

SCRUM

PRAKTYCZNY PRZEWODNIK
PO NAJPOPULARNIEJSZEJ
METODYCE AGILE

KENNETH S. RUBIN



Metodyki zwinne — Twój klucz do sukcesu!

Tytuł oryginału: Essential Scrum: A Practical Guide to the Most Popular Agile Process

Tłumaczenie: Janusz Grabis

ISBN: 978-83-8322-617-0

Authorized translation from the English language edition, entitled: ESSENTIAL SCRUM: A PRACTICAL GUIDE TO THE MOST POPULAR AGILE PROCESS; ISBN 0137043295; by Kenneth S. Rubin; published by Pearson Education, Inc, publishing as Addison Wesley. Copyright © 2013 Pearson Education, Inc.

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, 2023.

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.

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 230 98 63

e-mail: helion@helion.pl

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

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

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

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 — Mike Cohn	21
	Słowo wstępne — Ron Jeffries	23
	Wstęp	25
Rozdział 1	Wprowadzenie	33
	Czym jest Scrum?	34
	Początki Scruma	35
	Dlaczego Scrum?	36
	Wyniki Genomiki	37
	Czy Scrum może pomóc Tobie?	37
	Domena złożona	40
	Domena skomplikowana	40
	Domena prostoty	40
	Domena chaosu	41
	Nieporządek	41
	Praca sterowana przerwaniem	41
	Zakończenie	42
CZĘŚĆ I	POJĘCIA PODSTAWOWE	
Rozdział 2	Środowisko Scrum	45
	Wprowadzenie	45
	Role w Scrumie	46
	Właściciel produktu	47
	Mistrz młyna	47
	Zespół deweloperski	48
	Aktywności i artefakty Scruma	48
	Rejestr produktu	50
	Sprinty	52
	Planowanie sprintu	52

Wykonanie sprintu	54
Codienne działania scrumowe	55
Koniec pracy	56
Przegląd sprintu	57
Retrospekcja sprintu	58
Zakończenie	59

Rozdział 3 Zasady zwinności 61

Wprowadzenie	61
Zmienność i niepewność	64
Wspieraj pomocną zmienność	64
Stosuj budowanie w sposób iteracyjny i przyrostowy	65
Wykorzystuj zmienność poprzez inspekcję, adaptację i przejrzystość	66
Redukuj wszystkie formy niepewności jednocześnie	67
Przewidywanie i adaptacja	68
Pozostaw wszystkie opcje otwarte	69
Pogódź się z tym, że nie wszystko da się przewidzieć z góry	69
Preferuj podejście odkrywcze i adaptacyjne	71
Wspieraj zmiany w sposób uzasadniony ekonomicznie	72
Równoważ pracę, którą można przewidzieć z góry, z pracą adaptacyjną w samą porę	74
Wiedza potwierdzona	75
Potwierdź bezzwłocznie istotne założenia	76
Wykorzystuj różne konkurencyjne ścieżki pozyskiwania wiedzy	76
Organizuj przepływ pracy umożliwiający szybkie pozyskiwanie informacji zwrotnej	77
Praca cząstkowa	78
Stosuj rozsądne ekonomicznie rozmiary zapotrzebowania	79
Zrób rozpoznanie inwentarza i zarządzaj nim w celu dobrego przepływu	80
Skup się na pracy czekającej na realizację, a nie na pracownikach czekających na pracę	81
Bierz pod uwagę koszt opóźnień	83
Postęp	84
Zadaptuj się w oparciu o napływające informacje i zmodyfikuj plan	84
Mierz postęp poprzez ocenę działających rzeczy	84
Skup się na dostarczaniu wartości	85
Wydajność	86
Działaj szybko, ale nie w pośpiechu	86
Buduj z zachowaniem jakości	86
Stosuj minimalną potrzebną ilość ceremonii	87
Zakończenie	88

Rozdział 4 Sprinty 91

Wprowadzenie	91
Ograniczenie czasowe	92
Ograniczenie pracy cząstkowej	92
Wymuszanie priorytetów	93
Demonstrowanie postępów	93
Unikanie zbędnego perfekcjonizmu	93
Motywowanie domykania	94
Poprawianie przewidywalności	94

Krótki okres trwania	94
Łatwość planowania	94
Szybka informacja zwrotna	95
Lepsze zyski z inwestycji	95
Ograniczanie błędów	95
Rozbudzenie podekscytowania	95
Regularne punkty kontrolne	96
Stały czas trwania	97
Zalety taktowania	97
Uprozczone planowanie	98
Niezmiennosc celu	98
Czym jest cel sprintu?	99
Wzajemne zobowiązanie	99
Zmiana kontra doprecyzowanie	99
Konsekwencje zmiany	100
Pragmatyczność	101
Zakończenie przed czasem	102
Definicja ukończenia	103
Czym jest definicja ukończenia?	104
Definicja ukończenia może ewoluować	106
Definicja ukończenia kontra kryteria akceptacji	107
Ukończony kontra rzeczywiście ukończony	107
Zakończenie	108

Rozdział 5 Wymagania i historyjki użytkownika 109

Wprowadzenie	109
Wykorzystanie dyskusji	111
Stopniowe udoskonalanie	112
Czym są historyjki użytkownika?	113
Karta	113
Rozmowa	114
Potwierdzenie	115
Poziom szczegółowości	116
Inwestuj w dobre historyjki	118
Niezależność	118
Negocjowalność	118
Wartościowość	120
Ocenialność	121
Dobry (mały) rozmiar	121
Testowalność	122
Wymagania niefunkcjonalne	122
Historyjki pozyskiwania wiedzy	123
Zbieranie historyjek	124
Warsztaty pisania historyjek	125
Mapa historyjek	125
Zakończenie	127

Rozdział 6	Rejestr produktu	129
	Wprowadzenie	129
	Elementy rejestru produktu	129
	Cechy dobrego rejestru produktu	131
	Odpowiedni stopień uszczegółowienia	131
	Emergencja	132
	Przypisanie ocen	132
	Przypisanie priorytetów	133
	Pielęgnacja	134
	Czym jest pielęgnacja?	134
	Kto pielęgnuje rejestr?	135
	Kiedy odbywa się pielęgnacja?	136
	Definicja gotowości	138
	Zarządzanie przepływem	139
	Zarządzanie przepływem wersji dystrybucyjnych	139
	Zarządzanie przepływem sprintu	140
	Jakie rejestry produktu i w jakiej liczbie?	142
	Czym jest produkt?	142
	Duże produkty — hierarchiczne rejestry produktu	143
	Wiele zespołów — jeden rejestr produktu	144
	Jeden zespół — wiele produktów	145
	Zakończenie	146
Rozdział 7	Nadawanie ocen i prędkość	147
	Wprowadzenie	147
	Co i kiedy oceniamy	148
	Ocena elementów z rejestru portfela	149
	Ocena elementów z rejestru produktu	149
	Ocena zadań	150
	Konceptje nadawania ocen elementom rejestru produktu	150
	Nadawanie ocen jako zespół	151
	Oceny nie są zobowiązaniem	152
	Dokładność kontra precyzja	153
	Ocenianie w sposób względny	153
	Jednostki służące do oceny	155
	Punkty historyjkowe	155
	Idealne dni	156
	Planowanie pokerowe	156
	Skala ocen	157
	Zasady gry	158
	Zalety	160
	Czym jest prędkość?	160
	Obliczanie przedziału prędkości	161
	Prognozowanie prędkości	162
	Wpływanie na prędkość	163
	Nieprawidłowe korzystanie z prędkości	164
	Zakończenie	165

Rozdział 8	Dług techniczny	167
	Wprowadzenie	167
	Konsekwencje długu technicznego	169
	Nieprzewidywalny punkt przęgięcia	169
	Wzrost czasu potrzebnego do dostarczenia wersji	169
	Znaczna liczba błędów	170
	Rosnące koszty produkcji i wsparcia	170
	Umieranie produktu	171
	Spadek przewidywalności	171
	Działanie poniżej optymalnej wydajności	172
	Powszechna frustracja	172
	Spadek satysfakcji klientów	172
	Przyczyny długu technicznego	172
	Presja nieprzekroczenia terminu końcowego	173
	Próby sztucznego zwiększenia prędkości	173
	Mit: Rezygnacja z testowania zwiększa prędkość	174
	Dług narasta na długu już istniejącym	174
	Długiem technicznym trzeba zarządzać	176
	Zarządzanie przyrostem długu technicznego	177
	Stosowanie dobrych praktyk technicznych	177
	Solidna definicja ukończenia	178
	Prawidłowe rozumienie ekonomii długu technicznego	178
	Ujawnianie długu technicznego	180
	Ujawniaj dług techniczny na poziomie biznesowym	181
	Ujawniaj dług techniczny na poziomie inżynierskim	182
	Obsługa długu technicznego	183
	Nie każdy dług techniczny powinien być spłacany	184
	Zastosuj metodę skauta (obsłuż dług, kiedy go napotkasz)	186
	Spłacaj dług techniczny przyrostowo	186
	Spłacaj w pierwszej kolejności dług o najwyższych odsetkach	187
	Spłacaj dług techniczny podczas wykonywania pracy mającej wartość dla klienta	187
	Zakończenie	189

CZĘŚĆ II **ROLE**

Rozdział 9	Właściciel produktu	193
	Wprowadzenie	193
	Główne obowiązki	194
	Zarządzanie ekonomią	195
	Udział w planowaniu	196
	Pielęgnacja rejestru produktu	197
	Definiowanie kryteriów akceptacji i weryfikowanie, iż zostały one spełnione	197
	Współpraca z zespołem deweloperskim	198
	Współpraca z interesariuszami	199
	Cechy i umiejętności	199
	Umiejętności domenowe	199
	Umiejętności interpersonalne	200

Podejmowanie decyzji	201
Odpowiedzialność	201
Dzień z życia właściciela produktu	202
Kto powinien być właścicielem produktu?	204
Wewnętrzne prace deweloperskie	205
Komercyjne prace deweloperskie	205
Outsourcing	207
Tworzenie komponentów	208
Właściciel produktu pełniący inne role	208
Zespół właścicieli produktu	209
Pełnomocnicy właściciela produktu	210
Szef właścicieli produktu	211
Zakończenie	211

Rozdział 10 Mistrz młyna 213

Wprowadzenie	213
Główne obowiązki	213
Trener	213
Lider służby	214
Autorytet w dziedzinie procesu	215
Ochrona przed zakłóceniami	215
Usuwanie przeszkód	215
Agent zmiany	215
Cechy i umiejętności	216
Wiedza	216
Umiejętność stawiania pytań	216
Cierpliwość	217
Współpraca	217
Proaktywność	217
Przezroczystość	217
Dzień z życia	218
Wypełnianie roli	219
Kto powinien być mistrzem młyna?	219
Czy mistrz młyna to zajęcie na pełny etat	220
Połączenie roli mistrza młyna z innymi rolami	220
Zakończenie	221

Rozdział 11 Zespół deweloperski 223

Wprowadzenie	223
Zespoły o specyficznej roli	223
Główne obowiązki	224
Wykonanie sprintu	225
Codzienna inspekcja i adaptacja	225
Pielęgnacja rejestru produktu	225
Planowanie sprintu	225
Inspekcja i adaptacja produktu oraz procesu	225

Cechy i umiejętności	226
Samoorganizacja	226
Różnorodny pod względem umiejętności i samowystarczalny	228
Umiejętności typu T	229
Postawa muszkietierów	231
Komunikacja szerokopasmowa	232
Przezroczysta komunikacja	233
Prawidłowy rozmiar	233
Skupienie i poświęcenie	234
Praca w podtrzymywalnym tempie	236
Długotrwałość	237
Zakończenie	238
Rozdział 12 Budowa zespołów scrumowych	239
Wprowadzenie	239
Zespoły budujące cechy kontra zespoły budujące komponenty	240
Koordynacja wielu zespołów	244
Scrum scrumów	244
Pociąg wersji dystrybucyjnych	246
Zakończenie	249
Rozdział 13 Menedżerowie	251
Wprowadzenie	251
Kształtowanie zespołów	253
Definiowanie granic	253
Dostarczanie jasnego motywującego celu	254
Formowanie zespołów	254
Zmiana kompozycji zespołów	255
Upoważnianie zespołów	256
Pielęgnacja zespołów	257
Motywowanie ludzi	257
Rozwijanie umiejętności	257
Przewodzenie obszarowi funkcjonalności	258
Utrzymywanie integralności zespołu	258
Przystosowywanie i adaptowanie środowiska	259
Promowanie wartości zwinności	259
Usuwanie przeszkód organizacyjnych	259
Przystosowywanie grup wewnętrznych	260
Przystosowywanie partnerów	260
Zarządzanie przepływem wytwarzania wartości	261
Spoglądanie z perspektywy systemu	261
Zarządzanie ekonomią	261
Obserwacja pomiarów i raportowanie	261
Menedżerowie projektów	262
Obowiązki menedżera projektu w zespole scrumowym	263
Pozostanie przy roli niezależnego menedżera projektu	264
Zakończenie	267

CZĘŚĆ III PLANOWANIE

Rozdział 14 Zasady planowania w Scrumie	273
Wprowadzenie	273
Nie zakładaj, że z góry uda Ci się wszystko dobrze zaplanować	273
Planowanie z góry powinno być pomocne, ale nie przesadne	274
Pozostaw opcje planowania otwartymi do ostatniego rozsądnego momentu	275
Skup się bardziej na adaptowaniu i ponownym planowaniu niż na trzymaniu się już ustalonego planu	275
Prawidłowo zarządzaj inwentarzem pracy	277
Preferuj mniejsze i częstsze wersje dystrybucyjne	278
Planuj szybki proces nauki i wykonywanie zwrotów, kiedy jest to niezbędne	280
Zakończenie	280
Rozdział 15 Planowanie wielopoziomowe	281
Wprowadzenie	281
Planowanie portfela	283
Planowanie produktu (tworzenie wizji produktu)	283
Wizja	283
Rejestr produktu wysokiego poziomu	283
Mapa drogowa produktu	284
Planowanie wersji dystrybucyjnej	285
Planowanie sprintu	287
Planowanie codzienne	287
Zakończenie	288
Rozdział 16 Planowanie portfela	291
Wprowadzenie	291
Czas	291
Uczestnicy	292
Proces	292
Strategie tworzenia harmonogramu	293
Optymalizacja pod względem zysków z cyklu życia	294
Wyznaczanie kosztu opóźnienia	295
Staraj się obliczać dokładnie, ale nie precyzyjnie	298
Strategie napływu	299
Zastosowanie filtra ekonomicznego	299
Równoważ tempo przybywania z tempem ubywania	300
Szybko wykorzystuj nadarzające się okazje	302
Planuj mniejsze i częstsze wersje dystrybucyjne	303
Strategie odpływu	304
Skup się na pracy czekającej na realizację, nie na pracownikach czekających na pracę	304
Ustal limit pracy cząstkowej	305
Zaczekaj na cały zespół	306
Strategie aktywności	306
Użycie ekonomii końcowej	307
Zakończenie	308

Rozdział 17 Planowanie produktu (tworzenie wizji produktu)	309
Wprowadzenie	309
Czas	310
Uczestnicy	310
Proces	312
Przykład ZODC	312
Tworzenie wizji	313
Tworzenie rejestru produktu wysokiego poziomu	316
Definiowanie mapy drogowej produktu	317
Inne aktywności	319
Przeprowadzanie wizji w sposób ekonomiczny	321
Celuj w realistyczny poziom ufności	322
Skup się na krótkim horyzoncie	323
Działaj szybko	324
Płać za wiedzę potwierdzoną	324
Stosuj przyrostowe lub prowizoryczne finansowanie	325
Ucz się szybko i wykonuj zwroty (czyli stosuj strategię „szybkiej porażki”)	326
Zakończenie	327
Rozdział 18 Planowanie wersji dystrybucyjnej (planowanie długoterminowe)	329
Wprowadzenie	329
Czas	330
Uczestnicy	331
Proces	331
Ograniczenia wersji dystrybucyjnej	333
Wszystko sztywne	333
Ustalony zakres i czas	334
Ustalony zakres	335
Ustalony czas	335
Zmienna jakość	336
Uaktualnianie ograniczeń	336
Pielęgnacja rejestru produktu	337
Definiowanie minimalnego zestawu cech kwalifikującego do dystrybucji	338
Tworzenie mapy sprintów (umieszczanie elementów rejestru produktu w slotach)	339
Planowanie wersji z ustaloną datą	341
Planowanie wersji z ustalonym zakresem	345
Obliczanie kosztów	347
Komunikacja	348
Komunikowanie postępów dla wersji z ustalonym zakresem	348
Komunikowanie postępów dla wersji z ustaloną datą	350
Zakończenie	351

CZĘŚĆ IV WYKONYWANIE SPRINTÓW

Rozdział 19 Planowanie sprintu	355
Wprowadzenie	355
Czas	355
Uczestnicy	356
Proces	356
Podejścia do planowania sprintu	358
Dwuczęściowe planowanie sprintu	358
Jednoczęściowe planowanie sprintu	359
Określanie pojemności	360
Czym jest pojemność?	360
Pojemność w punktach historyjkowych	361
Pojemność w roboczogodzinach	362
Wybieranie elementów rejestru produktu	363
Nabieranie pewności	364
Doprecyzowywanie celu sprintu	365
Ostateczne ustalenie zobowiązania	365
Zakończenie	366
Rozdział 20 Wykonanie sprintu	367
Wprowadzenie	367
Czas	367
Uczestnicy	367
Proces	368
Planowanie wykonania sprintu	369
Zarządzanie przepływem	369
Praca równoległa i działanie w kopcu	370
Którą pracę rozpocząć?	372
Jak zorganizować pracę nad zadaniami?	372
Jaką pracę trzeba zrealizować?	373
Kto wykonuje pracę?	373
Codzienne działania scrumowe	374
Realizacja zadań — praktyki techniczne	374
Komunikowanie	375
Tablica zadań	375
Wykres spalania sprintu	376
Wykres rozpalania sprintu	379
Zakończenie	380
Rozdział 21 Przegląd sprintu	381
Wprowadzenie	381
Uczestnicy	382
Przygotowanie	383
Wskazanie, kogo zaprosić na przegląd	384
Ustalenie harmonogramu	384

Potwierdzenie, że praca w sprincie została zrealizowana	385
Przygotowanie do demonstracji	386
Wyznaczenie, kto co robi	386
Podejście	386
Podsumowanie	387
Demonstracja	388
Dyskusja	388
Adaptacja	389
Problemy przeglądu sprintu	389
Zatwierdzanie	390
Mała frekwencja	390
Duże projekty	391
Zakończenie	391

Rozdział 22 Retrospekcja sprintu 393

Wprowadzenie	393
Uczestnicy	395
Przygotowania	396
Zdefiniowanie punktu skupienia retrospekcji	396
Wybór ćwiczeń	397
Zebranie obiektywnych danych	397
Dopasowanie szczegółów retrospekcji	398
Podejście	398
Przygotowanie odpowiedniej atmosfery	400
Ustalenie wspólnego kontekstu	400
Identyfikacja spostrzeżeń	403
Wskazanie działań	405
Zakończenie retrospekcji	408
Wprowadzenie zmian w życie	408
Problemy retrospekcji sprintu	408
Zakończenie	411

Rozdział 23 Co dalej? 413

Nie ma stanu końcowego	413
Odkryj swoją własną ścieżkę	414
Stosowanie najlepszych praktyk	414
Używanie Scruma do odkrywania ścieżki naprzód	415
Naprzód!	416

Słowniczek 419

Bibliografia 443

Skorowidz 447

Rozdział 4

SPRINTY

Scrum organizuje pracę w iteracje lub cykle — o długości nie przekraczającej miesiąca — nazywane sprintami. Ten rozdział opisuje dokładniej, czym są sprinty. Poruszę tutaj kilka kluczowych charakterystyk sprintów: są ograniczone czasowo — mają jednakowy, krótki okres trwania, mają swój cel, który nie powinien być zmieniany po rozpoczęciu, i muszą osiągnąć stan końca zgodnie ze zdefiniowanymi przez zespół kryteriami końca pracy.

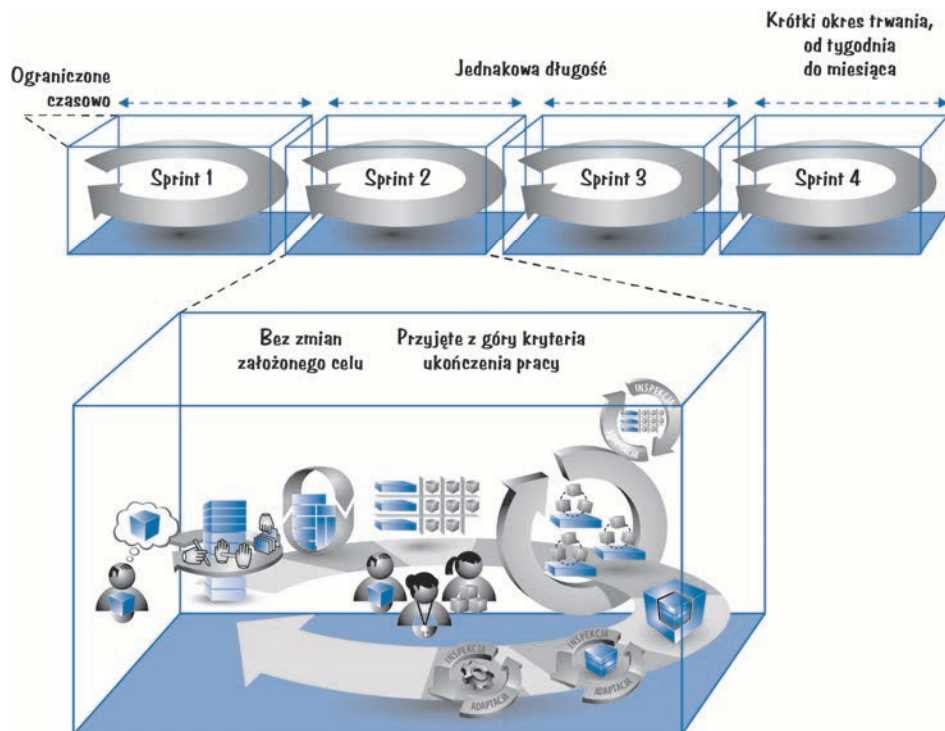
Wprowadzenie

Sprinty stanowią szkielet środowiska Scrum (patrz rysunek 4.1).

Na rysunku 4.1 sprint jest reprezentowany przez dużą szarą strzałkę rozciągającą się od rejestru produktu poprzez pętlę wykonania sprintu i otaczającą członków zespołu scrumowego. Pozostałe artefakty i aktywności scrumowe pokazane są w punktach odpowiadających chwili ich wystąpienia w czasie trwania sprintu. Wykonanie sprintu jest często mylone ze sprintem jako takim, ale w rzeczywistości jest to tylko jedna z aktywności mających miejsce w trakcie trwania sprintu, razem z planowaniem sprintu, przeglądem sprintu i retrospekcją sprintu.

Wszystkie sprinty są ograniczone czasowo — mają ustalony czas rozpoczęcia i zakończenia. Ponadto sprinty muszą być krótkie — mogą trwać od tygodnia do miesiąca. Ich długość powinna być jednakowa, chociaż pod pewnymi warunkami dopuszcza się wyjątki. W trakcie trwania sprintu nie wolno zmieniać jego celu pod względem zakresu prac lub składu zespołu. Wynikiem sprintu jest potencjalnie zdalny do wdrożenia przyrost produktu spełniający kryteria ukończenia prac ustalone wcześniej przez zespół scrumowy.

Mimo że każda organizacja implementuje Scrum na swój sposób, powyższe kryteria, z kilkoma wyjątkami opisanymi dalej, powinny obowiązywać w każdym sprincie i zespole scrumowym. Przyjrzyjmy się im z bliska, aby zrozumieć, dlaczego mają właśnie taką, a nie inną postać.



RYСУNEK 4.1. Sprints — szkielet środowiska Scrum

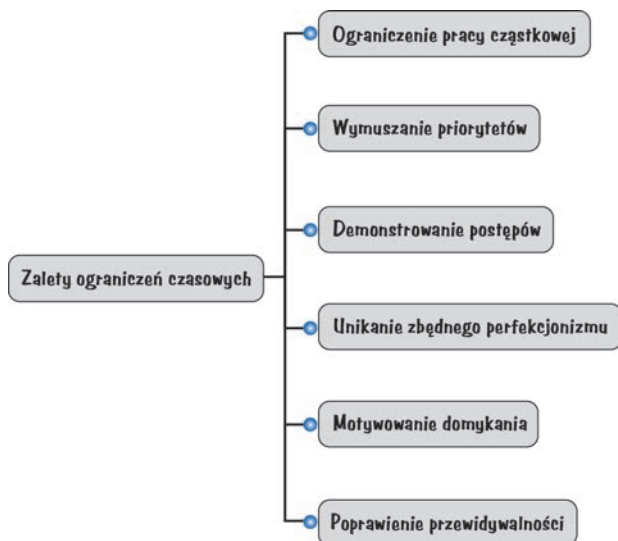
Ograniczenie czasowe

Sprints bazują na koncepcji **ograniczenia czasowego** — techniki zarządzania czasem, która pozwala panować nad wydajnością i zakresem pracy. Każdy sprint odbywa się w przedziale czasu mającym ustaloną datę rozpoczęcia i zakończenia, nazywanym pudełkiem czasu (ang. *timebox*). W tym czasie oczekuje się od zespołu pracy w podtrzymywalnym tempie, pozwalającej na ukończenie zadań będących w zgodzie z celem sprintu.

Ograniczenie czasowe jest istotne z kilku powodów (patrz rysunek 4.2).

Ograniczenie pracy częściowej

Ograniczenie czasowe jest techniką ograniczenia ilości pracy częściowej. Praca częściowa reprezentuje inwentarz pracy rozpoczętej, ale jeszcze nieukończonyj. Nieumiejętne zarządzanie tą pracą może mieć poważne konsekwencje finansowe. Sprint (ograniczenie czasowe) zapewnia nałożenie limitu na pracę częściową, ponieważ zespół zaplanuje wykonanie tylko tych elementów, które uważa za wykonalne w trakcie jego trwania.



RYSUNEK 4.2. Zalety ograniczeń czasowych

Wymuszanie priorytetów

Ograniczenie czasowe zmusza nas do ustalenia priorytetów i wykonania małej ilości pracy, która ma największe znaczenie. To skupia naszą uwagę na wyprodukowaniu czegoś wartościowego w szybkim tempie.

Demonstrowanie postępów

Ograniczenie czasowe pomaga nam zademonstrować postępy w pracach poprzez kończenie i weryfikowanie istotnych elementów wytwarzanego produktu do ustalonego dnia (zakończenia sprintu). Tego typu działanie redukuje ryzyko nieefektywnej organizacji pracy poprzez unikanie nierealnych form raportowania postępów w pracy, takich jak przestrzeganie planu. Ograniczenie czasowe pomaga również wykazywać postęp w pracach nad dużymi cechami, których ukończenie wymaga więcej niż jednego zakresu czasu. Każdy sprint daje poczucie pewności, iż osiągnięto mierzalny, wartościowy postęp w pracach, a także pozwala interesariuszom dowiedzieć się, co jeszcze trzeba zrobić, aby dostarczyć gotową cechę produktu.

Unikanie zbędnego perfekcjonizmu

Ograniczenie czasowe zapobiega zbędnemu perfekcjonizmowi. Każdemu z nas zdarzyło się nieraz spędzić zbyt wiele czasu, próbując zrobić coś w sposób „perfekcyjny” lub doprowadzić pewną cechę do stanu idealnego, w sytuacji, gdy stan dobry był w pełni satysfakcjonujący. Wprowadzając ustaloną datę zakończenia sprintu, czyli moment, kiedy musi zostać dostarczone dobre rozwiązanie, ograniczenie czasowe wymusza zakończenie prac, które teoretycznie można byłoby prowadzić w nieskończoność.

Motywowanie domykania

Ograniczenie czasowe zachęca do domykania prac. Ze swojego doświadczenia mogę powiedzieć, że praca jest faktycznie wykonywana, kiedy zespoły znają datę zakończenia zadania. Świadomość, że koniec sprintu niesie ze sobą nieprzekraczalny termin rozliczenia, zachęca członków zespołu do włożenia wysiłku w wykonanie pracy na czas. Bez takiej daty znika poczucie pilności zakończenia pracy.

Poprawianie przewidywalności

Ograniczenie czasowe poprawia przewidywalność. Chociaż nie jesteśmy w stanie przewidzieć precyzyjnie, jaką ilość pracy uda nam się wykonać w ciągu najbliższego roku, bez wątpliwości powinniśmy w miarę dokładnie przewidzieć zadania, które będziemy w stanie zrealizować w ciągu nadchodzącego krótkiego sprintu.

Krótki okres trwania

Krótkie okresy trwania sprintów przynoszą wiele korzyści (patrz rysunek 4.3).



RYСУNEK 4.3. Zalety krótkich sprintów

Łatwość planowania

Krótkie sprints ułatwiają planowanie. O wiele prościej jest zaplanować pracę na okres kilku tygodni niż na przykład na sześć miesięcy. Poza tym planowanie w tak krótkim horyzoncie czasowym wymaga o wiele mniej wysiłku i jest o wiele bardziej zgodne z rzeczywistością w porównaniu z planowaniem długoterminowym.

Szybka informacja zwrotna

Sprinty o krótkim czasie trwania pozwalają na szybkie uzyskanie informacji zwrotnej. Podczas każdego sprintu tworzymy działające oprogramowanie — możemy dokonać jego inspekcji, a następnie adaptacji tego, co budujemy i jak to budujemy. W ten sposób możemy szybciej wyeliminować ścieżki niekorzystne dla produktu lub samego procesu produkcji, zanim podejmiemy błędną decyzję, która otworzy drogę do kolejnych szkodliwych działań. Szybka informacja zwrotna pozwala nam również na wcześniejsze odkrywanie i wykorzystywanie pojawiających się krótkoterminowych okazji.

Lepsze zyski z inwestycji

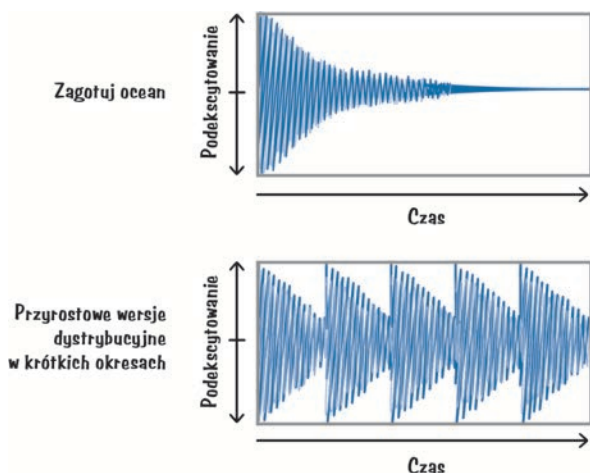
Sprinty o krótkim czasie trwania nie tylko poprawiają ekonomię poprzez szybką informację zwrotną, ale również pozwalają na wcześniejsze i bardziej regularne wypuszczanie kolejnych wersji oprogramowania. Dzięki temu mamy okazję zacząć wcześniej generować zysk, poprawiając całkowity zwrot z inwestycji (przykład znajdziesz w rozdziale 14.).

Ograniczanie błędów

Sprinty o krótkim czasie trwania ograniczają błędy. Jak bardzo możemy się pomylić w ciągu dwóch tygodni? Nawet jeśli całkowicie „położyliśmy” sprint, straciliśmy jedynie dwa tygodnie. Nacisk na sprinty o krótkim czasie trwania jest tak ważny, ponieważ pozwala na częstą koordynację i uzyskiwanie informacji zwrotnych. W ten sposób nawet jeśli jesteśmy w błędzie, jest to na szczęście mały błąd.

Rozbudzenie podekscytowania

Krótkie sprinty mogą pomóc w rozbudzeniu podekscytowania. Cechą natury ludzkiej jest spadek zainteresowania nagrodą i wynikająca z tego ekscytacja wraz z upływem czasu (patrz rysunek 4.4).



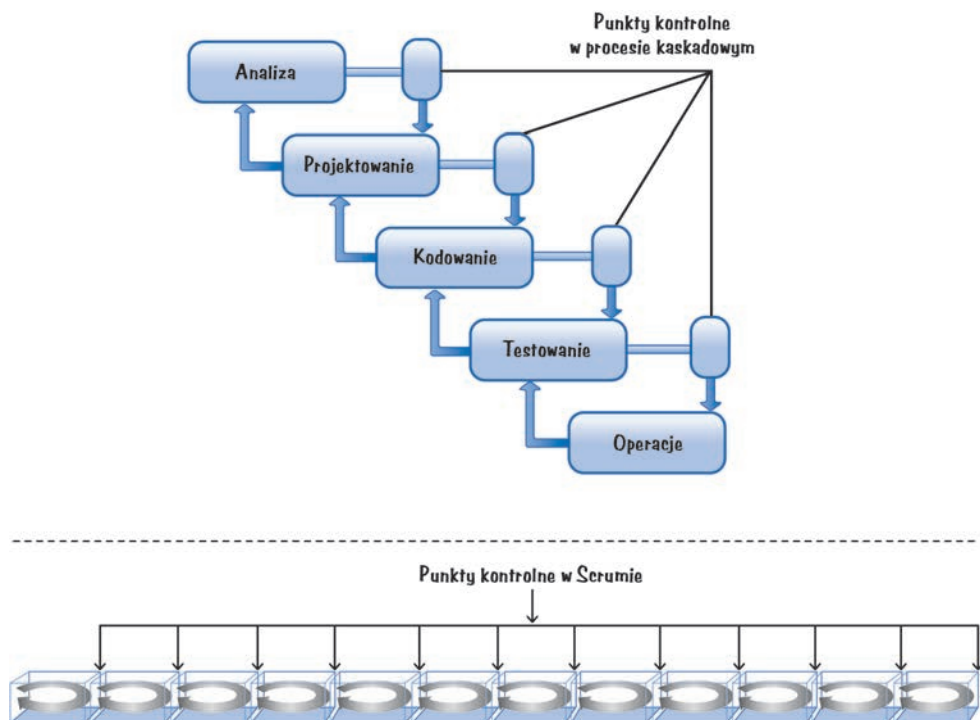
RYСУNEK 4.4. Podekscytowanie w funkcji czasu

Jeżeli pracujemy nad projektem o bardzo długim czasie trwania, nie tylko jest większe prawdopodobieństwo porażki, ale również zwiększone ryzyko utraty entuzjazmu do ponoszenia dalszych wysiłków. (Kiedy pracowałem w IBM, takie zjawiska określaliśmy mianem projektów w stylu „zagosz ocean”, ponieważ trwały one naprawdę bardzo długo, a ich ukończenie — o ile było możliwe — wymagało wielkiego wysiłku, porównywalnego z próbą zagotowania oceanu). Nie widząc postępów lub też perspektywy końca na horyzoncie, ludzie zaczynają tracić zainteresowanie. Pod koniec są gotowi zapłacić komuś za możliwość przejścia do innego produktu!

Sprinty o krótkim czasie trwania podtrzymują ekscytację uczestników dzięki nieustannemu dostarczaniu działających rzeczy. Zadowolenie płynące z pojawiających się wcześniej i regularnie działających przyrostów produktu rozbudza nasze zainteresowanie i pragnienie kontynuacji pracy nad celem końcowym.

Regularne punkty kontrolne

Krótkie sprinty dostarczają wielu użytecznych punktów kontrolnych (patrz rysunek 4.5).



RYСУNEK 4.5. Porównanie punktów kontrolnych

Jednym z cennych aspektów projektów sekwencyjnych są dobrze zdefiniowane kamienie milowe. Dają one menedżerom narzędzie w postaci zdefiniowanych punktów kontrolnych w trakcie cyklu życia projektu, które zazwyczaj służą do podjęcia decyzji o kontynuacji lub zaprzestaniu finansowania kolejnej fazy projektu. Chociaż potencjalnie użyteczne z punktu widzenia nadzoru, kamienie milowe nie odzwierciedlają prawdziwego stanu wartości dla klienta (o czym pisałem w rozdziale 3.).

Scrum daje menedżerom, interesariuszom, właścicielom produktu i innym uczestnikom o wiele więcej punktów kontrolnych w porównaniu z ilością oferowaną przez projekty sekwencyjne. Istotny punkt kontrolny (przegląd sprintu) pojawia się pod koniec każdego sprintu. Pozwala on każdemu na podjęcie decyzji w oparciu o demonstrowane, działające cechy produktu. Ludzie radzą sobie lepiej w złożonym środowisku, kiedy podczas punktów kontrolnych mają możliwość bardziej interaktywnej inspekcji i adaptacji.

Stały czas trwania

Zgodnie z zasadami zespół powinien wybrać stały rozmiar sprintów dla konkretnego projektu i nie zmieniać go, o ile nie istnieje ważny powód ku temu. Ważne powody to między innymi:

- Rozważanie przez Ciebie przejścia ze sprintów 4-tygodniowych na 2-tygodniowe w celu częstszego otrzymywania informacji zwrotnej. Przed podjęciem ostatecznej decyzji chcesz wykonać kilka sprintów 2-tygodniowych na próbę.
- Coroczny okres urlopowy lub okres kończący rok finansowy powodują, że bardziej praktyczne jest wykonywanie 3-tygodniowych sprintów zamiast zwyczajowych 2-tygodniowych.
- Produkt ma zostać wdrożony w ciągu tygodnia, zatem wykonywanie dwutygodniowego sprintu byłoby marnotrawstwem.

Fakt, iż zespół nie jest w stanie wykonać całej pracy w ciągu aktualnie obowiązującej długości sprintu, nie jest przekonującym powodem do zwiększenia długości sprintów. Podobnie niedopuszczalne jest lobbowanie na rzecz przedłużenia sprintu po dotarciu do ostatniego dnia i przekonaniu się, że zabrakło czasu na wykonanie całej pracy. Są to symptomy złego funkcjonowania oraz okazje na poprawienie procesu, ale nie powody usprawiedliwiające wydłużanie sprintów.

Zatem zgodnie z zasadami: jeśli zespół zgodzi się na wykonywanie sprintów dwutygodniowych, wszystkie sprinty powinny trwać dokładnie tyle czasu. Z praktycznych względów większość zespołów (choć nie wszystkie) definiuje dwa tygodnie jako dziesięć dni roboczych. Jeśli w trakcie sprintu pojawia się dzień wolny od pracy lub jednodniowe szkolenie, liczba dni roboczych sprintu zostaje zmniejszona, ale bez wpływu na fizyczny czas jego trwania.

Stosowanie sprintów o tej samej długości przynosi dodatkowe zalety wynikające z taktowania oraz upraszcza planowanie.

Zalety taktowania

Sprinty o takiej samej długości zapewniają **takt** — regularny, przewidywalny rytm lub też puls scrumowego wysiłku deweloperskiego. Stabilny, zdrowy takt pozwala zespołowi i organizacji na wzajemne przyswojenie reguł występowania poszczególnych zdarzeń zapewniających szybkość, a jednocześnie możliwy do kształtowania przepływ wartości biznesowej. Z własnego doświadczenia mogę powiedzieć, że regularny takt sprintów pozwala ludziom „wejść w temat”, „być przy piłce”, „wpaść w rutynę”. Uważam, że dzieje się tak, ponieważ regularny takt przekształca mało ciekawe, chociaż niezbędne działania w nawyk niewymagający poświęcania aktywnej uwagi i ten sposób pozwala skupić siłę umysłu na pracy przynoszącej radość i wartość dodaną.

Krótki takt sprintowy ma tendencję do niwelowania intensywności wysiłku. W przeciwieństwie do tradycyjnego, sekwencyjnego projektu, podczas którego obserwujemy szybki wzrost intensywności prac w fazach późniejszych, profile intensywności prac w kolejnych sprintach są podobne do siebie. Takt sprintów pozwala zespołom na pracę w podtrzymywalnym tempie — będę o tym pisał w rozdziale 11.

Wykonywanie sprintów z regularnym taktem znacząco obniża narzut pracy koordynacyjnej. Mając sprinty o ustalonej długości, możemy przewidzieć planowanie, przegląd i retrospekcję dla wielu sprintów jednocześnie. Dzięki temu, że wszyscy wiedzą, kiedy będą miały miejsce te aktywności, w znacznym stopniu redukujemy obowiązki niezbędne do zaplanowania dużej ilości sprintów.

Przykładowo: jeżeli w ciągu roku wysiłku deweloperskiego wykonujemy sprinty dwutygodniowe, możemy wysłać powtarzające się przez 26 iteracji zaproszenie na przegląd sprintu do wszystkich uczestników. Wyobraź sobie, ile dodatkowej pracy musielibyśmy wykonać, synchronizując spotkania dla wszystkich interesariuszy (zakładając oczywiście, że znaleźlibyśmy czas odpowiadający wszystkim istotnym interesariuszom, mającym swoje kalendarze wypełnione na wiele tygodni w przód) w sytuacji, kiedy pozwolilibyśmy na sprinty o różnej długości.

I w końcu, jeżeli nad projektem pracuje wiele zespołów o zbliżonym takcie sprintowym, możliwe jest synchronizowanie pracy między nimi (bardziej szczegółowe omówienie tego tematu znajdziesz w rozdziale 12.).

Uproszczone planowanie

Ujednolicony czas trwania sprintów upraszcza aktywności związane z planowaniem. Kiedy wszystkie sprinty mają taką samą długość (uwzględniając wahania wynikające z dni wolnych od pracy), zespół zyskuje komfort pod względem ilości pracy (**prędkości**), jaką może wykonać podczas typowego sprintu. Prędkość jest zazwyczaj wyznaczana w odniesieniu do sprintów. W przypadku sprintów o zmiennej długości nie możemy powiedzieć, że posiadamy znormalizowaną jednostkę pomiaru. Zdanie „Zespół ma przeciętną prędkość 20 punktów na sprint” byłoby pozbawione sensu.

Wyznaczenie prędkości zespołu pracującego ze zmienną długością sprintów jest możliwe, ale wymaga bardziej skomplikowanych obliczeń. Trzymanie się jednakowej długości sprintów upraszcza wyliczanie prędkości w oparciu o dane historyczne zespołu.

Uproszczeniu ulegają również inne działania matematyczne związane z planowaniem. Na przykład: jeżeli pracujemy nad **dystrybucją z ustaloną datą** i nasze sprinty mają jednakową długość, wyznaczenie liczby sprintów w ramach tej wersji sprowadza się do prostych obliczeń kalendarzowych (znamy datę dzisiejszą, datę dystrybucji i wiemy, że wszystkie sprinty mają taką samą długość). Gdyby sprinty mogły mieć różną długość, wyznaczenie liczby sprintów nie byłoby już takie proste (musielibyśmy przeprowadzić wczesne wyczerpujące planowanie) — wprowadziłoby niepotrzebny narzut pracy, a wyniki nie dawałyby takiej samej pewności jak te wynikające ze stałej długości sprintów.

Niezmienność celu

Istotna zasada Scruma mówi, że po ustaleniu celu sprintu i rozpoczęciu jego wykonania nie wolno wprowadzać zmian, które mogą fizycznie wpłynąć na jego cel.

Czym jest cel sprintu?

Każdy sprint można podsumować poprzez jego cel, który opisuje motywy biznesowe i wartość sprintu. Zazwyczaj cel sprintu jest wyrażony w jasny sposób i skupia się na pojedynczej rzeczy, na przykład jest to:

- Pomoc w stworzeniu raportu wstępnego.
- Załadowanie i sprawdzenie danych mapy Ameryki Północnej.
- Zademonstrowanie zdolności do wysłania wiadomości tekstowej poprzez zintegrowany stos oprogramowania, sterownika i sprzętu.

Czasami cel sprintu może być wyrażony w formie więcej niż jednego faktu, na przykład „U uruchomić drukowanie na podstawowym poziomie i dodać obsługę wyszukiwania po dacie”.

Podczas planowania sprintu zespół deweloperski powinien pomagać w ustaleniu i doprecyzowaniu celu, a następnie kierując się tym celem, wskazać elementy z rejestru produktu, które mogą zostać zrealizowane do końca sprintu (więcej na ten temat w rozdziale 19.). Wybrane elementy dziennika produktu posłużą do dalszego uszczegółowienia celu sprintu.

Wzajemne zobowiązanie

Cel sprintu jest podstawą do wzajemnego zobowiązania pomiędzy zespołem i właścicielem produktu. Zespół zobowiązuje się zrealizować cel przed końcem sprintu, natomiast właściciel produktu zobowiązuje się nie zmieniać celu sprintu w trakcie jego trwania.

To wzajemne zobowiązanie pokazuje wagę sprintów w równoważeniu wymogu adaptacji biznesu do zachodzących zmian, z potrzebą skoncentrowania się zespołu na zadaniu i kreatywnego wykorzystania swojego talentu do stworzenia wartości podczas ustalonej, krótkiej iteracji. Definiując cel sprintu i przestrzegając go, zespół scrumowy jest w stanie skupić się na dobrze zdefiniowanym, wartościowym zadaniu.

Zmiana kontra doprecyzowanie

Chociaż nie wolno fizycznie *zmieniać* celu sprintu, dopuszcza się jego *doprecyzowanie*. Pozwolę sobie rozróżnić te dwa pojęcia.

Czym cechuje się zmiana? Zmiana to każda modyfikacja zakresu pracy lub zasobów mogąca doprowadzić do ekonomicznie istotnego marnotrawstwa, szkodliwej przerwy w przepływie pracy lub znaczącego zwiększenia ilości prac w sprintcie. Typowe zmiany to dodanie lub usunięcie ze sprintu elementu rejestru produktu lub istotna zmiana zakresu elementu z rejestru produktu, który jest już w sprintcie. Oto przykład ilustrujący zmianę:

Właściciel produktu: „Kiedy mówiłem, że musimy mieć możliwość przeszukiwania policyjnej bazy danych pod kątem młodocianych przestępców, nie miałem na myśli tylko wyszukiwania po imieniu i nazwisku. Chciałem również mieć możliwość wyszukiwania w oparciu o zdjęcia tatuaży podejrzanych!”.

Dodanie możliwości przeszukiwania w oparciu o zdjęcia wymaga najprawdopodobniej zdecydowanie więcej wysiłku i niemal na pewno wpłynie na zdolność zespołu do dotrzymania zobowiązania dostarczenia funkcjonalności wyszukiwania po imieniu i nazwisku. W takiej sytuacji właściciel produktu powinien rozważyć stworzenie nowego elementu (historyjki) obejmującego wyszukiwanie w oparciu o zdjęcia i dodanie go do rejestru produktu z zamiarem skierowania do realizacji w kolejnym sprincie.

Czym cechuje się doprecyzowanie? Doprecyzowanie to dodatkowe szczegóły dostarczone zespołowi w trakcie sprintu, pomagające mu w osiągnięciu celu sprintu. Na początku sprintu elementy rejestru produktu mogą nie być dostatecznie uszczegółowione (będzie o tym mowa w rozdziale 5.). Stąd zupełnie zrozumiałe jest, że zespół będzie zadawał pytania wyjaśniające w trakcie sprintu, a właściciel produktu będzie na nie odpowiadał. Doprecyzowanie ilustruje poniższy przykład:

Zespół deweloperski: „Kiedy mówiłeś, że znalezione rekordy z bazy danych młodocianych przestępców powinny zostać wyświetlone w formie listy, miałeś na myśli jakiś konkretny porządek tej listy?”.

Właściciel produktu: „Tak, posortujcie ją alfabetycznie po nazwisku”.

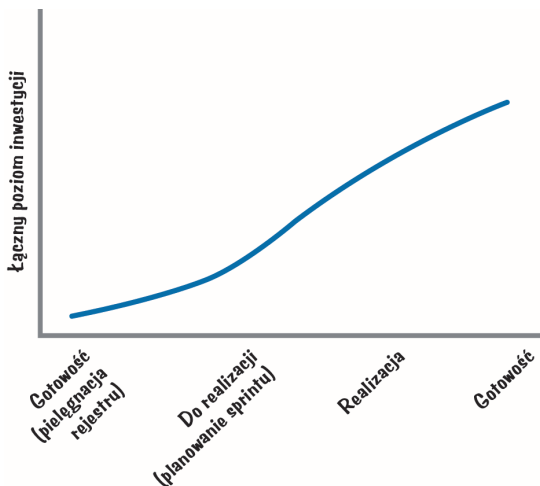
Zespół deweloperski: „Dobrze, tak zrobimy”.

W ten sposób właściciel produktu może i powinien dostarczać wyjaśnień w trakcie sprintu.

Konsekwencje zmiany

Może wydawać się, że zasada zabraniająca zmiany celu sprintu stoi w sprzeczności z fundamentalną zasadą Scruma mówiącą, że powinniśmy być otwarci na zmiany. Jesteśmy otwarci na zmiany, ale chcemy to robić w sposób zrównoważony i mający sens ekonomiczny.

Konsekwencje ekonomiczne zmiany rosną razem z poziomem naszego zaangażowania w pracę wynikającą ze zmian (patrz rysunek 4.6).



RYСУNEK 4.6. Łączny poziom inwestycji w kolejnych fazach

Inwestujemy w elementy rejestru produktu, aby przygotować je do realizacji w sprincie. Kiedy jednak sprint już wystartuje, nasz poziom inwestycji w te elementy wzrasta (ponieważ poświęciliśmy nasz czas w trakcie planowania sprintu na dyskusje i rozpisanie zadań). Jeśli chcemy wprowadzić zmianę po planowaniu sprintu, nie tylko narażamy naszą inwestycję w planowanie, ale również wprowadzamy dodatkowe koszty wynikające z konieczności ponownego rozplanowania zmian w trakcie sprintu.

Dodatkowo kiedy już rozpoczniemy wykonanie sprintu, nasza inwestycja w pracę zwiększa się jeszcze bardziej w miarę przechodzenia elementów rejestru produktu przez kolejne fazy od przygotowania (praca czekająca na wykonanie), przez realizację (wykonanie pracy) do gotowości (zakończenie pracy).

Żałujemy, że chcemy wymienić cechę X będącą częścią aktualnego zobowiązania sprintowego na cechę Y, która nie została wzięta do sprintu. Nawet jeśli nie zaczęliśmy jeszcze pracować nad cechą X, ponosimy stratę związaną z planowaniem. Dodatkowo cecha X może być powiązana z innymi cechami w sprincie, zatem dotycząca jej zmiana może wypłynąć również na inne elementy sprintu i odbić się na całym celu sprintu.

Jeżeli prace nad cechą X zostały już rozpoczęte, to oprócz wspomnianych już strat możemy wprowadzić inne formy marnotrawstwa. Na przykład trzeba będzie porzucić wyniki zrealizowanych już częściowo prac nad cechą X bez możliwości skorzystania z nich w przyszłości (nie zgadzamy się na umieszczenie w potencjalnie gotowym do wdrożenia przyroście produktu pracy ukończonej tylko w części).

Oczywiście, jeśli cecha X jest już gotowa¹, możemy potencjalnie stracić niemal pełną wartość poniesionych na nią inwestycji. Całe to marnotrawstwo kumuluje się!

Bezpośredni wpływ marnotrawstwa na ekonomię przedsięwzięcia to jedna rzecz, ale mogą również wystąpić straty pośrednie wynikające z potencjalnego pogorszenia motywacji i zaufania zespołu w obliczu zmiany. Kiedy właściciel produktu zobowiązuje się nie zmieniać celu, a następnie łamie to zobowiązanie, zespół w naturalny sposób traci motywację, co z całą pewnością wpłynie na chęć pilnej pracy i ukończenia pozostałych elementów rejestru produktu znajdujących się w sprincie. Złamanie zobowiązania może narazić na szwank zaufanie w zespole scrumowym — zespół deweloperski przestanie ufać w wolę właściciela produktu do przestrzegania przyjętych przez niego zobowiązań.

Pragmatyczność

Zakaz wprowadzania zmian w sprincie jest tylko zasadą, nie przepisem prawa. Zespół scrumowy musi być pragmatyczny.

Co zrobić, jeśli warunki biznesowe ulegną takim modyfikacjom, że zmiana celu sprintowego będzie nieunikniona? Żałujemy, że w trakcie trwania sprintu nasz konkurent wprowadził na rynek swój nowy produkt. Po jego analizie dochodzimy do wniosku, że musimy zmienić cel ustalony dla naszego bieżącego sprintu, ponieważ to, co teraz robimy, nagle straciło sens ekonomiczny w obliczu osiągnięć konkurencji. Czy powinniśmy ślepo przestrzegać zasady zabraniającej zmiany celu sprintu i jego zawartości? Prawdopodobnie nie.

¹ Zapewne chodzi tutaj o sytuację, w której dana cecha została wykonana, ale nie przeszła jeszcze przez końcowe fazy produkcji (testy automatyczne, weryfikacja manualna) i nie może zostać zintegrowana z resztą produktu — *przyj. tłum.*

Jeśli jeden z systemów produkcyjnych w firmie ulegnie poważnej awarii, a jedynymi osobami zdolnymi do jego naprawy są ludzie w naszym zespole, to czy powinniśmy przerwać bieżący sprint, aby dokonać naprawy? Czy też odpowiemy, że będziemy w stanie naprawić ten system na samym początku następnego sprintu? Prawdopodobnie nie.

W ostatecznym rozrachunku bycie pragmatycznym przebija zasadę zakazu zmiany celu. Musimy działać w sposób uzasadniony ekonomicznie. Każdy członek zespołu scrumowego powinien to rozumieć. Jeśli zmienimy zawartość bieżącego sprintu, odnotujemy negatywne skutki ekonomiczne, które opisałem wcześniej. Jeśli jednak konsekwencje ekonomiczne zmiany będą o wiele mniejsze od skutków jej odroczenia, przeprowadzenie zmiany można uznać za mądrą decyzję biznesową. Nie należy modyfikować sprintu, jeśli wprowadzenie zmiany i powstrzymanie się od niej nie niosą ze sobą konsekwencji finansowych.

Co do motywacji i zaufania zespołu — z własnego doświadczenia mogę powiedzieć, że kiedy właściciel produktu prowadzi z zespołem szczerą, skupioną na ekonomii dyskusję dotyczącą konieczności zmiany, większość potrafi zrozumieć konieczność podjęcia takich działań, a motywacja i zaufanie zostają podtrzymane.

Zakończenie przed czasem

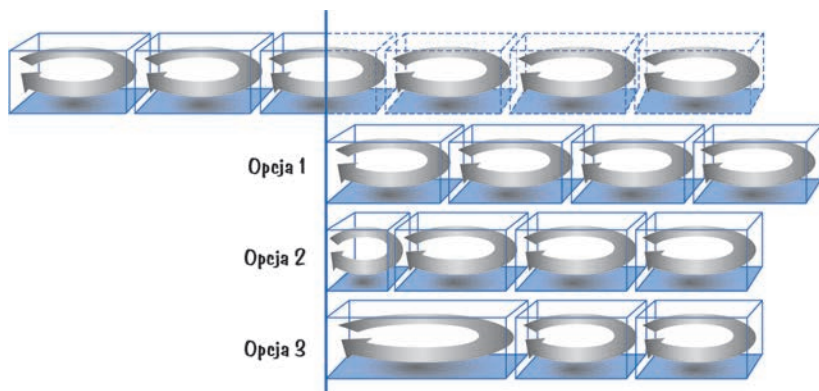
Jeśli cel sprintu stanie się zupełnie nieaktualny, zespół scrumowy powinien uznać kontynuację bieżącego sprintu za pozbawioną sensu i zaproponować właścicielowi produktu jego przedwczesne zakończenie. Zakończenie sprintu przed czasem kończy wszelką pracę, a zespół scrumowy organizuje spotkanie w celu przeprowadzenia retrospekcji. Następnie odbywa się planowanie kolejnego sprintu z nowym celem i innym zestawem elementów rejestru produktu.

Sprint przerywany jest w obliczu ważnego wydarzenia o znaczeniu ekonomicznym. Może to być działanie konkurencji, które fizycznie zmienia sposób finansowania sprintu lub całego produktu.

Chociaż właściciel produktu ma prawo przerwać każdy sprint, z własnego doświadczenia wiem, że opcja ta jest bardzo rzadko wykorzystywana. Często zespół scrumowy podejmuje działania o wiele mniejszego kalibru, aby dostosować się do nowych warunków. Pamiętaj, że sprints są krótkie i statystycznie rzecz biorąc, zespół będzie gdzieś w środku sprintu, kiedy wystąpi zdarzenie wymuszające zmianę. W takiej sytuacji do zakończenia sprintu pozostanie mniej więcej tydzień czasu i z ekonomicznego punktu widzenia przerwanie sprintu może być mniej opłacalne od pozostania na kursie do zakończenia. W wielu przypadkach można podjąć mniej dramatyczne decyzje, na przykład aby znaleźć czas na naprawienie krytycznego problemu produkcyjnego, wystarczy porzucić pracę nad jedną z cech, zamiast przerywać cały sprint.

Należy zdać sobie sprawę, iż przerywanie sprintu przed czasem ma nie tylko negatywny wpływ na morale, ale jest również poważnym zakłóceniem w szybkim i elastycznym procesie powstawania nowych cech, negującym wiele z zalet sprintów o jednakowej długości, o których mówiłem wcześniej. Przerwanie sprintu powinno być środkiem ostatecznym.

Jeżeli sprint zostanie przerwany, zespół scrumowy będzie musiał określić długość kolejnego sprintu (patrz rysunek 4.7).



RYSUNEK 4.7. Wybór długości następnego sprintu po przerwaniu sprintu

Istnieją trzy sensowne możliwości:

1. Pozostać przy oryginalnej długości sprintu. To rozwiązanie pozwala na zachowanie sprintów o jednakowej długości w całym procesie deweloperskim (oczywiście z wyjątkiem sprintu, który został przerwany). Jeżeli w produkcji bierze udział więcej zespołów, zachowanie oryginalnej długości sprintów spowoduje rozsynchronizowanie jedynie zespołu scrumowego, który przerwał swój sprint.
2. Skrócić kolejny sprint tak, aby jego data zakończenia pokryła się z datą zakończenia przerwanych sprintów. Na przykład: jeśli zespół scrumowy przerwał dwutygodniowy sprint pod koniec pierwszego tygodnia, kolejny sprint będzie miał tylko tydzień i w ten sposób zsynchronizuje zespół z jego oryginalnym rytmem sprintowym.
3. Przedłużyć kolejny sprint ponad rozmiar normalnego sprintu, tak aby objął swoim zasięgiem czas sprintu przerwanych i czas pełnego sprintu. W naszym poprzednim przykładzie ponowne zsynchronizowanie z oryginalnym rytmem sprintowym nastąpi po utworzeniu sprintu trzytygodniowego.

Jeżeli w procesie produkcyjnym bierze udział więcej niż jeden zespół, preferowane jest rozwiązanie nr 2 lub 3. W celu trafego wyboru jednej z powyższych opcji będziesz musiał rozważyć swój konkretny kontekst.

Definicja ukończenia

W rozdziale 2. pisałem o tym, że wynik każdego sprintu powinien być potencjalnie nadającym się do wdrożenia przyrostem produktu. Wspomniałem, że „potencjalnie nadający się do wdrożenia” nie oznacza wcale, że to, co zostało zbudowane, zostanie faktycznie wdrożone. Wdrożenie jest decyzją biznesową, często podejmowaną z zupełnie odmiennym taktem. W niektórych organizacjach wdrażanie co sprint może mieć rację bytu.

Potencjalną zdolność do wdrożenia lepiej jest traktować jako stan pewności, że to, co zbudowaliśmy w sprincie, zostało faktycznie ukończone, czyli nie pozostała do wykonania żadna istotna praca (taka jak testowanie, integracja itp.), którą należałoby wykonać przed wdrożeniem wyników sprintu, gdybyśmy chcieli podjąć decyzję biznesową o wdrożeniu. Ustalenie, czy to, co zostało wyprodukowane, nadaje się do potencjalnego wdrożenia, wymaga posiadania przez zespół scrumowy dobrze zdefiniowanej i uzgodnionej definicji ukończenia.

Czym jest definicja ukończenia?

Koncepcyjnie **definicja ukończenia** to lista kontrolna różnych typów pracy, jakie zespół powinien zakończyć z powodzeniem, zanim będzie mógł zadeklarować, że wyniki pracy nadają się potencjalnie do wdrożenia (patrz tabela 4.1).

TABELA 4.1. Przykładowa definicja listy kontrolnej definicji ukończenia

Definicja ukończenia	
<input type="checkbox"/>	Weryfikacja projektu
<input type="checkbox"/>	Kodowanie
<input type="checkbox"/>	Kod został zrefaktoryzowany
<input type="checkbox"/>	Kod posiada format zgodny ze standardem
<input type="checkbox"/>	Umieszczono komentarze w kodzie
<input type="checkbox"/>	Kod został umieszczony w repozytorium
<input type="checkbox"/>	Kod został przejrany
<input type="checkbox"/>	Aktualizacja dokumentacji
<input type="checkbox"/>	Testy
<input type="checkbox"/>	Wykonano testy jednostkowe
<input type="checkbox"/>	Wykonano testy integracyjne
<input type="checkbox"/>	Wykonano testy regresyjne
<input type="checkbox"/>	Wykonano testy na wszystkich obsługiwanych platformach
<input type="checkbox"/>	Przetestowano pod względem różnych wersji językowych
<input type="checkbox"/>	Zerowy stan znanych problemów
<input type="checkbox"/>	Testy akceptacyjne
<input type="checkbox"/>	Rozwiązanie umieszczone na serwerach produkcyjnych

Faktyczna zawartość listy będzie zależeć od wielu zmiennych:

- Natury budowanego produktu.
- Technologii używanych do budowy produktu.
- Organizacji, która buduje produkt.
- Aktualnych przeszkód ograniczających możliwe rozwiązania.

W większości przypadków absolutne minimum definicji ukończenia powinno wyrażać się przez dostarczenie kompletnego fragmentu funkcjonalności — takiego, który został zaprojektowany, zbudowany, zintegrowany, przetestowany, udokumentowany i zweryfikowany pod względem dostarczania faktycznej wartości dla klienta. Skonstruowanie użytecznej listy kontrolnej wymaga podzielenia tych głównych zadań na bardziej szczegółowe. Co oznacza na przykład przetestowanie? Czy chodzi o puszczenie testów jednostkowych, sprawdzenie całego systemu, weryfikację na różnych platformach, czy też testowanie wersji językowych programu? Prawdopodobnie do tej listy mógłbyś dołożyć jeszcze inne testy, specyficzne dla Twojego produktu. Czy wszystkie wymienione typy testowania zostały włączone do definicji ukończenia?

Pamiętaj, że jeśli nie przeprowadzasz istotnych testów w każdym sprincie (na przykład testów wydajnościowych), będziesz musiał wykonać je w innym czasie. Czy utworzysz na końcu sprint poświęcony wyłącznie testowaniu wydajności? Jeśli zrobisz w ten sposób w sytuacji, kiedy testowanie wydajności jest istotnym elementem „ukończenia”, to tak naprawdę nie będziesz posiadał potencjalnie nadających się do wdrożenia przyrostów produktu co sprint. Co gorsza, jeżeli pod koniec okaże się jeszcze, że wyniki tych testów nie są zadowalające, staniesz w obliczu poważnego problemu w końcowej fazie produkcji i będziesz musiał poświęcić sporo czasu i pieniędzy na jego naprawienie — więcej w porównaniu z kosztem wykonywania testów wydajnościowych wcześniej.

Czasami testowanie może wymagać więcej czasu, niż zajmuje sam sprint. Jeżeli wynika to z ogromnej liczby testów manualnych, jakie trzeba przeprowadzać, zespół musi zacząć je automatyzować, tak aby można było dokończyć testowanie w ramach sprintu. Jeżeli czas trwania testów wynika z ich natury, trzeba zaakceptować fakt, iż testowanie rozpocznie się w jednym sprincie, a zakończy w którymś z kolejnych. Mogę podać przykład trenowanej przeze mnie organizacji, która budowała urządzenie składające się ze sprzętu, sterowników i oprogramowania. Jednym z ich standardowych testów był trwający 1500 godzin test wytrzymałościowy — w tym czasie urządzenie pracowało non stop w celu sprawdzenia, czy nie zawiedzie. Takiego testu nie można przeprowadzić w trakcie dwutygodniowego sprintu, zatem zespół scrumowy musiał zmodyfikować swoje kryteria ukończenia tak, aby móc ogłosić zakończenie prac w sprincie mimo trwającego jeszcze testu 1500 godzin.

Często jestem pytany: „Co zrobić, jeśli ostatniego dnia sprintu ujawniony zostanie poważny błąd? Czy w takiej sytuacji można uznać element rejestru produktu za ukończony?” Nie, nie można! A ponieważ zgodnie z zasadą nie przedłużamy sprintów ponad zaplanowane ograniczenie czasowe, nie przedłużymy go również o dzień lub dwa w tym przypadku, żeby móc naprawić błąd w trwającym jeszcze sprincie. Zamiast tego w zaplanowanym dniu zakończenia sprintu niekompletny element rejestru produktu zostaje wyjęty z bieżącego sprintu i wstawiony do rejestru produktu w miejscu odpowiednim ze względu na jego priorytet wobec pozostałych elementów rejestru. To pozwoli na jego potencjalne dokończenie w jednym z przyszłych sprintów.

Zespoły scrumowe muszą posiadać solidną definicję ukończenia prac, taką, która pozwala na osiągnięcie wysokiego stopnia pewności, że to, co zostało zbudowane, ma odpowiednią jakość i może zostać wdrożone. Każde odstępstwo od tej zasady pozbawia organizację biznesowych możliwości wdrażania produktów w stosownym do tego czasie i powoduje narastanie długu technicznego (o czym będzie mowa w rozdziale 8.).

Definicja ukończenia może ewoluować

Definicję ukończenia możesz traktować jako definicję stanu prac pod koniec sprintu. Dla wielu bardzo wydajnych zespołów celem końcowym pracy jest umożliwienie jej potencjalnego wdrożenia i cel ten pozostaje względnie stały w trakcie całego procesu produkcyjnego.

Na przykład: kiedy byłem właścicielem produktu podczas przeprojektowywania witryny internetowej Scrum Alliance w 2007 roku, wykonywaliśmy sprinty jednotygodniowe. Końcowy stan naszej definicji ukończenia można było podsumować stwierdzeniem „uruchomienie na serwerach produkcyjnych”. Zespół i ja uznaliśmy, że był to najbardziej rozsądny stan, jaki chcieliśmy osiągnąć w każdym sprincie. Zdefiniowaliśmy ten stan na początku prac deweloperskich i pozostał on niezmieniony przez cały czas mojego szefowania jako właściciela produktu.

Wiele zespołów zaczyna jednak od definicji ukończenia, która nie pozwala na stwierdzenie, iż wszystkie cechy zostały wykonane w stopniu pozwalającym na ich publikację lub dystrybucję. Dla niektórych pewne rzeczywiste przeszkody mogą uniemożliwić osiągnięcie takiego stanu na początku procesu produkcyjnego, mimo że jest to cel, który chcą ostatecznie osiągnąć. Te zespoły (z konieczności) będą zaczynać od uboższego stanu końcowego sprintu i stopniowo modyfikować swoje kryteria ukończenia w miarę usuwania przeszkód przez organizację.

Miałem okazję odwiedzić organizację zajmującą się budowaniem systemów informatycznych dla klinik medycznych. Ich produkty instalowane są w szpitalach i magazynują różnorodne dane medyczne (w niektórych przypadkach bezpośrednio z urządzeń wykonujących testy diagnostyczne). Zespół deweloperski pracujący nad danym testem klinicznym posiadał umiejętność zainstalowania oprogramowania w laboratorium i przekonania się, że działa ono dobrze ze sprzętem — rzecz, która powinna być przetestowana, zanim produkt zostanie wdrożony. Ponieważ jednak deweloperzy nie mieli stałego dostępu do laboratorium, na początku zespół nie włączył testów klinicznych do swoich kryteriów ukończenia pracy. Zamiast tego wyznaczył sprinty testów klinicznych pod koniec prac nad każdą wersją dystrybucyjną.

Podczas dyskusji dowiedziałem się, że zarówno ludzie z marketingu, jak i członkowie zespołu scrumowego nie cierpieli tych testów klinicznych przed wypuszczeniem wersji. Nikt nie był w stanie przewidzieć, ile sprintów zajmie usunięcie wszystkich błędów, a produkt nie mógł zostać wypuszczony, zanim nie zostały usunięte wszystkie błędy. Kiedy zastanawialiśmy się nad możliwymi rozwiązaniami tego problemu, w rozmowy włączył się dyrektor techniczny, zadając pytanie: „Czy gdybyście mieli dostęp do laboratorium klinicznego, byłibyście w stanie wykonywać testy na miejscu w każdym sprincie?”.

Członkowie zespołu przedyskutowali jego pytanie i odpowiedzieli: „Tak, ale to oznacza, że będziemy realizować mniej cech w każdym sprincie”. Dyrektor zgodził się usunąć przeszkodę, udostępniając zespołowi laboratorium w lokalnej klinice uniwersyteckiej. Właściciel produktu stwierdził, że wyprodukowanie mniejszej liczby cech w każdym sprincie jest rozsądnym kompromisem w zamian za wiedzę, że cechy, które zostały zbudowane, przeszły również testy kliniczne. W tym momencie zespół mógł zmodyfikować swoją definicję ukończenia, osiągając faktyczny stan „przyrostu nadającego się potencjalnie do dystrybucji”, co dało wszystkim zainteresowanym osobom większe poczucie pewności odnośnie do pracy realizowanej w każdym sprincie.

Zespół może napotykać przeszkody, których nie da się usunąć „z marszu”. Wiadomo wtedy, że definicja ukończenia będzie musiała ewoluować w trakcie procesu produkcyjnego. Dobrym przykładem takiej sytuacji jest wytwarzanie produktu, na który składa się sprzęt i oprogramowanie.

Widziałem wiele przypadków zastosowania Scruma do takiej produkcji i często słyszałem od ludzi pracujących nad oprogramowaniem: „Sprzęt jest zawsze spóźniony!”. W sytuacjach takich jak ta zespół budujący oprogramowanie bez sprzętu, na którym to oprogramowanie ma działać, nie może twierdzić, że wyniki każdego sprintu nadają się potencjalnie do dystrybucji. Może co najwyżej stwierdzić, że spełnił kryteria gotowości „dla emulatora”, ponieważ w pierwszych sprintach testy wykonuje się w oparciu o emulator prawdziwego urządzenia. Później, kiedy dostępne będzie rzeczywiste urządzenie, definicja ukończenia zostanie zmodyfikowana i zacznie odzwierciedlać stan potencjalnej gotowości do wdrożenia lub przynajmniej coś bardziej zbliżonego do tego stanu.

Definicja ukończenia kontra kryteria akceptacji

Definicja ukończenia dotyczy przyrostu produktu wytwarzanego w czasie sprintu. Na przyrost produktu składa się kilka z elementów rejestru produktu, zatem każdy taki element musi zostać wykonany zgodnie z zadaniami przewidzianymi na liście kontrolnej definicji ukończenia.

W rozdziale 5. piszę o tym, że każdy element z rejestru produktu umieszczony w sprincie powinien posiadać swoje **warunki zadowolenia** (kryteria akceptacji specyficzne dla tego elementu) zdefiniowane przez właściciela produktu. **Kryteria akceptacji** będą ostatecznie weryfikowane przez **testy akceptacyjne**, które właściciel produktu wykona, aby sprawdzić, czy gotowa funkcjonalność działa zgodnie z oczekiwaniami. Na przykład: jeśli element rejestru produktu brzmi „Pozwolić klientowi na zakup przy użyciu karty kredytowej”, warunkiem zadowolenia może być „Działa z użyciem kart AmEx, Visa i MasterCard”. Zatem każdy element z rejestru produktu będzie posiadał swoje indywidualne kryteria akceptacji. Te kryteria specyficzne dla elementu są dodatkiem, nie zastępstwem, do kryteriów ukończenia wyznaczonych przez listę kontrolną definicji ukończenia, które obowiązują wszystkie elementy rejestru produktu.

Element rejestru produktu można uznać za ukończony, jedynie jeśli spełnione zostały zarówno kryteria akceptacji specyficzne dla tego elementu (na przykład „działa ze wszystkimi kartami kredytowymi”), jak i wszystkie punkty definicji ukończenia na poziomie sprintu (na przykład „uruchomiono na serwerze produkcyjnym”).

Jeżeli określanie elementów rejestru produktu, które spełniły swoje kryteria akceptacji, mianem *ukończonych* jest mylące, nazywaj je elementami *skompletowanymi* lub *zaakceptowanymi*.

Ukończony kontra rzeczywiście ukończony

Niektóre zespoły przyjęły koncepcję „ukończenia” i „rzeczywistego ukończenia”. Okazuje się, że „rzeczywiście ukończone” jest bardziej ukończone od jedynie „ukończonego”. Zespoły nie powinny potrzebować tych dwóch różnych koncepcji, ale muszą się przyznać, że sam stosuję oba pojęcia w odniesieniu do mojego syna i jego pracy domowej. Miałem zwyczaj pytać go, czy już skończył (ukończył) swoją pracę domową, a on odpowiadał, że tak. Potem podczas spotkania z jego wychowawczynią zapytałem: „Czy kiedy mój syn zwraca pracę, jest ona zrobiona?”. W odpowiedzi usłyszałem „Nie za bardzo!”.

Po bardziej wnikliwej dyskusji z moim synem zrozumiałem, że jego rozumienie ukończenia oznaczało „Zrobiłem tyle pracy, ile byłem w stanie wykonać!”. Od tego momentu zacząłem używać określenia *rzeczywistego ukończenia*, które zgodnie z naszym obopólnym porozumieniem oznaczało „ukończone w takim stopniu, aby nauczyciel nabrał przekonania, że wykonałeś swoją pracę”.

Zespoły, które nie są przyzwyczajone do szybkiego dokończania prac, chętniej stosują kryterium rzeczywistego ukończenia jako podporę. Dla nich istnieje wyraźne rozróżnienie pomiędzy ukończeniem (wykonaniem takiej ilości pracy, jaką były w stanie zrealizować) a rzeczywistym ukończeniem (wykonaniem takiej ilości pracy, aby klienci nabrali przekonania, że zadanie zostało faktycznie zrealizowane). Zespoły, które rozumieją, że mogą uznać prace za ukończone tylko wtedy, kiedy wykonana została cała praca zaspokajająca potrzeby klienta, nie potrzebują dwóch stanów. Dla nich praca ukończona oznacza pracę rzeczywiści ukończoną.

Zakończenie

W tym rozdziale podkreśliłem kluczową rolę sprintów w środowisku scrumowym. Sprinty stanowią szkielet Scruma, z którym związana jest większość aktywności i artefaktów. Sprinty są krótkie, ograniczone czasowo i trwają jednakowo długo. Zazwyczaj zdefiniowane są poprzez cel sprintu, który może być modyfikowany tylko w przypadku ważnych przesłanek ekonomicznych. Wynikiem sprintów powinny być potencjalnie nadające się do wdrożenia przyrosty produktu wykonane według uzgodnionej wcześniej definicji ukończenia. W następnym rozdziale skupię się na danych wejściowych sprintu — wymaganiach — i ich najczęstszej formie — historyjkach użytkownika.

SKOROWIDZ

A

adaptacja, 67, 68
adaptowanie środowiska, 259
aktywności Scruma, 48
aktywność
 planowania
 portfela, 293
 sprintu, 357
 wersji dystrybucyjnej, 332
przeglądu sprintu, 387
retrospekcji sprintu, 399
wykonania sprintu, 368
analiza marginalna, 307
artefakty Scruma, 48
autoryzacja, 256

B

budowa zespołów scrumowych, 239
budowanie w sposób
 iteracyjny, 65
 przyrostowy, 65
burza mózgów, 125

C

całość przed kolejnym krokiem, 79
cechy
 mistrza młyna, 216
 obowiązkowe, 317, 323, 338
 właściciela produktu, 199, 200
 zespołu deweloperskiego, 226

cel sprintu, 99
 doprecyzowanie, 99
 zmiana, 100
cel wersji dystrybucyjnej, 318
ceremonie, 87
charakterystyka sprintu, 52
ciągła dystrybucja, 330
codzienne działania scrumowe, 55, 374
cykl życia produktu, 84
czarterowanie projektu, 321
czas trwania produkcji, 148

D

DEEP, 131
definicja
 gotowości, 138, 363
 ukończenia, 104, 106, 178, 385
definiowanie kryteriów, 197
demonstrowanie postępów, 93
dług techniczny, 23, 167
 celowy, 183
 konsekwencje, 169
 natywne, 168
 nieuniknione, 168
 obsługa, 183
 odkryty, 183
 przyczyny, 172
 spłacanie, 186
 strategiczny, 168
 ujawnianie, 180
 zarządzanie, 176
 znany, 183

dokładność, 153

domena

- chaosu, 41
- prostoty, 40
- skomplikowana, 40
- złożona, 40

doprecyzowanie celu sprintu, 99

dostarczanie wartości, 85

działanie

- w kopcu, 370
- w roju, 371

E

ekonomia

- długu technicznego, 178
- na poziomie rejestru produktu, 196
- na poziomie sprintów, 196
- na poziomie wersji dystrybucyjnej, 195
- pojedynczej wersji dystrybucyjnej, 279
- wielu wersji dystrybucyjnych, 279

eksploracja, 71

elementy

- rejestru produktu, 51, 129, 131
- skompletowane, 107

empiryczna kontrola nad procesem, 67

epos, 116

F

filtr

- ekonomiczny, 203, 299, 322
- strategiczny, 310

finansowanie

- pro wizoryczne, 325
- przyrostowe, 325

funkcjonalności, *Patrz* rejestr produktu

G

głosowanie kropkowe, 405

H

harmonogram

- dla portfela, 293, 295
- przeglądu sprintu, 384
- wypuszczania wersji dystrybucyjnych, 329

hierarchia abstrakcji historyjek, 117

hierarchiczne rejestry produktów, 144

historyjki

- pozyskiwania wiedzy, 123
- rejestru produktu, 110
- użytkownika, 111, 113
 - karta, 113
- kryteria INVEST, 118
- mały rozmiar, 121
- negocjowalność, 118
- niezależność, 118
- ocena jakości, 131
- oceniałość, 121
- potwierdzenie, 115
- poziom abstrakcji, 116
- rozmowa, 114
- testowalność, 122
- wartościowość, 120

I

idealne dni, 52, 156

identyfikacja spostrzeżeń, 403

inflacja punktowa, 165

informacje potwierdzające, 115

inspekcja, 67

integracja modułów i testowania, 77

integralność zespołu, 258

INVEST, 118

inwentarz, 323

iteracje, 34

J

jakość produktu, 86

jednostki oceny

- idealne dni, 156
- punkty historyjkowe, 155

K

kalendarz sprintów, 341

kanban, 41

kandydat

- na mistrza młyna, 219
- na właściciela produktu, 204

karty, 113

karty do planowania pokerowego, 158

klastry współpracy, 266

- komunikacja
 - przezroczysta, 233
 - szerokopasmowa, 232
 - komunikowanie postępów, 348, 350
 - kontrola empiryczna procesu, 67
 - koordynacja zespołów, 244
 - koszt
 - eksploracji, 71
 - opóźnienia, 83, 295
 - obliczenia, 296
 - profile, 297
 - pracy symultanicznej, 235
 - produkcji, 170
 - wielozadaniowości, 370
 - zmiany, 72, 74, 171
 - krótkie sprinty, 94
 - kryteria
 - akceptacji, 107
 - DEEP, 131
 - INVEST, 118
 - kształtowanie zespołów, 253
- L**
- limit pracy cząstkowej, 305
 - linia zdarzeń, 401
- Ł**
- łączenie
 - ról, 209, 221
 - zespołów, 243
- M**
- mapa
 - drogowa produktu, 284, 286, 317
 - historyjek, 125
 - sprintów, 339
 - menedżer projektu
 - czas, 263
 - integracja, 263
 - jakość, 263
 - kierowanie koordynacją, 267
 - ryzyko, 263
 - zakres, 263
 - zaopatrzenie, 263
 - zespół, 263
 - menedżer zasobów
 - definiowanie granic, 253
 - dostarczanie celu, 254
 - formowanie zespołów, 254
 - motywowanie ludzi, 257
 - obserwacja pomiarów, 261
 - promowanie wartości zwinności, 259
 - przewodzenie obszarowi funkcjonalności, 258
 - przystosowywanie grup wewnętrznych, 260
 - przystosowywanie partnerów, 260
 - raportowanie, 261
 - rozwijanie umiejętności, 257
 - upoważnianie zespołów, 256
 - usuwanie przeszkód organizacyjnych, 259
 - utrzymywanie integralności zespołu, 258
 - zarządzanie ekonomią, 261
 - zmiana składu zespołu, 255
 - metoda
 - kanban, 42
 - skauta, 186
 - mierzenie postępu, 85
 - miniprocess kaskadowy, 371
 - mistrz młyna, 47, 213
 - agent zmiany, 215
 - autorytet w dziedzinie procesu, 215
 - chronienie przed zakłóceniami, 215
 - cierpliwość, 217
 - mistrz służby, 214
 - proaktywność, 217
 - przezroczystość, 217
 - trenowanie, 213
 - umiejętność stawiania pytań, 216
 - usuwanie przeszkód, 215
 - wiedza, 216
 - współpraca, 217
 - model procesu scrumowego, 67
 - moment podejmowania decyzji, 69
- N**
- nadawanie ocen, 147, 150
 - najlepsze praktyki, 414
 - negocjowanie wymagań, 109
 - niepewność
 - klienta, 68
 - końca, 67
 - środków, 67
 - niepotrzebne formalności, 87
 - niezmienność celu, 98

0

- obiektywne dane, 397
- obliczanie
 - kosztów, 347
 - kosztu opóźnienia, 296
 - przedziału prędkości, 161
- obniżanie kosztu zmiany, 73
- obowiązki
 - menedżera projektu, 263
 - menedżera zasobów, 252
 - mistrza młyna, 213
 - właściciela produktu, 194
 - zespołu deweloperskiego, 224
- obsługa długu technicznego, 183
- obszary funkcjonalności, 251
- ocena, 147, 152
 - elementów portfela, 149
 - elementów produktu, 149
 - elementów rejestru, 133, 148, 151
 - zadań, 150
- oceniwanie
 - absolutne, 154
 - względne, 154
- oczekiwanie
 - na pracę, 81
 - na realizację, 81
- ograniczanie
 - błędów, 95
 - pracy cząstkowej, 92
- ograniczenia wersji dystrybucyjnej, 333–336
- ograniczenie czasowe
 - demonstrowanie postępów, 93
 - motywowanie domykania prac, 94
 - poprawianie przewidywalności, 94
 - unikanie perfekcjonizmu, 93
 - wymuszanie priorytetów, 93
 - zmniejszanie pracy cząstkowej, 92
- opcje planowania, 275
- outsourcing, 207

P

- pełnomocnik właściciela produktu, 210
- perfekcjonizm, 93
- persona, 125
- pętla
 - informacji zwrotnej, 77
 - zdobywania wiedzy, 77
- pielęgnacja
 - w procesach sekwencyjnych, 136
 - zespołów, 257
 - rejestru produktu, 134–138, 197, 225, 337
- pisanie historyjek, 125
- planowanie
 - codzienne, 287
 - długoterminowe, 329
 - hierarchiczne, 289
- pokerowe
 - karty, 158
 - reguły, 159
 - skala ocen, 157
- portfela, 283, 291–308
 - optymalizacja, 294
 - proces, 292
 - strategie, 294
 - strategie aktywności, 306
 - strategie napływu, 299
 - strategie odpływu, 304
 - tworzenie harmonogramu, 293
 - uczestnicy, 292
- produktu, 202, 283, 309
 - finansowanie, 325
 - obszary wartości, 314, 315
 - proces, 312
 - przedstawianie wizji, 315
 - szybkie działanie, 324
 - szybkie zdobywanie wiedzy, 326
 - uczestnicy, 310
 - wytyczne ekonomiczne, 322
- sprintu, 52, 99, 287, 355
 - dane wejściowe, 357
 - doprecyzowywanie celu, 365
 - dwuczęściowe, 358
 - jednoczęściowe, 359
 - nabieranie pewności, 364
 - określanie pojemności, 360
 - ustalanie zobowiązania, 365
 - wybieranie elementów rejestru, 363
- uproszczone, 98
- w samą porę, 275
- według kamieni milowych, 330
- wersji dystrybucyjnej, 285, 303, 329
 - komunikowanie postępów, 348, 350
 - lokalizacja cech obowiązkowych, 344
 - obliczanie kosztów, 347
 - ograniczenia, 333
 - określanie zakresu cech, 343

- proces, 331, 356
- rejestr produktu, 343
- uczestnicy, 331, 356
- z ustaloną datą, 341
- z ustalonym zakresem, 345
- wielopoziomowe, 281
- wykonania sprintu, 369
- z góry na dół, 227, 274
- pociąg wersji dystrybucyjnych, 246
- podejmowanie decyzji, 69, 72, 325
- podział zasad na kategorie, 63
- pojemność
 - w punktach historyjkowych, 361
 - w roboczogodzinach, 362
 - zespołu, 360
- porażka, 124
- porównanie
 - oceny, 154
 - punktów kontrolnych, 96
 - unikania i akumulowania długu, 180
 - wysiłku z dokładnością, 153
 - zaangażowania klienta, 198
 - zadań menedżera, 269
 - zasad, 89
- postęp, 84
- potwierdzenie ukończenia pracy, 385
- poziom
 - ufności, 322
 - zaufania, 312
- poziomy
 - autoryzacji, 256
 - planowania, 281
- pozyskiwanie
 - informacji zwrotnej, 77
 - wiedzy, 76
 - wymagań, 70
- praca
 - adaptacyjna, 75
 - cząstkowa, 78, 277, 305, 323
 - czekająca na realizację, 81, 304
 - nad cechą, 372
 - nad zadaniami, 372
 - planowana, 75
 - równoległa, 370
 - sterowana przerwaniem, 41
- prace deweloperskie
 - komercyjne, 205
 - wewnętrzne, 205
- pracownik czekający na pracę, 81, 304
- praktyki scrumowe, 46
- precyzja, 153
- prezentacja pracy zespołu, 386
- prędkość, 23, 98, 147, 160
 - nieprawidłowe wykorzystanie, 164
 - prognozowanie, 162
 - sztuczne zwiększanie, 173
- priorytety elementów rejestru, 134
- problemy retrospekcji sprintu, 409
- proces
 - sekwencyjny, 62, 65, 112
 - sterowany planem, 64, 66
 - wodospadowy, 62
 - zdefiniowany, 64
 - zwinnej produkcji, 34
- produkcja
 - oprogramowania, 64
 - przemysłowa, 64
 - sterowana planem, 61, 74
- produkt, 142
- prognoza, forecast, 49
- prognozowanie
 - prędkości, 162
 - ekstremalne, 375
 - sterowane testami, 174, 372
- przedwczesne zakończenie sprintu, 102
- przedział prędkości, 162
- przebieg sprintu, 57, 381
 - podejście, 386
 - problemy, 389
 - przygotowania, 384
 - uczestnicy, 382
- przejrzystość, 67
- przepływ, 139
 - jednoelementowy, 80
 - wytwarzania wartości, 261
- przeprowadzanie wizji, 321
- przerywanie sprintu, 102
- przewidywanie, 68
- przyczyny długu technicznego, 172
- punkt
 - przebiegu, 169
 - skupienia retrospekcji, 396
- punkty
 - historyjkowe, 52, 155
 - kontrolne, 96

R

refaktoryzacja kodu, 169

rejestr

portfela, 149

produktu, 34, 50, 111, 283, 316

duże produkty, 143

elementy, 129

emergencja, 132

ocenie elementu, 132

pielęgnacja, 134

podział elementów, 140

priorytety elementów, 133

stopień uszczegółowienia, 131

strumień wymagań, 141

wiele produktów, 145

wiele zespołów, 144

sposzrzeń, 407

sprintu, 53, 148, 364

retrospekcja sprintu, 58, 393

podjęcie, 398

problemy, 408

przygotowania, 396

uczestnicy, 395

zakończenie, 408

rola

menedżer projektu, 262

menedżer zasobów, 251

mistrza młyna, 46, 213

właściciel produktu, 46, 193

użytkownik, 125

zespół deweloperski, 46, 223

rozmiar

elementu rejestru, 132

historyjki, 116

kolejki, 82

wersji dystrybucyjnej, 148

zespołu, 233

rozmiary

elementów w rejestrze produktu, 51

zapotrzebowania, 79

rozmowa, 114

rozpoznanie inwentarza, 80

ryzyko procesu deweloperskiego, 76

S

samoorganizacja, 226

samospełniająca się przepowiednia, 73

Scrum, 25, 34

historyjki użytkownika, 113

oceny, 147

prędkość zespołu, 147

rejestr produktu, 130

scrumów, 244

sprinty, 91

wymagania, 109

zasady planowania, 273

zasady zwinności, 61

Scrummerfall, 66

sejsmograf emocji, 402

sekwencyjny procesie produkcyjny, 112

skala ceremonii, 88

skutki długu technicznego, 170–172

spadek

przewidywalności, 171

satysfakcji klientów, 172

spłacanie długu, 184–187

sposoby ujawniania długu, 182

sprint, 52, 91

definicja ukończenia, 103

krótki okres trwania, 94

niezmiennność celu, 98

ograniczenie czasowe, 92

stały czas trwania, 97

stały rozmiar sprintów, 97

stopień uszczegółowienia wymagań, 113

stosowanie najlepszych praktyk, 414

strategia „szybkiej porażki”, 326

strategie

aktywności, 306

napływu, 299

odpływu, 304

planowania portfela, 294

strumień wymagań, 141

synchronizacja zespołów, 246

szacunek względny, 52

szczegóły retrospekcji, 398

szef właścicieli produktu, 211

Ś

ścieżka naprzód, 415

środowisko

Cynefin, 38

Pragmatic Marketing, 206

Scrum, 45, 49, 92

T

tablica
 historyjek, 320
 zadań, 375
takt sprintowy, 98
techniki programistyczne, 374
temat, 117
tempo
 podtrzymywalne, 236
 pracy, 86
 przybywania, 300
 ubywania, 300
testowanie, 174
testy akceptacyjne, 107
tradycyjne formy produkcji, 61
tworzenie
 komponentów, 208
 mapy historyjek, 127
 mapy sprintów, 339
 rejestru produktu, 316
 wizji produktu, *Patr* planowanie produktu

U

ujawnianie długu technicznego, 180
ukończenie prac, 104, 105, 108
ułatwianie planowania, 94
umiejętności
 mistrza młyna, 216
 właściciela produktu, 199
 zespołu deweloperskiego, 226
upraszczanie planowania, 98
ustalanie wspólnego kontekstu, 400
ustalona data wersji, 341
ustalony zakres wersji, 345
uszczegóławianie rejestru, 131
używanie Scruma, 413

W

walidacja założeń, 76
warsztaty pisania historyjek, 125
warunki zadowolenia, 107, 115
WaterScrum, 66
wersja dystrybucyjna, 139, 278, 285, 303
 cechy obowiązkowe, 317, 338
 cel, 318
 ograniczenia, 333

wiedza potwierdzona, 75
wizja produktu, 283
właściciel produktu, 47, 50, 193
 definiowanie kryteriów, 197
 odpowiedzialność, 201
 pielęgnacja rejestru produktu, 197
 podejmowanie decyzji, 201
 udział w planowaniu, 196
 umiejętności domenowe, 199
 umiejętności interpersonalne, 200
 współpraca z interesariuszami, 199
 współpraca z zespołem, 198
 zarządzanie ekonomią, 195
wskazanie
 działań, 405
 sposobów, 405
wskaźniki wymagań, 110, 111
wybór
 ćwiczeń, 397
 działań, 406
 zadań do realizacji, 373
wydajność, 86
 jakość produktu, 86
 niepotrzebne formalności, 87
 tempo pracy, 86
wykonanie sprintu, 54, 367
 komunikowanie, 375
 praktyki techniczne, 374
 proces, 368
 uczestnicy, 367
 zarządzanie przepływem, 369
wykorzystanie zasobów, 82
wykorzystywanie okazji, 302
wykres
 Gantt, 276
 rozpalania, 349, 351, 379
 spalania, 348, 376
wyliczanie kosztu opóźnienia, 83
wymagania, 109–113, 124
wymagania niefunkcjonalne, 122
wymuszanie priorytetów, 93
wynik sprintu, 56
wzorzec ETC, 416

Z

- zalety
 - krótkich sprintów, 94
 - małych rozmiarów zapotrzebowania, 79
 - ograniczeń czasowych, 93
 - taktowania, 97
 - wdrożenia Scruma, 38
- założenie, 76
- zapisywanie historyjek, 113
- zarządzanie
 - czasem, 92
 - długiem technicznym, 176
 - obsługa, 183
 - przyrost długu, 177
 - ujawnianie, 180
 - ekonomią, 195, 261
 - inwentarzem, 64, 277
 - portfelem, *Patrz* planowanie portfela
 - przepływem, 369
 - sprintu, 140
 - wersji dystrybucyjnych, 139
 - wytwarzania wartości, 261
 - przyrostem długu, 177
- zasada
 - metod zwinności, 89, 90
 - najmniejszego zaskoczenia, 233
 - procesów sterowanych planem, 89, 90
- zasady
 - inspekcji, adaptacji i przejrzystości, 67
 - planowania, 273, 280
 - zwinności, 61
 - postęp, 84
 - praca częściowa, 78
 - przewidywanie i adaptacja, 68
 - wiedza potwierdzona, 75
 - wydajność, 86
 - zmiennosc i niepewność, 64
 - zasoby, 251
 - zastosowania
 - metody kanban, 42
 - Scruma, 35
 - zespół
 - budujący cechy, 240
 - budujący komponenty, 240
 - deweloperski, 48, 223
 - adaptacja produktu, 225
 - długotrwałość, 237
 - inspekcja, 225
 - komunikacja szerokopasmowa, 232
 - pielęgnacja rejestru produktu, 225
 - planowanie sprintu, 225
 - postawa muszkieterów, 231
 - przezroczysta komunikacja, 233
 - rozmiar, 233
 - samoorganizacja, 226
 - tempo pracy, 236
 - umiejętności typu T, 229
 - wykonanie sprintu, 225
 - zróżnicowanie, 229
 - wielofunkcyjny, 34
 - właścicieli produktu, 209, 211
 - zmiany, 72
 - zmienianie celu sprintu, 99
 - zmiennosc, 64
 - zobowiązanie, *commitment*, 49
 - zróżnicowanie zespołu deweloperskiego, 229
 - zwinna produkcja, 34
 - zwrot z inwestycji, 95, 280
 - zysk z cyklu życia, 294

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 

SPRAWDŹ, JAK SCRUM MOŻE ZMIENIĆ TWÓJ PROJEKT!

Tempo rozwoju współczesnych aplikacji jest zbyt duże, aby wciąż stosować skostniałe, stare metodyki zarządzania projektami. Obecnie należy błyskawicznie dostarczyć klientowi pierwszą wersję aplikacji i dopiero na podstawie jego uwag rozwijać oraz modyfikować program. Zwinne metodyki, w tym Scrum, pozwalają tego dokonać. Takie projekty nie zawsze są perfekcyjnie dopracowane, ale zazwyczaj pozwalają poznać opinię zleceniodawcy i lepiej zaplanować kolejne przebiegi — tzw. sprinty.

Ta książka to doskonałe źródło informacji o Scrumie. W trakcie lektury poznasz założenia tej metodyki oraz jej zalety i wady. W kolejnych rozdziałach zaznajomisz się z podstawowymi pojęciami oraz dowiesz się, jak szacować wydajność zespołu. Rozdział IV to gwóźdź programu, ponieważ omawia sprint — kluczowy element tej metodyki. Znajdziesz tu najlepsze metody planowania sprintu, założenia, których należy przestrzegać, oraz definicję jego ukończenia, a ponadto odkryjesz, co to jest dług techniczny oraz jak go pilnować. To jeszcze nie koniec! Kolejne rozdziały tej książki zostały poświęcone rolom odgrywanym przez różne osoby w czasie tworzenia projektu oraz jego poszczególnym etapom. Książka ta jest doskonałym źródłem informacji na temat metodyki Scrum oraz niezbędną lekturą dla osób rozważających jej wdrożenie w swoich projektach.

Dzięki tej książce dowiesz się:

- jak zwinnie zarządzać projektami
- jakie niespodzianki oraz pułapki kryje metodyka Scrum
- w jaki sposób oszacować wydajność zespołu
- jak odnieść sukces w zarządzaniu projektami

	KOD KORZYŚCI Sięgnij po więcej! ▶	
 helion.pl	ISBN 978-83-8322-617-0	
 HELION SA ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl	 9 788383 226170	
Cena: 99,00 zł		

 Addison-Wesley
Pearson Education