

sprintu, 136, 143

wydania, 111

przyszłościowe, 129

pojęcie „gotowe”, 137

poker planistyczny, 99

Polycom, 53

potrzeby klientów, 64

prawo Brooksa, 114

primus inter pares, 33

prognozowanie szybkości, 126

prostopoda, 61

prototypy, 68
przegląd sprintu, 140, 144
przepracowanie, 47
punkty, 98

R

refaktoryzacja architektury, 74
rejestr produktu, 79, 86, 104, 124
cechy, 80
nowo powstający, 81
odkrywanie elementów, 84
opisywanie elementów, 85
porządkowanie, 81
skalowanie, 104
szacunkowy, 81
ustalanie struktury, 86
wystarczająco
szczegółowy, 80
zawierający priorytety, 81
rejestr sprintu, 139
ustalanie priorytetów, 87
retrospekcja sprintu, 142, 145
Robert G. Cooper, 58
rola
właściciela produktu, 29
Roman Pichler, 27
ryzyko, 89

S

Salesforce, 18, 113, 157
scenariusze, 69
scrum, 22, 67, 72, 79, 82, 124,
146, 144
skalowanie roli właściciela
produktu, 41
smartfon, 59
spalanie
belka spalania, 122
wydania, 121
sponsoring, 154

sprint, 92, 93, 115, 119, 135
planowanie, 136
wybór celu, 92
systematyzacja, 130
szkic produktu, 54
szybkość, 119

T

techniki facylitacji, 33
Theodore Levitt, 39
trener, 154

U

ukryte specyfikacje wymagań,
106
uprawnienia, 46
user experience, 35, 59, 63, 68
user stories, 68, 152, 157

V

Visio, 73

W

waga roli, 156
warianty produktu, 73
wartość, 88
wielkość elementów, 98
William Ockham, 62
wizja produktu, 54, 72, 112
cechy, 55
krótka i zwięzła, 57
obszerna i angażująca, 56
techniki tworzenia, 68
wspólna i jednocząca, 56
wizje profetyczne, 75
właściciel produktu, 15, 18, 30,
32, 34, 35, 45, 46, 82, 87,
132, 140, 151, 152, 156

o podzielonych
obowiązkach, 48
o zbyt małych
uprawnieniach, 46
odległy, 48
przepracowany, 47
zastępczy, 49

wspólna linia bazowa dla
szacunków, 129

wydania
częste, 115

wydanie
wczesne, 115
wykonalność, 97
wykres spalania, 131, 139, 148
wydania, 120
wymagania funkcjonalne, 91
wymagania niefunkcjonalne,
101, 102, 103

Z

zakresy punktów, 99
zależności, 91
zamrożona jakość, 115
zasłona dymna, 147
zastępczy właściciel produktu, 49
zdatność do wypuszczenia na
rynek, 90
zebrania scrumowe, 138, 144
zespół, 36, 82, 85, 87, 88, 99,
114, 119
złożona hierarchia właścicieli
produktu, 44
zmiennie tempo pracy, 147
znikający właściciel produktu,
146
zrównoważone tempo, 137
zwinne zarządzanie produktem,
22

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ 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 WYDAWNICZA

 **Helion SA**

SKUTECZNIE ZARZĄDZAJ PROJEKTAMI!

Współczesne projekty wymagają niezwyklej elastyczności i błyskawicznego dostosowywania się do panujących warunków. Czasy ogromnych projektów, gdy klient przez wiele miesięcy czekał na produkt, odchodzą w niepamięć. W zwinnych metodologiach zarządzania główne jest regularne dostarczanie kolejnych wersji produktu w krótkich odstępach czasu. Dzięki temu na bieżąco kontrolowany jest kierunek rozwoju, a ewentualna korekta nie przysparza problemów. Już teraz poznaj kluczowe zasady zwinnego zarządzania projektami!

W trakcie lektury tej książki poznasz jedną z najpopularniejszych metodyk — Scrum. Dowiesz się, jakie role definiuje Scrum i jaki jest zakres obowiązków wszystkich osób związanych z projektem. Nauczysz się tworzyć wizję produktu, pracować z jego rejestrem oraz planować wydanie. Podstawowym pojęciem w metodyce Scrum jest sprint. Poznaj jego specyfikę, zasady prowadzenia oraz techniki kontrolowania postępów prac. Koniecznie zwróć uwagę na najczęściej popełniane błędy. Dostarcz produkt na czas, poczuć satysfakcję i odnieś sukces — to się opłaci!

Dzięki tej książce:

- poznasz role w metodologii Scrum
- zaplanujesz sprint
- będziesz skutecznie zarządzać rejestrem produktu
- dostarczysz produkt na czas!

helion.pl
księgarnia internetowa

Nr katalogowy: 17584



Księgarnia internetowa:
<http://helion.pl>



Zamówienia telefoniczne:
0 801 339900
0 601 339900

 **Addison-Wesley**
Pearson Education



Helion

Sprawdź najnowsze promocje:
• <http://helion.pl/promocje>
Książki najchętniej czytane:
• <http://helion.pl/bestsellery>
Zamów informacje o nowościach:
• <http://helion.pl/newosci>

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

sięgnij po **WIĘCEJ**



KOD KORYZYSKI

cena: 39,00 zł

ISBN 978-83-246-8526-4



Informatyka w najlepszym wydaniu