

TWORZENIE OPROGRAMOWANIA

W

30

DNI



Programuj zwinnie i szybko!

KEN SCHWABER, JEFF SUTHERLAND

Tytuł oryginału: Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust

Tłumaczenie: Katarzyna Żarnowska

ISBN: 978-83-246-7530-2

Copyright © 2012 by Ken Schwaber and Jeff Sutherland.
All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

All rights reserved. This translation published under license with the original publisher John Wiley & Sons, Inc.

Translation copyright © 2013 by Helion S.A.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, without either the prior written permission of the Publisher.

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

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

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

Wydawnictwo HELION dołożyło 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/top30d>
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

Spis treści

O autorach	9
Podziękowania	11
Wstęp	13
Część I Dlaczego każda firma na świecie może wyprodukować oprogramowanie w 30 dni?	15
<i>Prawdopodobnie jesteś sfrustrowany swoją firmą produkującą oprogramowanie. Chciałbyś, żeby kierujący nią ludzie byli bardziej elastyczni, lepiej rozumieli Twoje potrzeby i pomogli Ci osiągnąć większe zyski. Zastanowimy się, jakie są powody Twojej frustracji i jak temu zaradzić.</i>	
1 Kryzys oprogramowania: niewłaściwy proces daje niewłaściwe rezultaty	17
<i>Wiele firm produkujących oprogramowanie wykorzystuje proces, który tworzy straty, niekontrolowane ryzyko, niepewność, niespodzianki i niską wartość produktu. Zastanowimy się, dlaczego ten proces został wybrany oraz dlaczego gwarantuje on porażkę. Przyjrzymy się organizacjom, którym udało się od niego uwolnić.</i>	
2 Scrum: właściwy proces daje właściwe rezultaty	29
<i>Istnieje pewien proces odpowiedni do rozwijania oprogramowania. Kiedy Twój deweloperzy zaczną z niego korzystać, natychmiast zyskasz na produktywności, jakości, wartości, kontroli, przewidywalności oraz satysfakcji. W tym rozdziale przyjrzymy się, jak tego dokonać.</i>	

- 3 Spróbuj sam: program pilotażowy 41
Przeczytałeś nasze zapewnienie o lepszym sposobie rozwijania oprogramowania. Jednakże wiele osób z pewnością dawało Ci już takie zapewnienia, wyciągając od Ciebie przy okazji dużo pieniędzy i nie oferując nic w zamian. W tym rozdziale pokażemy Ci, jak sprawdzić, że nasze podejście działa, bez konieczności płacenia za to.
- 4 Co mogę zrobić? 53
Nauczyłeś się, jak uzyskiwać lepsze wyniki, i sam to wypróbowałeś. Podobają Ci się rezultaty i wiesz, co zalecić organizacji. W tym rozdziale omówimy, co możesz zrobić, by wdrożyć doświadczenia uzyskane w trakcie projektu pilotażowego.
- Część II Jak stworzyć oprogramowanie w 30 dni?** 57
Rozwijanie lepszego oprogramowania jest nie tyle trudniejsze, ile inne od wszystkiego, do czego byłeś przyzwyczajony. W tej części przyjrzymy się korzystnemu zestawieniu podejść, które pomogą Ci przejść z miejsca, w jakim jesteś teraz, do zwinności organizacyjnej.
- 5 Wstęp do metodyki Scrum 59
Nasz tajemniczy składnik, dzięki któremu lepiej wykorzystasz oprogramowanie, nazywa się Scrum. Tak, to aktywność zapożyczona z rugby, która pozwala na to, by piłka wciąż była w grze. W tym rozdziale omówimy metodę Scrum, jej działanie oraz powody, dla których działa.
- 6 Scrum na poziomie projektu 63
Najtrwalsze udoskonalenia w rozwijaniu oprogramowania zaczynają się na poziomie projektu. Możesz wykorzystać metodę Scrum, żeby przekonać się o jej użyteczności lub by Twoja inicjatywa odniosła sukces. Zobaczmy, co będziesz mógł powiedzieć Twoim programistom po przeczytaniu tego rozdziału.
- 7 Rozwiń potencjał Scruma 73
Sukces zwykle rodzi sukces. Im więcej inicjatyw rozwijanych za pomocą metody Scrum przynosi sukces, tym więcej ludzi będzie chciało z niej skorzystać. Zamiast zmieniać całą organizację, zobaczymy, jak zorganizować środowisko deweloperskie poza obecnym działem, który przynosi rozczarowania. W ten sposób będziesz w stanie wciąż zbierać najlepsze żniwo z rosnącej liczby projektów i wydań.
- 8 Scrum na poziomie przedsiębiorstwa 95
Scrum na poziomie projektu lub wydania zapewnia zwinność w zakresie inicjatywy, pozwala szybko reagować na pojawiające się możliwości lub powstające wyzwania. By uzyskać najbardziej znaczące korzyści, empiryczne podejście metody Scrum należy dopasować do całej organizacji. Zobaczmy, jak to zrobić oraz dlaczego niektóre podejścia dają krótkotrwałe efekty, a inne są zawsze aktualne.

9	Transformacja przedsiębiorstwa: gruntowna i trwała zmiana	101
	<i>Chcesz odchudzić swoją organizację, sprawić, by była bardziej wydajna i zwinna. Co więcej, chcesz, aby cechy te utrwaliły się i stały elementem kultury Twojej organizacji. Sprawdzimy, jak zmienić podejście przedsiębiorstwa, by móc to osiągnąć.</i>	
10	Stosowanie metody Scrum do wdrażania metody Scrum	111
	<i>Stworzyliśmy metodę Scrum, by rozwiązywać skomplikowane problemy, na przykład te dotyczące rozwijania oprogramowania. Okazało się, że Scrum jest użyteczną techniką, służącą do zarządzania zmianą organizacyjną, która również jest złożonym problemem. W tym przypadku także pojawiają się korzyści związane z przejrzystością, usuwaniem strat, kontrolą ryzyka oraz przewidywalnością. W tym rozdziale zapoznamy się z takim właśnie wykorzystaniem metody Scrum.</i>	
A	Terminologia	117
	<i>Powoli i stopniowo wprowadziliśmy trochę nowej terminologii. Ten dodatek jest indeksem nowych terminów.</i>	
B	Przewodnik po metodzie Scrum	121
	<i>Przeczytaj kanoniczny przewodnik po metodzie Scrum, jej rolach, artefaktach oraz wydarzeniach. To jest biblia Scruma.</i>	
C	Strategia działania prężnego przedsiębiorstwa	137
	<i>Ten dodatek prezentuje bardziej szczegółowy plan zmian w przedsiębiorstwie omówionych w rozdziale 10.</i>	
	Skorowidz	163

6

Scrum na poziomie projektu

KORZYSTAJ Z METODY Scrum wtedy, kiedy jest to konieczne, na przykład w przypadku nagłej potrzeby lub katastrofального przebiegu prowadzonego projektu. Dzięki informacjom zawartym w tym rozdziale dowiesz się, jak bezzwłocznie zacząć. Nauczysz się, jak zbudować wartościowy komponent w ciągu 30 dni.

Wpływ na organizację nie będzie brany pod uwagę. Najważniejsze są krótkoterminowe korzyści, nie udogodnienia dla całej firmy. Zyski zaczną gwałtownie wzrastać. Projekt według metody Scrum jest realizowany w izolacji od tradycyjnych praktyk i procesów, przy użyciu tylko takich metod, które nadają wartość wykonywanej pracy.

Scrum: z dołu do góry i w ukryciu

W ciągu ostatnich 20 lat wiele projektów korzystających z metody Scrum było przeprowadzanych w niższych partiach organizacji, w ukryciu. Zespół projektowy postanawiał wypróbować metodę Scrum i uzyskiwał imponujące rezultaty. Później korzystał z metody kolejny zespół i wkrótce wiele jednostek w całej firmie rozwijało oprogramowanie szybciej i częściej. Metoda Scrum w szybkim tempie rozprzestrzeniała się w całej organizacji.

Określamy ten model mianem Scrum PRN¹. Tak jak decyzja o przyjmowaniu lekarstw z adnotacją p.r.n. leży w gestii pacjenta lub opiekuna, tak Scrum PRN jest wykorzystywany w razie nagłej potrzeby. Może być wdrożony natychmiast. Dopuszcza się modyfikacje metody w przypadku, gdy pojawiają się nowe możliwości lub konieczne jest zażegnanie kryzysu. Do użycia metody Scrum PRN nie jest wymagana specjalna zgoda. Natychmiastowa konieczność stworzenia oprogramowania jest wystarczającym argumentem.

¹ Jedno z tłumaczeń PRN, lub *pro re nata*, brzmi „jeśli wymagają tego okoliczności”. Dopisek *p.r.n.* na recepcie oznacza „brać w razie potrzeby” lub „brać, jeśli to konieczne”.

Zyski i odkrycia

Koszt 30-dniowego sprintu może wahać się pomiędzy 160 000 zł a 480 000 zł, w zależności od wielkości zespołu Scrumowego, pensji poszczególnych członków oraz kosztów dodatkowych. W zamian otrzymujemy:

- *Wiedzę na temat poziomu umiejętności deweloperów:* Ten proces pozwoli nam ocenić, czy deweloperzy potrafią zbudować oprogramowanie, którego oczekujemy, oraz ile są w stanie stworzyć w trakcie jednego sprintu.
- *Funkcjonalność:* Część funkcjonalności, bez względu na to, jak niewielka, może być wykorzystana na końcu każdego sprintu. Ta funkcjonalność jest dodatkiem do każdego dostarczonego wcześniej przyrostu.
- *Możliwość modyfikacji planu:* Inwestycje mogą zostać poddane ocenie i powtórnemu planowaniu przed każdym sprintem. Nazywa się to planowaniem *just-in-time* (z ang. „dokładnie na czas”). Modyfikacja planu jest przeprowadzana w odpowiedzi na oczekiwane zmiany. W ten sposób jest eliminowany czas przeznaczany zwykle na planowanie elementów, które nie zostaną wykonane.

Projekty Scrum demaskują zarówno rzeczy, których oczekiwaliśmy, jak i te niespodziewane. Dzięki Scrumowi wiemy, co się dzieje, i możemy podejmować przemyślane decyzje dotyczące następnych działań. Zaletą jest możliwość kontroli, zarządzania inwestycją krok po kroku.

Zarządzanie pracą: wykres spalania

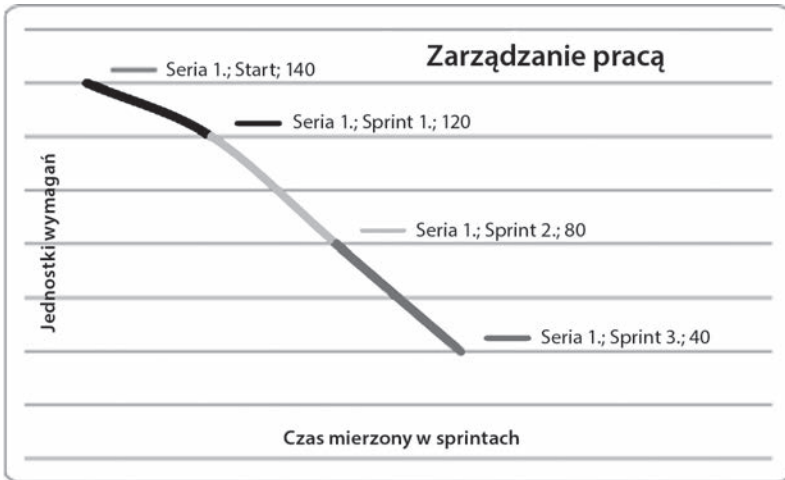
Jedną z największych zalet metody Scrum są informacje, których dostarcza. Kadra zarządzająca może korzystać z tych informacji w celu maksymalizacji wartości oraz zarządzania ryzykiem.

Pod koniec każdego sprintu praca deweloperów jest kontrolowana, sprawdza się liczbę zrealizowanych i gotowych do użycia elementów. Dzięki temu można ocenić postęp względem ustalonego celu i na tej podstawie ostrożnie prognozować przyszłość.

Do zarządzania projektem potrzebujemy trzech zmiennych. Na pierwszym miejscu są wymagania, czyli funkcjonalności, dzięki którym zrealizujemy naszą wizję. Niektóre wymagają małego wysiłku, inne średniego, a kolejne dużego. Druga zmienna to czas, mierzony w sprintach. Trzecią wartością jest wykonana praca, którą mierzymy w użytecznych elementach dostarczonej funkcjonalności.

Wymagania przekształcone w przyrosty funkcjonalności, obserwowane przez dłuższy czas, dają nam pojęcie o trendach. Na przykład na początku pierwszego sprintu zespół deweloperki określił rozmiar wymagań na 140 jednostek pracy. Programiści dostarczyli 20 jednostek pracy w pierwszym sprincie oraz po 40 jednostek w drugim i trzecim sprincie. Te wartości łatwo jest śledzić za pomocą wykresu spalania. Wykres spalania mierzy wymagania w jednostkach pracy.

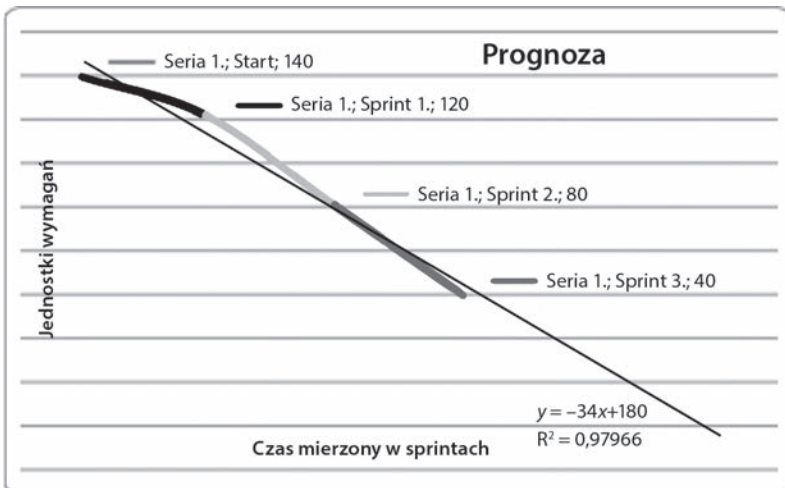
Pozostała do wykonania praca jest przeliczana pod koniec każdego sprintu. Musi się ona równać liczbie jednostek pracy prognozowanych na początku sprintu pomniejszonej o jednostki ukończone, zamienione w danym sprincie na przyrosty funkcjonalności. Wykres spalania dla przykładowego projektu wygląda jak ten pokazany na rysunku 6.1.



Rysunek 6.1. Przykładowy wykres spalania

Daje on obraz postępu prac.

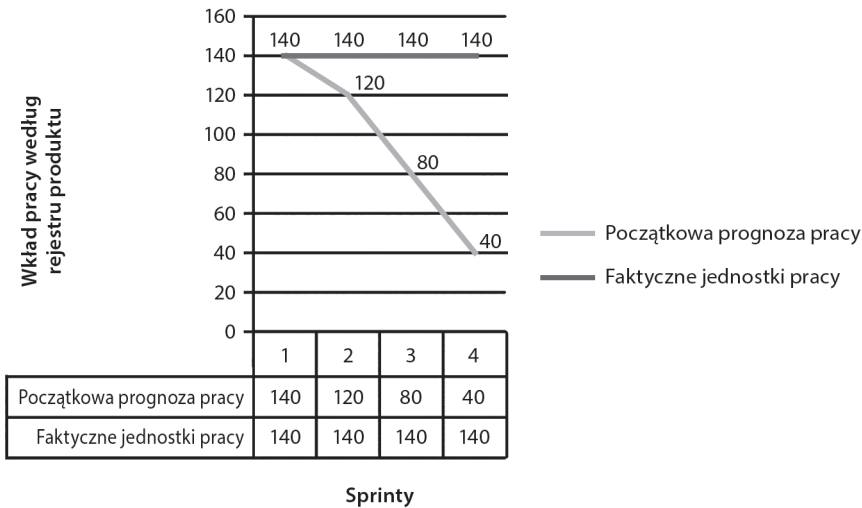
Do prognozowania można wykorzystać poprzednie średnie. W pierwszych trzech sprintach średnia zrealizowanych jednostek wymagań wynosiła 33,3 przy standardowym odchyleniu równym 11,5. Linia trendu jest zaznaczona na rysunku 6.2.



Rysunek 6.2. Przykładowa prognoza

Ten wykres spalania pozwala nam założyć, że projekt zostanie ukończony w piątym sprincie. Tworzenie oprogramowania rzadko jednak jest takie proste. To złożona praca, w której więcej jest niewiadomych niż elementów znanych. Prognozowanie w rozwoju oprogramowania jest bardzo ryzykowne. Każdego dnia wpływ na prognozy może mieć aktualna kondycja członków zespołu, stabilność użytej technologii oraz sytuacja na rynku, gdzie nowe wymagania mogą pojawić się niespodziewanie. Im dalej w przyszłość wybiega linia trendu, tym mniej jest wiarygodna.

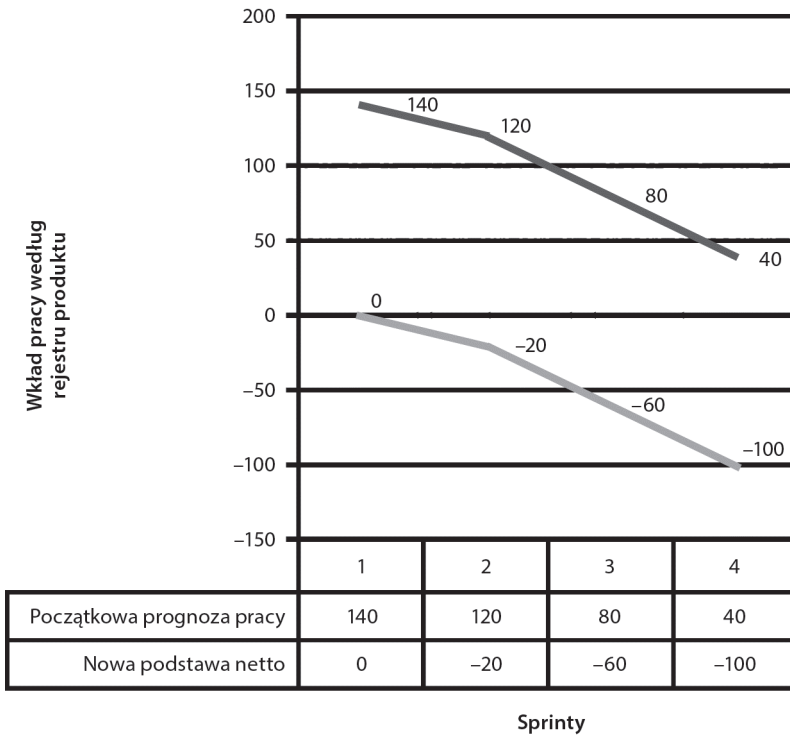
W miarę jak prace nad projektem stają się coraz bardziej zaawansowane, pojawiają się nowe wymagania. Klienci zgłaszają nowe potrzeby. W trakcie inspekcji przyrostów ujawniają się nowe możliwości. Załóżmy, że na nasz projekt składa się 140 jednostek wymagań. Gdybyśmy w trzech pierwszych sprintach dodali do rejestru odpowiednio 20, 40 i jeszcze 40 nowych wymagań, wykres spalania byłby płaski, mylnie sugerując, że nie została wykonana żadna praca (rysunek 6.3). Jest to spowodowane tym, że do rejestru w każdym sprincie dodano dokładnie tyle samo nowych wymagań, ile funkcjonalności udało się ukończyć zespołowi deweloperskiemu.



Rysunek 6.3. Spalanie faktyczne kontra prognozowane

Aby zachować użyteczność wykresu, obliczono nową podstawę „netto”: [(podstawa wyjściowa + dodatkowe wymagania) – (ukończone wymagania) = nowa podstawa netto]. Ta nowa podstawa netto jest widoczna na rysunku 6.4. Pokazuje nam ona, że projekt najprawdopodobniej zostanie ukończony znacznie później niż w przewidywanym piątym sprincie.

Korzystając z metodyki Scrum, wiemy, kiedy kolejne sprinty tracą na wartości i należy przestać je finansować. W tym momencie komponent oprogramowania jest gotowy do wydania, mamy również informację zwrotną od użytkowników. Dodatkowe funkcjonalności, o które proszą użytkownicy, zazwyczaj nie wchodzi w skład pierwotnej wizji projektu, stworzonej przez zespół Scrumowy. Na podstawie tych informacji kolejne wydanie zostaje zmodyfikowane poprzez dodanie wymagań, o które prosili użytkownicy, oraz usunięcie z rejestru elementów niepotrzebnych.



Rysunek 6.4. Podstawa netto odzwierciedla zmiany w rejestrze produktu

Standish Group ocenia, że 50 procent funkcjonalności oprogramowania jest używane rzadko lub wcale². Na przykład na olbrzymiej stronie *hp.com* 80 procent klientów korzysta z zaledwie 14 procent funkcjonalności³. Aby zoptymalizować wartość, właściciel projektu musi podjąć decyzję o tym, kiedy należy zakończyć przeprowadzanie sprintów, a tym samym zatrzymać prace nad wytwarzaniem oprogramowania o niskiej wartości. Przy zastosowaniu tej taktyki czas pracy nad projektem może zostać zredukowany nawet do 40 procent. Możesz osiągnąć tę produktywność poprzez uważną obserwację wartości komponentów, które wyprodukowałeś.

Nie ignoruj złożonych projektów — miej oczy zawsze otwarte

Wiemy już, że dzięki metodzie Scrum możemy łatwiej sprostać wyzwaniom i wykorzystać nadarżające się okazje. Przed rozpoczęciem pierwszego sprintu zwykle chcemy znać czas trwania projektu oraz jego koszt. Możemy dokonać wstępnych szacunków na podstawie rezultatów pierwszych sprintów. Wyobraźmy sobie, że budujemy 20 jednostek funkcjonalności

² J. Johnson, *Chaos Manifesto: The Laws of CHAOS and the CHAOS 100 Best PM Practices*, The Standish Group, Boston 2011, s. 25.

³ Takie dane otrzymał Ken Schwaber od Johna Sawyera 10 listopada 2009, podczas prezentacji metody Scrum dla firmy Hewlett-Packard w Palo Alto, w Kalifornii (USA).

w dwóch sprintach. Zakładamy, że cały wymyślony przez nas system składać się będzie z 220 jednostek funkcjonalności. Musimy zatem zbudować jeszcze 180 jednostek. Przyjmując 20 wykonanych jednostek na sprint, możemy założyć, że produkt będzie gotowy w ciągu kolejnych 9 iteracji. Jeśli dodajemy lub odejmujemy funkcjonalności w trakcie prac, musimy uwzględnić to we wskaźniku jednostek wykonanych w każdym sprincie.

Należy oczywiście zachować szczególną ostrożność przy prognozowaniu przyszłości na podstawie przeszłych działań. Szacowanie to proces konstruowania nowych jednostek danych. Jest on podobny do procesu interpolacji, który tworzy nowe jednostki pomiędzy już znanymi. Rezultaty szacowania są jednak mniej znaczące i bardziej niepewne. Wiemy, że tworzenie oprogramowania jest procesem złożonym i zawiera więcej niewiadomych niż pewników. Możemy dokonywać szacunków, muszą być one jednak weryfikowane. Pod koniec każdej iteracji weryfikujemy naszą faktyczną, a nie szacowaną pozycję. Rzeczywistość jest pewniejsza niż oczekiwania.

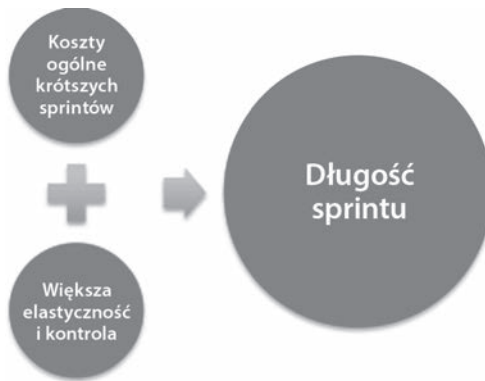
Problem z nowymi możliwościami polega na tym, że są nowe. Można je wykorzystać albo przez stworzenie czegoś od podstaw, albo przez dodanie czegoś nowego do starego elementu. W każdym z tych przypadków jest wiele możliwości, nad którymi trzeba się zastanowić, rozwiązań do przystosowania oraz oprogramowania do stworzenia lub zmiany jego zastosowania. Tradycyjnie, zanim zaczniemy pracę, musimy pomyśleć. To planowanie wymagań, którego rezultatem jest dokument produktu lub dokument wymagań marketingowych. Problem polega na tym, że nie wiemy dokładnie, czego chcemy. Nawet jeśli mamy konkretne pomysły, najlepsze podejście klaruje się w trakcie procesu. Ponieważ naturą skomplikowanych problemów jest to, że więcej w nich niewiadomych niż znanych rzeczy, planowanie jest niezwykle trudne i na tym etapie zdarza się sporo błędów i pominięć. W metodzie Scrum plany tworzymy już w trakcie prac. Przewidywalność jest rezultatem decyzji podejmowanych w odpowiednim czasie, na podstawie realnych wyników. Mimo że szacujemy koszty i czas trwania na początku projektu, w jego trakcie bezustannie oceniamy efekty. W tradycyjnych metodach również możemy prognozować czas i koszty na początku projektu, nie jesteśmy jednak w stanie uzyskać żadnych użytecznych danych, pomocnych w dostosowaniu go do zmieniających się realiów, zanim projekt nie zostanie ukończony w przynajmniej 90 procentach.

Długość sprintu

Organizacje, które korzystają z metody Scrum, najczęściej wykorzystują 30-dniowe iteracje, chociaż Scrum pozwala, by sprinty były krótsze. Te dłuższe są wykorzystywane w bardziej stabilnych sytuacjach, krótsze sprawdzają się lepiej w tych skomplikowanych i wymagających. Odpowiednią długość sprintu dobiera się do konkretnego projektu, biorąc pod uwagę następujące dane (rysunek 6.5):

- koszty ogólne krótszych sprintów,
- większą elastyczność i kontrolę.

Sprinty nie mogą trwać dłużej niż miesiąc.



Rysunek 6.5. Zmienne mające wpływ na długość sprintu

Powody, dla których warto wybrać krótsze sprinty

Cztery jednodniowe iteracje zapewniają większą elastyczność i kontrolę niż jedna 30-dniowa. Oto niektóre ze zmiennych, mogących mieć wpływ na Twój wybór długości sprintu:

1. *Niestabilny rynek*: Długość sprintu określa, jak często można zmieniać kierunek rozwoju produktu. Rynek dla produktu może być nowy lub zmienny. Inne organizacje i konkurencja również wprowadzają nowe produkty. Możesz zachować większą elastyczność, by zapewnić szybką reakcję na nowe możliwości. Możesz też wstrzymać inwestycje w nowe funkcjonalności, dopóki nie pojawi się okazja do zmiany kierunku rozwoju.
2. *Niestabilny zespół*: Zespół Scrumowy może potrzebować nawet roku, by uzyskać wprawę, czasem nie udaje się to nigdy. Krótsze sprinty dają wszystkim możliwość bliższego spojrzenia na dynamikę zespołu, dzięki czemu problemy mogą być sprawnie rozwiązane, a produktywność zwiększona.
3. *Niestabilna technologia*: Za każdym razem, kiedy korzysta się z nowych technologii, już na początkowym etapie są potrzebne informacje na temat ich użyteczności. W przypadku nowych produktów przydatność nowych technologii często decyduje o sukcesie. Wypróbuj je na małym komponentie funkcjonalności. Zobacz, jak działają, czy działają oraz czy oferują odpowiednie wsparcie dla systemu. Na przykład jeśli produkt ma obsługiwać wielu użytkowników jednocześnie lub musi być dobrze zabezpieczony, należy zorientować się we wczesnej fazie rozwoju, czy dana technologia jest do tego odpowiednia. Jeśli nie, możesz przeformułować projekt albo go odwołać.
4. *Ustalenie stabilnej prędkości*: Najlepszym sposobem na oszacowanie kosztów projektu jest przegląd produktywności podobnych projektów z przeszłości, identycznych technologii oraz zespołów, które pracowały ze sobą przez dłuższy czas. Jeśli nie jesteśmy w stanie tego zrobić, kolejną rzeczą, która może nam pomóc w prognozowaniu kosztów, jest przeprowadzenie kilku krótszych sprintów. W miarę jak deweloperzy uczą się pracować ze sobą i z technologią, zostaje ustalona stabilna prędkość lub też liczba funkcjonalności, które będą oni w stanie zbudować w trakcie każdego sprintu. Kiedy już ustabilizuje się to na odpowiednim poziomie,

będziesz mógł przewidzieć możliwości zespołu programistycznego względem zaplanowanej pracy i w ten sposób określić koszty oraz datę dostarczenia produktu. Pamiętaj jednak, że prognoza nie daje żadnej gwarancji.

5. *Zapewnienie możliwości uczenia się:* Ludzie lubią odnosić sukces. Kiedy ktoś uczy się jeździć na rowerze, nartach lub łyżwach, zwykle na początku próbuje przez krótki czas. Można wtedy dokonać oceny i wprowadzić zmiany, a następnie spróbować ponownie. Krótsze sprinty ułatwiają naukę.
6. *Kontrola ryzyka:* Pożądany zwrot z inwestycji może być nieosiągalny. Kiedy rynek jest zmienny lub nieznan, technologie niesprawdzone, a pracownicy nowi, krytyczne znaczenie ma zebranie informacji dotyczących kosztów i zysków na wczesnym etapie prac. Takie dane można uzyskać w krótszych sprintach, które umożliwią także częstszą kontrolę projektu. W ten sposób, nawet jeśli okaże się, że niezbędna jest zmiana lub rezygnacja z projektu, dokonujemy mniejszej inwestycji.

Ogólnie rzecz biorąc, dłuższe sprinty są wykorzystywane w sytuacjach, w których jest mniejsze ryzyko, niepewność, zmienność. Na przykład w przypadku produktu lub systemu, który będzie wykorzystywany wewnętrznie. Dłuższe sprinty sprawdzą się też, gdy zależy nam na zwiększeniu możliwości systemu lub jego kosztów, a nie na wyprzedzeniu konkurencji. W podobnych przypadkach najlepsze będą sprinty 30-dniowe.

Powody, dla których nie warto wybierać krótszych sprintów

Dwa dwutygodniowe sprinty kosztują więcej niż jeden 30-dniowy. Pojawia się też dwa razy więcej spotkań dotyczących planowania, przeglądu i retrospekcji. Zespół Scrumowy jest zmuszony do formułowania nowego projektu dwa razy częściej. Również sprinty będą musiały się rozpętać i zwalniać dwa razy częściej.

Ceną za krótsze sprinty jest wzrost czasu potrzebnego na planowanie i przegląd. Można to nazwać kosztem możliwości lub kosztem ubezpieczenia. Rysunek 6.6 pokazuje przykładowe dane dotyczące czasu planowania w sprintach różnej długości. Pokazano na nim liczbę spotkań koniecznych w trakcie 30 dni pracy, porównując jeden 30-dniowy sprint, dwa dwutygodniowe i cztery jednodniowe. Mianem „spotkania w trakcie sprintu” zostały określone zebrania dotyczące planowania, przeglądu i retrospekcji. Koszty codziennych zebrań pozostają takie same. Koszty zespołu Scrumowego wyniosły 650 000 zł dla 30-dniowego sprintu.

Długość sprintu	Czas na spotkania w trakcie sprintu	Liczba dni przeznaczonych na spotkania	Wzrost kosztów
30 dni	2 dni	2 dni	
2 tygodnie	1,8 dnia · 2	3,6 dnia	34 000 zł
1 tydzień	1,5 dnia · 4	6 dni	126 000 zł

Rysunek 6.6. Koszty krótszych sprintów

Biorąc pod uwagę większą możliwość kontroli, przewidywalność i elastyczność, wiele organizacji dochodzi do wniosku, że koszty krótszych sprintów są akceptowalne.

Nie próbuj przeprowadzać sprintów o takich długościach

W przypadku sprintów krótszych niż tydzień czas pozostały do przerobienia wymagań na użyteczne funkcjonalności jest zwykle niewystarczający. Zespołowi programistów trudno jest w ciągu tygodnia zbudować komponent, który byłby użyteczny lub dostarczał niezbędnych informacji marketingowych.

Zalecamy również, by sprinty nie były nigdy dłuższe niż 30 dni lub jeden miesiąc. Przy sprintach trwających dłużej niż 30 dni (lub miesiąc) mogą pojawić się następujące problemy:

1. Interesariusze tracą zainteresowanie i zapominają o projekcie.
2. W miarę wzrastania liczby wymagań ogólna złożoność projektu rośnie znacznie gwałtowniej niż linearnie. Aby skutecznie zarządzać wzmożoną złożonością, nie zapominając jednocześnie o pierwotnych założeniach, zespół deweloperski potrzebuje większej ilości dokumentacji i lepszej infrastruktury.
3. Ilość informacji, które należy przejrzeć i zapamiętać, a potem na ich podstawie podjąć decyzję, tłumi efektywność krótkich zebrań stosowanych w metodzie Scrum.

Wszystkie sprinty w projekcie powinny być tej samej długości

Jeśli to możliwe, postaraj się, by wszystkie sprinty w projekcie deweloperskim były tej samej długości. Praca deweloperów daje najlepsze rezultaty, kiedy zachowuje określone tempo, rytm. Po sześciu 30-dniowych sprintach programiści wypracowują wzorzec, na podstawie którego będą dalej pracować. Jeśli skrócimy czas trwania sprintu do jednego tygodnia, na początku deweloperzy będą korzystać z wypracowanego wzorca 30-dniowego rozkładu pracy, co spowoduje, że projekt się przeciągnie. Po zmianie długości sprintu deweloperzy zwykle jeszcze w trzech kolejnych zakładają większą ilość pracy, niż są w stanie wykonać. Zespół musi wypracować nowe tempo za każdym razem, kiedy zmienia się czas trwania sprintu, co powoduje spadek produktywności. Stałe długości sprintów pozytywnie wpływają na produktywność.

Może się oczywiście zdarzyć, że zmiana czasu trwania sprintu jest uzasadniona i wynika z niezadowalających efektów. Być może zespół deweloperski nie współpracował tak, jak powinien, wymagania nie były zrozumiałe, zbyt wiele czasu poświęcono jednemu problemowi lub też pojawiły się nowe zagadnienia. Problemy stają się przejrzyste znacznie szybciej w trakcie krótszych sprintów. Można wtedy sprawniej przekierować pracę lub dokonać zmian w zespole. W ten sposób minimalizujemy straty. Możesz zmieniać długości sprintów, jednak staraj się to robić tylko wtedy, gdy to konieczne. Jeśli zbyt często będziesz dokonywał zmian, członkowie zespołu zdekoncentrują się, tracą jasność przekazu i przestaną rozumieć, co jest możliwe do zrobienia. Tworzenie oprogramowania jest procesem złożonym i podatnym na nieprzewidywane sytuacje, staraj się więc wprowadzać uproszczenia wszędzie, gdzie jest to możliwe.

Przykłady użycia metody Scrum PRN w projektach

Fidelity Investments wykorzystało Scrum w 1997 roku do wprowadzenia internetowych udogodnień dla swoich klientów. Klienci Charlesa Schwaba oraz E-Trade mogli już zarządzać swoimi oszczędnościami przez internet, podczas gdy klienci Fidelity nie mieli takiej możliwości. W tym czasie Fidelity twardo trzymało się kaskadowego modelu zarządzania projektami. Firma wielokrotnie podejmowała próby wprowadzenia udogodnień internetowych, za każdym razem ponosząc porażkę. W przyływie desperacji Fidelity zwróciło się w stronę metody Scrum. W ciągu zaledwie kilku miesięcy pierwsza wersja *Fidelity.com* była sprawna i gotowa do użytku. Po 18 miesiącach od rozpoczęcia prac inwestorzy równie chętnie korzystali z *Fidelity.com* i z usług konkurencji. Ogłoszono sukces i odłożono metodę Scrum na półkę.

Firma nauczyła się korzystać z metody Scrum w decydujących momentach. W siedmiu kolejnych przypadkach, kiedy Fidelity potrzebowało zbudować przełomowy komponent oprogramowania, firma przeprowadzała projekty według metody Scrum. Organizacja nie wykorzystwała jednak do końca potencjału tej metody. Każdy projekt Scrum mógł być bardziej efektywny od poprzedniego. Jednakże firma postanowiła korzystać z tej metody tylko w krytycznych przypadkach.

Następny rozdział

W następnym rozdziale wytłumaczymy, jak progresywnie zwiększać korzyści płynące z metody Scrum za pomocą ciągłego, mierzalnego podejścia, które minimalizuje zakłócenia, maksymalizując jednocześnie zyski dla organizacji.

Skorowidz

A

adaptacja, 125
akumulacja długu technologicznego, 92
aplikacja Adobe Premiere Pro CS3, 89
artefakty
 inżynieryjne, 159
 w Scrumie, 133, 159
atributy Scrum, 141
autonomia zespołu, 37

B

badanie opinii, 78
budowanie więzi, 75

C

cel sprintu, 117, 131
codzienne zebrania Scrum, 117, 131
cykl życia produktu, 160
częstość, 31
członkowie zespołu, 51, 55, 75

D

dane metryczne, 152
 procesowe, 153
 projektowe, 153
definicja pracy wykonanej, 87
deweloper, 36, 44, 59, 117
diagram Gantta, 40
dług technologiczny, 87, 89, 92

długość sprintu, 68–71
dobór metod pracy, 56
działanie Scrum, 32, 33

E

ekspansja organizacyjna, 151
emergencja, 117
empiryczny, 117
empiryzm, 53

F

fabryka oprogramowania, 73
firma
 Adobe, 89
 Carbonite, 97
 Curaspan, 45
 Fidelity, 72
 F-Secure, 54
 Iron Mountain, 46, 113
 Kronos, 55
 Primavera, 56, 96
 PTC, 25
 PwC, 39
 SeaChange, 112
funkcjonalności oprogramowania, 67

I

infrastruktura
 narzędzi, 158, 160
 pracownicy oprogramowania, 75

inspekcja, 125
 inspekcja rezultatów, 29
 inspekcje i adaptacje, 29, 35
 iteracja, 31, 118
 iteracje 30-dniowe, 68

J

jakość, 118
 jakość oprogramowania, 79, 91

K

komunikacja, 105, 107, 157, 159
 kontrola, 36
 kontrola nad ryzykiem, 40
 kontynuacja sprintów, 116
 koszt
 30-dniowego sprintu, 64
 operacyjny, 82
 sprintu, 70
 własności, 81

L

liczba
 błędów, 90
 defektów, 79
 linearne sekwencjonowanie pracy, 38
 linia trendu, 65, 118
 lista wymagań, 31

M

menedżer
 pracowni, 74
 programu pilotażowego, 45, 48
 projektu, 51
 metoda empiryczna, 29, 33, 42, 53
 kontrola, 36
 przewidywalność, 36
 zarządzanie ryzykiem, 36
 metoda Scrum, *Patrz* Scrum
 metoda zwinna, 19
 mistrz młyna, Scrum Master, 59, 118, 127, 146
 model kaskadowy
 problemy, 33, 34
 model predykcyjny, 20, 51
 ludzie, 22
 problemy PTC, 25
 technologia, 22
 wymagania, 22
 model przyrostowy, 31–34, 118

modele rozwijania oprogramowania, 19
 monitorowanie postępu, 134, 135

O

ocena postępu, 152
 okres 30-dniowy, 29
 opracowanie wymagań, 160
 osiągnięcie wpływu, 152

P

planowanie, 22, 68, 77, 157
 just-in-time, 64
 pracy, 60
 sprintu, 60, 116, 130
 wydań, 161
 poczucie bezpieczeństwa, 55
 podstawa, baseline, 118
 praca wykonana, 87
 pracownia Scrum, 93
 infrastruktura, 75
 szkolenie, 75
 warunki użytkowania, 76
 zestaw metryczny, 78
 problemy
 modelu kaskadowego, 33
 w produkcji oprogramowania, 20, 22
 proces
 empiryczny, 142
 kaskadowy, waterfall, 118,
 Patrz także model predykcyjny
 planowy, 142
 sekwencyjny, *Patrz* model predykcyjny
 transformacji, 115
 produktywność, 79, 118
 produktywność deweloperów, 61
 prognoza, 65, 119
 program pilotażowy, 41, 43, 149
 ewaluacja, 50
 iteracje, 49
 lista funkcjonalności, 47
 przekaz, 45
 przyrosty funkcjonalności, 50
 testowanie, 48
 zespół, 44, 50
 projekt
 Scrum, 64
 Sentinel, 18–20
 transformacji przedsiębiorstwa, 101
 zwinny, 19

projektowanie wspomagane komputerowo, 25
 projekty programistyczne
 ludzie, 24
 technologia, 24
 wymagania, 24
 prototyp, 40
 przegląd
 metody Scrum, 141
 sprintu, 61, 116, 119, 132
 przejrzystość, 29, 55, 82–87, 119, 125
 przekaz firmy Curaspan, 45
 przeszkody przy wdrażaniu, 154
 przewidywalność, 36
 przygotowanie do Scruma, 145
 przyrost, 31, 119, 135
 przyrost funkcjonalności, 83
 przyrosty
 niekompletne, 86
 skończone, 83
 PTC, Parametric Technology Corporation, 25
 pulpit
 pracowni, 81
 produktywności i jakości, 80
 projektu, 79
 wartości i zwrotu z inwestycji, 80
 punkt funkcyjny, 119

R

raport
 CHAOS, 17
 Standish Group, 20
 raportowanie, 77
 raportowanie projektu, 160
 rejestr
 produktu, 67, 119, 133, 161
 sprintu, 119, 134
 wymagań, 30
 retrospekcja sprintu, 61, 119, 132

S

samoorganizacja zespołu, 51, 56, 120
 Scrum, 9, 59, 120, 125
 artefakty, 59
 atrybuty, 141
 na poziomie możliwości, 57, 73
 na poziomie projektu, 57, 63
 na poziomie przedsiębiorstwa, 58, 95
 pracownia, 73–82
 role, 59

wdrażanie, 95–98, 101, 111, 148–154
 wydarzenia, 59
 Scrum PRN, 63, 72
 skalowanie Scruma, 155
 spalanie, 66
 sprint, 59–61, 116, 120, 129
 planowanie, 60
 przeгляд, 61, 132
 rejestr, 134
 retrospekcja, 61, 132
 wybór długości, 68–71
 stosowanie Scruma, 112
 strategie wdrażania Scruma, 148–154
 struktura metody Scrum, 124
 studium przypadku
 projekt Sentinel, 18
 PTC, 25
 subtelna kontrola, 38
 szybkość, velocity, 120

T

testowanie, 76
 tradycyjne zarządzanie projektem, 159
 transfer nauki, 38
 transformacja przedsiębiorstwa, 101
 czynności, 102, 107
 komunikowanie wizji, 105
 konsolidacja osiągnięć, 108
 korzyści, 103
 opór bierny, 106
 osadzanie, 109
 rozszerzanie działań, 107
 wizja, 104
 zespół, 103
 transformacja zespołów wdrożeniowych, 115
 trend, 64
 tworzenie
 przekazu, 45
 zespołu, 44, 60

U

użytkowanie pracowni, 76

W

wartość oprogramowania, 79
 wczesne testowanie, 160
 wdrażanie Scruma, 95–98, 101, 111, 148–154
 wizja, 120
 właściciel produktu, Product Owner, 59, 120, 126

wydajność zespołu, 37, 39

wykres

spalania, 35, 65, 66, 120

Stacey, 23, 24

wymagania, 31, 120

wytwarzanie oprogramowania, 18, 24, 29, 56

Z

zarządzanie

cyklem eksploatacji, 25

jakością, 77

pracą, 64

projektem, 34, 37

rejestrem, 160

ryzykiem, 36

zespołem Scrumowym, 59

zasady metody Scrum, 142

zespół

deweloperski, 120, 126

projektowy, 37

Scrumowy, Scrum Team, 59, 120, 125

transformacyjny, 106, 114

wdrożeniowy, 115

zwiększanie wydajności zespołu, 37–39

zwinność, 39

zwinność oprogramowania, 144

zwrot z inwestycji pracowni, 81

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



- 1. ZAREJESTRUJ SIĘ**
- 2. PREZENTUJ KSIĄZKI**
- 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**

TWÓJ KLUCZ DO SUKCESU!

Czy Twoja firma jest w stanie stworzyć oprogramowanie w 30 dni? Jeśli uważasz, że nie, zapewne wciąż stosujesz stare i niezbyt poręczne narzędzia. Jeżeli tylko nauczysz się zwinnie zarządzać swoim projektem, ten cel jest do osiągnięcia. „Zwinny” to słowo, które zrobiło furorę w świecie menedżerów projektu. Dzięki takiemu podejściu do zarządzania stworzysz aplikację dopasowaną do potrzeb użytkowników i na pewno zdążysz na czas. Warto spróbować!

Najpopularniejszą ze zwinnych metodyk jest Scrum. Ta książka została napisana właśnie przez twórców Scruma. Powstała, aby zaznajomić Cię z procesem tworzenia oprogramowania z użyciem tej metodyki. Pokaże Ci, jak zacząć przygodę z nowym projektem i sprawić, by odniósł sukces. W trakcie lektury dowiesz się, jak wdrożyć Scrum w Twojej organizacji, oraz poznasz role wyznaczane poszczególnym członkom projektu. Ponadto zaznajomisz się z kolejnymi etapami całego procesu i spojrzysz na tę metodykę z punktu widzenia całego przedsiębiorstwa. Książka ta jest doskonałym przewodnikiem, który pozwoli Ci dokonać prawdziwego przełomu w pracy nad projektami. Po jej lekturze i wdrożeniu metodyki Scrum będziesz znacznie bliżej sukcesu!

ZARZĄDZAJ PROJEKTAMI:

ZWINNIE

ELASTYCZNIE

SKUTECZNIE

Z UŻYCIEM METODYKI SCRUM

helion.pl
księgarnia
internetowa

Nr katalogowy: 14903



Księgarnia internetowa:

<http://helion.pl>



Zamówienia telefoniczne:

0 801 339900



0 601 339900

 WILEY



Helion

Sprawdź najnowsze promocje:

• <http://helion.pl/promocje>

Książki najchętniej czytane:

• <http://helion.pl/bestsellery>

Zamów informacje o nowościach

• <http://helion.pl/nowosci>

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 KORZYŚCI

cena: 39,90 zł

ISBN 978-83-246-7530-2



9 788324 675302