

Zwinny samuraj

Jak programują mistrzowie zwinności

Podręcznik zwinnego zespołu!

- Czym wyróżniają się zwinne projekty?
- Jak działa zwinne planowanie?
- Co dają testy jednostkowe i ciągła integracja?



Jonathan Rasmusson



Tytuł oryginału: The Agile Samurai: How Agile Masters Deliver Great Software

Tłumaczenie: Andrzej Stefański

ISBN: 978-83-246-3483-5

Copyright © 2010 Pragmatic Programmers, LLC
All rights reserved

Copyright © 2012 by Helion S.A.

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

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

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

Autor oraz Wydawnictwo HELION 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 Wydawnictwo HELION nie ponoszą 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/zwisam>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

Spis treści

Podziękowania	9
Dobrze Cię widzieć	11
Część I. Wprowadzenie	15
Rozdział 1. Zwinność w pigułce	17
1.1. Dostarczaj czegoś wartościowego co tydzień	18
1.2. Jak działa zwinne planowanie?	21
1.3. Zrobione oznacza zrobione	23
1.4. Trzy proste prawdy	24
Rozdział 2. Poznaj swój zwinny zespół	27
2.1. Czym wyróżniają się zwinne projekty?	28
2.2. Co napędza zwinny zespół	30
2.3. Typowe role	36
2.4. Wskazówki co do tworzenia Twojego zwinnego zespołu	45
Część II. Inicjacja projektu zwinnego	49
Rozdział 3. Jak zapakować autokar	51
3.1. Co zabija większość projektów	52
3.2. Zadawaj trudne pytania	52
3.3. Zrób tablicę koncepcyjną	54

3.4. Jak to działa	54
3.5. Tablica koncepcyjna w pigułce	55
Rozdział 4. Kontekst projektu	57
4.1. Zapytaj: po co tu jesteśmy?	58
4.2. Tworzenie krótkiego podsumowania	60
4.3. Projekt opakowania	63
4.4. Stwórz listę „NIE”	66
4.5. Poznaj swoich sąsiadów	68
Rozdział 5. Realizacja	75
5.1. Pokaż rozwiązanie	76
5.2. Zapytaj, co nie da nam spokojnie spać	77
5.3. Określ rozmiar	81
5.4. Wyjaśnij dokładnie, co zamierzasz dostarczyć	83
5.5. Pokaż, co się będzie działo	90
Część III. Planowanie zwinnego projektu	97
Rozdział 6. Zbieranie historii użytkowników	99
6.1. Problem z dokumentacją	100
6.2. Wprowadź historie użytkownika	103
6.3. Cechy dobrych historii użytkownika	104
6.4. Jak przeprowadzić warsztaty zbierania historii	112
Rozdział 7. Szacowanie: piękna sztuka zgadywania	119
7.1. Problem z wysokopoziomowymi szacunkami	120
7.2. Zamiana cytryn w lemoniadę	121
7.3. Jak to działa?	127
Rozdział 8. Zwinne planowanie: zmagania z rzeczywistością	135
8.1. Problemy z planowaniem statycznym	136
8.2. Stwórz zwinny plan	138
8.3. Bądź elastyczny co do zakresu projektu	140
8.4. Twój pierwszy plan	143

8.5. Wykres malejący	151
8.6. Zmiana projektu w projekt zwinny	155
8.7. Zastosowanie w praktyce	156
Część IV. Realizacja zwinnego projektu	165
Rozdział 9. Zarządzanie iteracjami: działanie	167
9.1. Jak dostarczać wartościowe rzeczy co tydzień	168
9.2. Zwinna iteracja	169
9.3. Potrzebna pomoc	170
9.4. Krok 1. Analiza i projektowanie: przygotowanie do pracy	171
9.5. Krok 2. Programowanie: praca	177
9.6. Krok 3. Testowanie: sprawdzanie pracy	178
9.7. Kanban	180
Rozdział 10. Tworzenie zwinnego planu komunikacji	185
10.1. Cztery rzeczy do zrobienia w każdej iteracji	186
10.2. SPH — spotkanie planowania historii	186
10.3. Pokaz	188
10.4. Zaplanuj następną iterację	188
10.5. Jak poprowadzić miniprzegląd	190
10.6. Jak nie prowadzić codziennych podsumowań	192
10.7. Wykorzystaj to, co działa	193
Rozdział 11. Przygotowanie wizualizacji przestrzeni roboczej	197
11.1. Oho... Mamy kłopoty!	198
11.2. Jak stworzyć wizualizację przestrzeni roboczej	201
11.3. Pokaż swoje zamiary	203
11.4. Stwórz i ogłoś wspólny słownik dla danej dziedziny	204
11.5. Uważaj na błędy	205
Część V. Tworzenie zwinnego oprogramowania	209
Rozdział 12. Testowanie jednostkowe: wiedzieć, że działa	211
12.1. Witamy w Vegas!	212
12.2. Wprowadź testy jednostkowe	214

Rozdział 13. Refaktoryzacja: spłacanie długu technicznego	221
13.1. Wprowadzanie dynamicznych zmian	222
13.2. Dług techniczny	223
13.3. Spłacanie przez refaktoryzację	225
Rozdział 14. Programowanie oparte na testach (TDD)	233
14.1. Najpierw napisz testy	234
14.2. Wykorzystanie testów do opanowania złożoności	238
Rozdział 15. Ciągła integracja: utrzymywanie gotowości produkcyjnej	243
15.1. Pokaz	244
15.2. Kultura gotowości produkcyjnej	246
15.3. Czym jest ciągła integracja?	247
15.4. Jak to działa?	248
15.5. Przygotuj proces publikacji kodu	249
15.6. Stwórz automatyczną kompilację	250
15.7. Pracuj nad małymi fragmentami	252
15.8. Co dalej?	254
Dodatki	257
Dodatek A. Zasady zwinności	259
A.1. Manifest Agile	259
A.2. Dwanaście zasad zwinności	260
Dodatek B. Zasoby internetowe	261
Dodatek C. Bibliografia	263

Rozdział 4.

Kontekst projektu



Oprogramowanie to jedna z tych unikalnych aktywności, w których łączy się projektowanie, konstruowanie, sztukę i naukę w jednym. Zespoły muszą podejmować tysiące decyzji i kompromisów każdego dnia. Bez dobrego kontekstu lub dokładnego zrozumienia niemożliwe jest wybieranie najlepszych rozwiązań w sposób świadomy i wyważony.

Na początku tworzenia tablicy koncepcyjnej będziemy próbowali bardzo jasno wyjaśnić „dlaczego” kryjące się za naszym projektem, odpowiadając na takie pytania, jak:

- ◆ Dlaczego tu jesteśmy?
- ◆ Jak można to krótko podsumować?
- ◆ Jak powinna wyglądać reklama naszego produktu?
- ◆ Czego nie zamierzamy robić?
- ◆ Kto nas otacza?

Po przeczytaniu tego rozdziału Ty i Twój zespół będziecie dokładnie rozumieli, jaki jest cel projektu, będziecie wiedzieli, dlaczego go tworzycie, i będziecie w stanie jasno i szybko przekazać to innym.

Ale najpierw zacznijmy od zapytania sponsorów: dlaczego tu jesteśmy.

4.1. Zapytaj: po co tu jesteśmy?

Po co tu jesteśmy?

- Aby bezpiecznie śledzić i monitorować pracę na budowie.



Zanim jakikolwiek zespół projektowy będzie mógł odnieść prawdziwy sukces, musi zrozumieć *dłaczego* kryjące się za tym, co tworzy. Gdy członkowie zespołu rozumieją *dłaczego*, mogą:

- ♦ podejmować lepsze, bardziej świadome decyzje;
- ♦ w lepszy sposób godzić przeciwieństwa i wypracowywać kompromisy;
- ♦ tworzyć lepsze, bardziej innowacyjne rozwiązania, ponieważ mogą myśleć samodzielnie.

Chodzi tutaj też o to, byś sam mógł znaleźć *myśl przewodnią* oraz wszystko *zobaczyć na własne oczy*.

Idź i przekonaj się sam

Czym innym jest logicznie zrozumieć, dlaczego się tutaj znaleźliśmy, a czym zupełnie innym samemu to zobaczyć. Aby naprawdę dokładnie wejść w skórę Twojego klienta i zrozumieć, czego on potrzebuje, musisz pójść i zobaczyć to na własne oczy.

Pójść i zobaczyć oznacza, że członkowie Twojego zespołu muszą ruszyć tyłek i zobaczyć miejsce, w którym odbywa się cała akcja.

Toyota: mistrzowie sprawdzania na własne oczy

W swojej wspaniałej książce *Droga Toyoty* [Lik04] Jeffrey Liker opisuje historię głównego inżyniera, któremu powierzono przeprojektowanie Toyoty Sienna 2004 pod kątem klientów z Ameryki Północnej. Aby poczuć, jak ludzie w Ameryce Północnej żyją, pracują i używają swoich samochodów, on i jego zespół przejechali Toyotą Sienna przez wszystkie stany i prowincje USA, Kanadę oraz Meksyk.

Oto, co odkryli:

- ◆ Kierowcy w Ameryce Północnej częściej jedzą i piją w swoich samochodach niż kierowcy w Japonii (gdzie przemierzane dystanse są z reguły krótsze). Z tego powodu w każdej Toyocie Sienna możesz zobaczyć na środku podstawkę i czternaście uchwytych na kubki w standardzie.
- ◆ Drogi w Kanadzie mają wyższą koronę (wypukłość na środku) niż w Ameryce, dlatego kontrolowanie „znoszenia” podczas jazdy jest tam bardzo ważne.
- ◆ Mocniejsze boczne wiatry w Ontario czynią odporność na boczne powiewy dużo ważniejszym zagadnieniem do rozwiązania. Jeśli jedziesz gdzieś przy silnym bocznym wietrze, nowa Sienna jest dużo bardziej stabilna i łatwiejsza do opanowania.

Choć główny inżynier mógłby to wyczytać w raporcie marketingowym, nie przejąłby się tym tak bardzo i nie zrozumiał tak dokładnie, jak teraz — po przejechaniu i zobaczeniu tych rzeczy na własne oczy.

Na przykład jeśli tworzysz system uprawnień dla firmy budowlanej pracującej przy wykopach, jedź na budowę. Porozmawiaj z osobami odpowiedzialnymi za bezpieczeństwo. Zobacz baraki. Popatrz na spartańskie warunki, kiepskie połączenie z internetem i ograniczone powierzchnie, na których muszą pracować Twoi klienci. Spędź dzień na budowie i popracuj z ludźmi, którzy będą używać Twojego systemu dzień i noc. Zaangażuj się, zadawaj pytania, stań się swoim klientem.

Odkryj swoją myśl przewodnią

Myśl przewodnia to krótkie wyrażenie, fraza lub zdanie podsumowujące cel lub przyczynę Twojego projektu lub misji. To takie zdanie, światło rozjaśniające drogę, na które można się powołać w każdym momencie, które w wirze walki pomoże zdecydować, czy należy atakować, czy może utrzymywać pozycję.

W książce *Made to stick* [HH07] Chip i Dan Heath opisują historię, w której pracownicy firmy Southwest Airlines zastanawiali się, czy dodać sałatkę z kurczakiem w czasie jednego z lotów.

Gdy padło pytanie, czy zmniejszy to koszty i cenę biletu (myśl przewodnia dyrektora generalnego Herbsa Kellehera), stało się jasne, że dodanie opcjonalnej sałatki z kurczakiem nie ma żadnego sensu.

Myślą przewodnią Twojego projektu nie musi być nic wielkiego lub inspirowającego. Może to być coś naprawdę prostego i związłego.

Kluczem do tego ćwiczenia jest skłonienie ludzi do rozmowy o tym, dlaczego *ich zdaniem* znajdują się oni w tym miejscu, a następnie potwierdzenie u Twojego klienta, czy rzeczywiście dokładnie o to chodzi.

4.2. Tworzenie krótkiego podsumowania



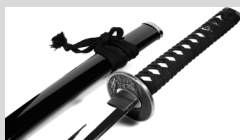
Krótkie podsumowanie

- dla [kierowników budowy]
- którzy [muszą śledzić, jakiego rodzaju czynności są wykonywane na placu budowy]
- produkt [SBPB*]
- który jest [systemem kontroli bezpieczeństwa pracy]
- i [tworzy, śledzi i sprawdza upoważnienia do bezpiecznego wykonywania pracy]
- inaczej niż [obecnie używany rejestr papierowy]
- nasz produkt [jest dostępny przez internet i może być wykorzystywany w dowolnym czasie oraz miejscu]

*SBPB: System Bezpieczeństwa Prac Budowlanych

Szybko! Decydenci z funduszy venture (VC), do których chcieliście się dostać przez ostatnie trzy miesiące, właśnie wsiadli do windy i masz trzydzieści sekund, żeby wyłożyć im pomysł na Twój nowy, wykluwający się startup. Sukces sprawi, że Twoje przedsięwzięcie złapie wiatr w żagle. Porażka oznacza dalsze życie na makaronie z serem.

Taka jest idea **krótkiego podsumowania** (ang. *elevator pitch*) — sposobu na przekazanie esencji Twojego pomysłu w bardzo krótkim czasie. Krótkie podsumowania nie są jednak tylko dla początkujących przedsiębiorców. Są one również wspaniałym sposobem na zwięźle opisanie nowych projektów programistycznych.

Są tysiące powodów, aby wykonać Twój projekt

Ostatnio przeprowadziłem to ćwiczenie z zespołem mającym zaprojektować system fakturowania dla nowego działu firmy i byłem zdumiony liczbą przyczyn, które członkowie zespołu podali jako powód tworzenia nowego systemu.

Niektórzy myśleli, że powodem jest redukcja liczby stron na fakturze w celu oszczędzania papieru. Inni myśleli, że miało to uprościć faktury i dzięki temu zredukować obciążenie biura obsługi klienta. Jeszcze inni myśleli, że jest to okazja do uruchomienia profilowanych kampanii marketingowych, aby spróbować sprzedać klientom dodatkowe produkty i usługi.

Wszystkie odpowiedzi były dobre i każda z nich osobno tłumaczyłaby prowadzenie projektu. Ale tylko dzięki długim dyskusjom, debatom i próbom zrozumienia pojawił się prawdziwy cel powstania projektu, którym było po prostu uproszczenie faktur i zredukowanie rozmiaru biura obsługi klienta.

Dobre krótkie podsumowanie powinno spełniać kilka funkcji w Twoim projekcie.

I. Wprowadzić jasność.

Zamiast próby dostarczenia wszystkiego dla każdego, krótkie podsumowanie zmusza zespół do odpowiedzenia na trudne pytanie — czym jest produkt i dla kogo.

II. Zmusić zespół do myślenia o kliencie.

Skupiając się na tym, co ma być zrobione w projekcie i dlaczego, zespoły zyskują wartościowy wgląd w to, co jest ważne w produkcie, a przede wszystkim, dlaczego klient go kupuje.

III. Trafić w sedno.

Jak laser, krótkie podsumowanie przecina masę pyłu i dociera do sedna tego, czym jest projekt. Ta jasność pomaga ustalić priorytety i znakomicie poprawia stosunek sygnału do szumu na temat tego, co naprawdę się liczy.

Przyjrzyjmy się teraz szablonowi, który pomoże przygotować Twoje podsumowanie.

Szablon krótkiego podsumowania

- ◆ **dla** [docelowy klient]
- ◆ **który** [opis potrzeby lub okazji]
- ◆ **produkt** [nazwa produktu]
- ◆ **jest** [kategoria produktu]
- ◆ **który** [kluczowa korzyść, ważny powód, by kupić]
- ◆ **w odróżnieniu od** [główna konkurencyjna alternatywa]
- ◆ **nasz produkt** [opis lub najważniejsza przewaga].

Nie ma jednego sposobu na wykonanie krótkiego podsumowania. Ja lubię ten, który opisał Geoffrey Moore w książce *Crossing the Chasm* [Moo91].

- ◆ *dla* [docelowy klient] — tłumaczy, dla kogo jest projekt lub kto skorzysta z jego używania;
- ◆ *który* [opis potrzeby lub okazji] — rozszerza problem lub potrzebę, którą chce zaspokoić klient;
- ◆ *produkt* [nazwa produktu] — Nadaje życia naszemu projektowi, przypisując mu nazwę. Nazwy są ważne, ponieważ przekazują zamiary;
- ◆ *jest* [kategoria produktu] — tłumaczy, czym w rzeczywistości ta usługa jest lub co oferuje produkt.

Bycie zwięzłym jest trudne

Jednym z powodów tak dużej mocy krótkiego podsumowania jest jego zwięzła forma. Nie myśl jednak, że napisanie czegoś krótkiego jest proste.

Może trzeba będzie przejść wiele prób, zanim Ty i Twój zespół znajdziecie dobre podsumowanie, dlatego nie martw się, jeśli nie trafisz za pierwszym razem. Napisanie dobrego podsumowania może być bardzo trudną pracą, ale wartą wysiłku.

Napisałbym do Ciebie krótszy list, ale nie miałem czasu. — stwierdził Blaise Pascal w *Prowincjałkach*.

- ♦ *który* [kluczowa korzyść, ważny powód, by kupić] — tłumaczy, dlaczego Twój klient zechce kupić przede wszystkim ten produkt;
- ♦ *w odróżnieniu od* [główna konkurencyjna alternatywa] — mówi, dlaczego nie używamy nadal tego, co już jest;
- ♦ *nasz produkt* [opis lub najważniejsza przewaga] — wyróżnia i tłumaczy, w jaki sposób nasz produkt się wyróżnia lub dlaczego jest lepszy niż konkurencyjne rozwiązania. To najważniejsza rzecz. Od tego zależy, czy otrzymamy pieniądze na nasz projekt.

Dwa zdania krótkiego podsumowania obejmują wszystko, czego potrzebujemy, by szybko przekazać esencję swojego projektu lub pomysłu. Mówią one nam, czym i dla kogo jest nasz produkt, a przede wszystkim, dlaczego ktokolwiek mógłby chcieć go kupić.

Możesz przygotowywać krótkie podsumowanie ze swoim zespołem na wiele różnych sposobów. Możesz wydrukować szablon i kazać wszystkim wypełnić go samodzielnie, zanim zbierzesz wszystkich razem.

Jeśli wolisz oszczędzić kilka drzew, po prostu wyświetl szablon na ekranie i wypełnij go razem z grupą, przechodząc przez wszystkie elementy szablonu po kolei.

Z krótkim podsumowaniem w dłoni sięgnijmy do Twojej kreatywności i zaprojektujmy opakowanie Waszego produktu.

4.3. Projekt opakowania



Oprogramowanie jest czasami złem koniecznym dla firm. Zamiast podejmować całe ryzyko i niepewność związaną z dużymi projektami, wiele z nich będzie wołało iść do sklepu, zapłacić i po prostu kupić to, co jest niezbędne.

Choć dokładnie dopasowane pakiety oprogramowania o wartości milionów dolarów prawdopodobnie jeszcze długo nie zagoszczą na półkach supermarketów, pojawia się interesujące pytanie. Gdybyśmy mogli znaleźć nasze oprogramowanie na półce w supermarkecie, jak powinno wyglądać jego opakowanie? I jeszcze ważniejsze: czy byśmy je kupili?

Tworzenie opakowania dla Twojego projektu i postawienie pytania, dlaczego ktoś miałby je kupić, każe Twojemu zespołowi skupić się na tym, co ważne dla Twojego klienta, oraz na tym, jakie są dodatkowe korzyści z Twojego produktu. Lepiej, żeby zespół był świadom obu tych rzeczy podczas tworzenia oprogramowania.

Jak to działa?

Wiem, co teraz myślisz. „Nie jestem kreatywny. Nie pracuję w reklamie. Prawdopodobnie nie jestem w stanie stworzyć reklamy swojego produktu”.

Mam dla Ciebie dobrą wiadomość. Oczywiście, że możesz. A ja zamierzam Ci pokazać w trzech prostych krokach, jak to zrobić.

Krok 1: urządź burzę mózgów na temat korzyści wynikających z Twojego produktu

Nigdy nie mów klientom o cechach Twojego produktu — ich to nie obchodzi. Ludzie są jednak zainteresowani tym, w jaki sposób Twój produkt ułatwi im życie. Innymi słowy: korzyściami wynikającymi z Twojego produktu.

Załóżmy przykładowo, że próbujemy przekonać rodzinę do korzyści wynikających z kupienia minivana. Moglibyśmy pokazać im listę wszystkich cech. Możemy też pokazać, w jaki sposób minivan uczyniłby ich życie lepszym.

Cechy



Korzyści

245 koni mechanicznych
System kontroli trakcji
ABS

Łatwiejsze wyprzedzanie na autostradzie
Oszczędność pieniędzy
Bezpieczniej zatrzyma się z Twoimi ukochanymi

Upewnij się, że wszystkie cechy przekułeś na zalety!

Widzisz różnicę?

Tak więc pierwszym krokiem przy tworzeniu opakowania Twojego produktu jest zebranie Twojego zespołu oraz klientów i przeprowadzenie burzy mózgów na temat wszystkich powodów, dla których ludzie mogliby chcieć używać tego produktu. Wybierz z nich trzy najważniejsze.

Krok 2: stwórz hasło reklamowe

Kluczem do dobrego sloganu jest to, by powiedzieć tak dużo, jak to tylko możliwe za pomocą bardzo niewielu słów. Nie muszę Ci mówić, na czym skupiają się poniższe firmy, ponieważ ich slogany mówią wszystko:

- ◆ Acura — Prawdziwa definicja luksusu. Dla Ciebie.
- ◆ FedEx — Spokojna głowa.
- ◆ Starbucks — Codzienne chwile przyjemności.

Czy odczuwasz emocje płynące z tych haseł?

Teraz rozluźnij się. To całkiem niezłe slogany, a Twój nie musi być aż tak profesjonalny. Po prostu spotkaj się ze swoim zespołem, ogranicz burzę mózgów na temat hasła reklamowego do dziesięciu lub piętnastu minut i ciesz się z tego, że możesz poćwiczyć kreatywną część swojego mózgu. Pamiętaj, żaden slogan nie jest przesłodzony.

Krok 3: Zaprojektuj opakowanie

Wspaniale! Jesteś prawie u celu. Ze swoimi trzema przekonującymi powodami do zakupu i silnie przyciągającym sloganem jesteś już gotowy, by je połączyć.

Tutaj nazwa produktu

Dobre zdjęcie

Najlepszy slogan tutaj

Korzyść nr 1 _____

Korzyść nr 2 _____

Korzyść nr 3 _____

W tym ćwiczeniu wyobraź sobie, że Twój klient wszedł do sklepu z oprogramowaniem i zobaczył opakowanie Twojego produktu leżące na półce, a gdy je wziął do ręki, wyglądało tak przekonująco, że od razu kupił dzieś sztuk dla siebie i swoich przyjaciół.

A teraz szybko, narysuj to opakowanie!

Nie musi to być druga *Mona Lisa*. Użyj zwykłego papieru do flipcharta, kolorowych markerów, papieru, nalepek i cokolwiek jeszcze wpadnie Ci w ręce. Wykrzycz swój slogan. Pokaż klientom zalety produktu. Poświęć piętnaście minut, aby zaprojektować opakowanie produktu najlepsze, jakie jesteś w stanie.

Wspaniale! Widzisz, nie było to takie trudne. Zabaw się trochę przy tym (nie każdego dnia możesz używać kredek i kreślić przekonujący obraz produktu). Jest to dobre ćwiczenie na budowanie zespołu i zabawny sposób, aby krytycznie pomyśleć na temat pytania *dlaczego* stojącego za Twoim oprogramowaniem.

Teraz zobaczmy, co można zrobić, aby rozpocząć ustalanie oczekiwań co do zakresu naszego projektu.

4.4. Stwórz listę „NIE”

W ZAKRESIE	POZA ZAKRESEM
Stworzenie nowego uprawnienia Uaktualnianie/odczytywanie /kasowanie istniejącego uprawnienia Wyszukiwanie Proste raportowanie Drukowanie	Połączenie ze starym systemem blokowania dróg Działanie offline
NIEOKREŚLONE	
Integracja ze śledzeniem logistycznym System obsługi kart bezpieczeństwa	

Podczas ustalania oczekiwań co do zakresu Twojego projektu powiedzenie, czego nie zamierzasz zrobić, może być tak samo ważne, jak powiedzenie tego, co zamierzasz zrobić.

Tworząc listę „NIE”, jasno określasz, co jest w zakresie, a co poza zakresem Twojego projektu. Zrobienie tego nie tylko jasno ustali oczekiwania Twojego klienta, ale także zapewni, że Ty i Twój zespół skupiacie się na rzeczach naprawdę ważnych, ignorując wszystko inne.

Jak to działa?

Lista „NIE” jest wspaniałą wizualizacją, przejrzystością pokazującą, co jest w zakresie, a co poza zakresem Twojego projektu. Po prostu siadasz ze swoim klientem i zespołem i podczas burzy mózgów na temat wszystkich ogólnych funkcjonalności, które klient chciałby zobaczyć w swoim oprogramowaniu, wypełniasz puste pola.

W	POZA
 Wielkie głązy do podniesienia	 Rzeczy, którymi nie zawracamy sobie głowy
NIEOKREŚLONE	
Rzeczy, które nadal musimy zakwalifikować	

W zawiera rzeczy, na których chcemy się skupić. Tutaj mówimy: „To są wielkie głązy, które będziemy przesuwac w tym projekcie”. Mogą to być najważniejsze funkcjonalności (jak raportowanie) lub ogólne cele (jak skalowalność w stylu Amazon).

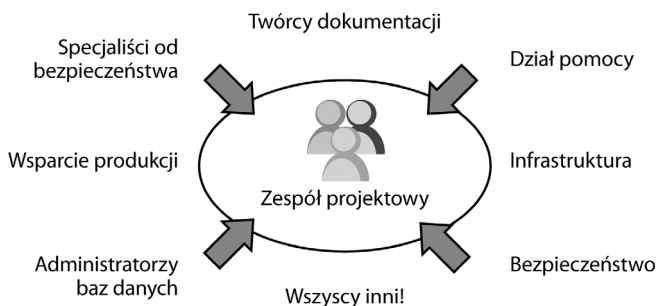
POZA zawiera rzeczy, którymi nie chcemy sobie zawracać głowy. Mogą to być rzeczy, które chcemy odłożyć do kolejnej wersji, lub po prostu takie, które są poza zakresem tego projektu. Na razie jednak nie zamierzamy się nimi przejmować. Spadają ze stołu.

NIEWYJAŚNIONE to lista rzeczy, co do których jeszcze nie podjęliśmy decyzji. Jest to bardzo wartościowa pozycja, ponieważ pokazuje prawdę o wielu projektach programistycznych. Mogą one znaczyć wiele rzeczy dla wielu ludzi — a tego właśnie chcemy uniknąć. W końcu będziemy chcieli przenieść wszystkie nasze **NIEWYJAŚNIONE** do części **W** lub **POZA**.

Piękno tej wizualnej prezentacji tkwi w tym, jak dużo widać na pierwszy rzut oka. Szybko przeglądając ważne pozycje znajdujące się w zakresie po lewej stronie, poza zakresem po prawej i w końcu niewyjaśnione na dole, każdy może wyrobić sobie jasny obraz tego, gdzie leżą granice naszego projektu.

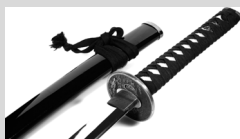
Mając wyraźnie zdefiniowany zakres, możemy iść do przodu i zobaczyć, kto znajduje się w otoczeniu projektu.

4.5. Poznaj swoich sąsiadów



Dobrzy sąsiedzi mogą być Twoimi najlepszymi przyjaciółmi. Są pod ręką, gdy zatrzaskną Ci się drzwi. Są pod ręką, gdy potrzebujesz wiertarki. No, a zrobi się całkiem miło, gdy pomożesz im uruchomić sieć bezprzewodową w domu.

Pytanie za milion dolarów



Przygotowywałem kiedyś wspólnie z dużą kanadyjską firmą tablicę koncepcyjną, gdy wiceprezes działu zapytał, jak ten nowy system będzie zintegrowany z istniejącym przestarzałym mainframe'em.

W pokoju mógłbyś usłyszeć spadającą kroplę. Wiceprezes, który podpisuje czeki i jest osobiście odpowiedzialny za sukces tego projektu, nie wiedział, że nowy system nigdy nie będzie zintegrowany ze starym. On ma go po prostu zastąpić.

I to właśnie dlatego stworzyliśmy listę „NIE” — aby uniknąć dużej zmiany oczekiwań na jakimś dalszym etapie projektu. Lepiej zrobić to teraz, niż próbować zmienić coś takiego, gdy projekt jest już tworzony.

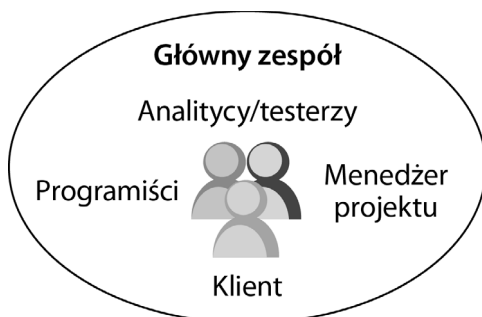
Możesz wierzyć lub nie, ale Twój projekt ma też sąsiadów. Tyle że zamiast przechowywać zapasowe klucze i pożyczać Ci wiertarkę, zarządzają oni bazami danych, robią audyty bezpieczeństwa i zapewniają działanie sieci.

Poznając swoich sąsiadów, możesz na początku zbudować relacje, które będą owocowały podczas trwania projektu. Poza tym kulturalnie jest powiedzieć od czasu do czasu „Cześć” i nie biec do nich dopiero, gdy Twój dom się pali. I, co najważniejsze, podstawą każdej odnoszącej sukcesy społeczności projektowej jest zaufanie.

Moja pierwsza duża wpadka projektowa

Wszyscy popełniamy błędy. Jeden z moich największych popełniłem, prowadząc zespół ThoughtWorks wykonujący pewną pracę w firmie Microsoft.

Poszedłem tam i zacząłem projekt, myśląc, że nasza społeczność projektowa wygląda tak:



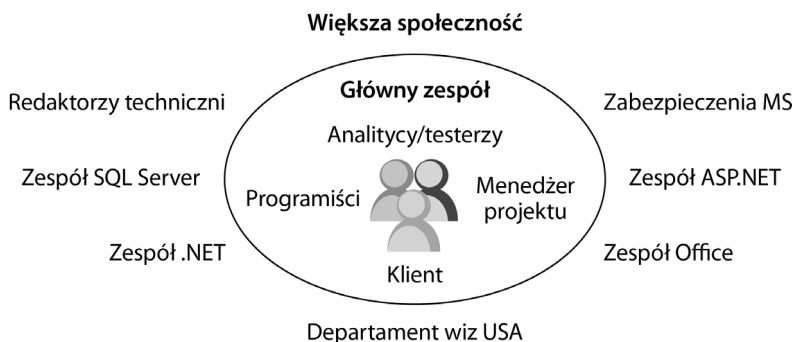
I przez pewien czas wszystko było w porządku. Zespół pracował w sposób zwinny. Regularnie dostarczaliśmy działające oprogramowanie i życie było piękne.

Nagle, pod koniec projektu, zaczęło się dziać coś dziwnego. Grupy i ludzie, których nigdy nie widziałem ani nie spotkałem, nagle zaczęli się pojawiać znikąd, stawiając dziwne wymagania mnie i mojemu zespołowi.

- ♦ Jedna z grup chciała sprawdzić naszą architekturę (jak gdyby nasza architektura wymagała sprawdzania!).
- ♦ Inna chciała się upewnić, że wypełniamy korporacyjne reguły bezpieczeństwa (ba!).
- ♦ Jeszcze inna chciała zrobić przegląd naszej dokumentacji (jakiej dokumentacji?).

Kim byli ci wszyscy ludzie? Skąd się wzięli? I dlaczego tak bardzo chcieli pokrzyżować nasze plany?

W ciągu jednej nocy nasza miła mała społeczność projektowa zmieniła się z małego zespołu sześciu ludzi w coś dużo większego i rozbudowanego.



Choć chciałem obwiniać wszystkich dookoła, że ingerują w nasze plany, tak naprawdę nie doceniłem stwierdzenia, że *społeczność związana z projektem jest zawsze większa, niż Ci się wydaje*¹.

W punkcie „Poznaj swoich sąsiadów” będziesz chciał zrobić mapę społeczności związanej z Twoim projektem, namierzyć wszystkich i rozpocząć budowanie relacji, zanim będziesz ich potrzebował. W ten sposób, gdy przyjdzie ten moment, nie będziecie sobie kompletnie obcy i będą oni w dużo lepszej pozycji, by Ci pomóc.

Jak to działa?

Zbierz cały zespół i zróbcie burzę mózgów, na której wynotujecie wszystkich ludzi, z którymi powinniście się skontaktować przed uruchomieniem projektu. Nieocenieni będą przy tym członkowie zespołu pracujący w firmie od dawna i świadomi wszystkich reguł korporacyjnych oraz pułapek organizacyjnych, które będzie trzeba ominąć.

Gdy już wiesz, na czym stoisz, porozmawiaj z każdą z grup i ustal, czy możesz przypisać do każdej nazwisko i dane kontaktowe. Twój menedżer projektu czy inna osoba, która będzie odpowiedzialna w projekcie za utrzymywanie takich zewnętrznych relacji, może następnie opracować plan zaangazowania tych grup.

¹ *The Blind Men and the Elephant* [Sch03].

Kawa, pączki i szczerość

Jeśli chodzi o budowanie poprawnych relacji z sąsiadami, trudno przecenić dobrą kawę z pączkiem...

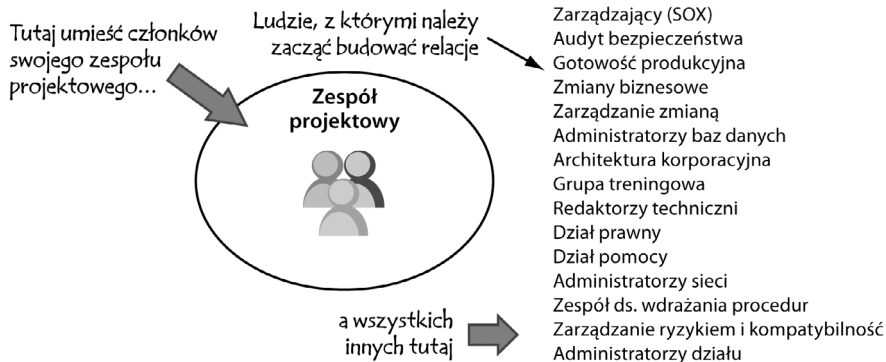
Kawa, ponieważ jest serwowana w miłym, ciepłym kubku i sąsiedzi, oprócz tego, że się z niej cieszą, będą Cię kojarzyć z uczuciem ciepła.

Pączki, ponieważ gdy będziesz mówił swoim sąsiadom, jak bardzo doceniasz to, że masz ich obok siebie, ich ciała będą cieszyły się słodkim smakiem cukru i dzięki temu sąsiedzi będą Cię kojarzyć ze słodyczą.

Ale najważniejszym narzędziem do budowania wspaniałych relacji z Twoimi sąsiadami jest szczerość.

Aby Twoi sąsiedzi poczuli się naprawdę docenieni i wartościowi, musisz ich tak postrzegać. Wyłapią oni nieszczerą pochwałę w ułamku sekundy. Ale prawdziwe uznanie i szczere podziękowania pozwolą szybko ich do siebie przekonać. A Ty i Twój projekt więcej na tym skorzystacie.

Większa społeczność



**Wielki Mistrz
i początkujący
wojownik**

UCZEŃ: *Mistrzu, sporo tych ćwiczeń wymaga czasu od sponsorów i udziałowców. A co jeśli są oni niedostępni lub zbyt zajęci, aby odpowiadać na tego typu pytania dotyczące projektu?*

MISTRZ: *Wtedy możesz sobie pogratulować. Za to, że właśnie odkryłeś największe ryzyko swojego projektu.*

UCZEŃ: *Co to za ryzyko?*

MISTRZ: *Zaangażowanie klienta. Bez zaangażowanego klienta Twój projekt ma problemy, zanim się jeszcze zacznie. Jeśli Twoi klienci nie mają czasu, żeby powiedzieć Ci, dlaczego pisesz dla nich to oprogramowanie, to może wcale nie należy go pisać.*

UCZEŃ: *Czy mówisz, że wtedy powinniśmy zatrzymać projekt?*

MISTRZ: *Mówię, że aby mieć udany projekt, potrzebujesz zaangażowania klientów i udziałowców. I że bez tego jesteś zgubiony, niezależnie, czy Ci się to podoba, czy nie.*

Uczeń: *Jeśli coś takiego nastąpi, co powinienem zrobić?*

MISTRZ: *Musisz jasno, ale dobitnie wytłumaczyć klientom, czego będzie wymagało przeprowadzenie tego projektu z sukcesem. Ich zaangażowanie i poświęcenie są niezbędne. Może to nie jest dobry czas na ten projekt? Prawdopodobnie oni są naprawdę zajęci i po prostu mają za dużo na głowie. Jeśli tak jest, powiedz im, że będziesz do ich dyspozycji, gdy będą gotowi, a tymczasem masz innych klientów do obsłużenia.*

UCZEŃ: *Dziękuję, Mistrzu. Muszę to przemyśleć.*

Co dalej?

Zanim pójdziemy dalej, zatrzymajmy się i odetchnijmy.

Czujesz to?

Widzisz, co się tutaj dzieje?

Z każdym kolejnym ćwiczeniem tablicy koncepcyjnej istota i zakres projektu stają się coraz bardziej zrozumiałe.

- ◆ Znamy *dla*czego stojące za naszym projektem.
- ◆ Mamy dobre krótkie podsumowanie.
- ◆ Wiemy, jak powinno wyglądać opakowanie naszego produktu.
- ◆ Ustawiamy wyraźne ograniczniki zakresu projektu.
- ◆ Mamy dość dobre rozeznanie wśród naszych sąsiadów.

Wiem, co teraz myślisz. Znamy już wystarczająco kontekst! Kiedy przejdziemy do rzeczy i zaczniemy rozmawiać o tym, jak to zbudować? Właśnie teraz.

W rozdziale 5. zaczniemy wizualizować Twój projekt od strony technicznej i dowiemy się, co będzie potrzebne do jego zrealizowania.

Zatem odwróć stronę i przygotuj się do działania.

Skorowidz

A

Acura, slogan, 65
adaptacyjne planowanie, 22
adaptive planning, *Patrz* adaptacyjne planowanie
agile planning, *Patrz* zwinne planowanie
analityk, 39, 91
analiza, 171
 na-czas, 172
architektura, wybór, 77
automatyczna kompilacja, 250, 251, 252
 oprogramowanie, 251

B

Beck, Kent, 102, 218, 241
Bloomberg, Michael, 79
błędy, 205
budżet, 92, 93

C

Carnegie, Dale, 190
ciągła integracja, 247, 248, 252
coach, 45

coaching, 45
collective code ownership,
 Patrz współwłasność kodu
cone of uncertainty, *Patrz* stożek niepewności
CruiseControl, 251

Ć

ćwiczenie Druckera, 42

D

deliver by date, *Patrz* dostarczenie oprogramowania w określonym terminie
deliver by feature set, *Patrz* dostarczenie określonego zestawu funkcjonalności
diagram przepływu, 173
dług techniczny, 223, 224
Dodds, Keith, 52
dokumentacja, 100, 101
 porównanie z historiami użytkownika, 107
 problemy, 100, 101
dostarczenie określonego zestawu funkcjonalności, 149, 150, 151

dostarczenie oprogramowania
w określonym terminie, 149, 150
Droga Toyoty, książka, 59
Druckera, ćwiczenie, 42

E

elastyczność co do zakresu, 21
elevatorski pitch, *Patrz* krótkie
podsumowanie
Evans, Eric, 205

F

Feathers, Martin, 231
Feathers, Michael, 216
FedEx, slogan, 65
flexible on scope, *Patrz* elastyczność
co do zakresu
Fowler, Martin, 102, 231, 253

G

Galton, Francis, 130
Gibbons, Robin, 54
Git, 249
główna lista historii, 21, 26, 138, 143
gotowość produkcyjna, 246

H

hasło reklamowe, 65
Heath, Chip i Dan, 60
historie użytkownika, 21, 103, 104
cechy, 104, 105
negocjowalne, 106
pomoc w pisaniu, 108
porównanie z dokumentacją, 107
słowa kluczowe, 103
szablon, 111, 112
testowalne, 106

I

idealne dni, 127
inception deck, *Patrz* tablica
konceptyjna
informacje zwrotne, 19, 190
integracja ciągła, 247, 248, 252
INVEST, 107
iteracja, 21, 26, 138, 169
"0", 178
planowanie, 189
przykłady, 170
rzecz do zrobienia, 186

J

Jobs, Steve, 31

K

Kanban, 180, 181, 182
Kelleher, Herbs, 60
klient, 26, 36, 91
kontakt, 19
nowe wymagania, 157
w zespole, 31, 37
zaangażowany, 30, 31, 37
zwinny, 36, 37
kod
publikacja, 249
współwłasność, 179
kompilacja automatyczna, 250, 251,
252
oprogramowanie, 251
krótkie podsumowanie, 55, 62, 63
funkcje, 61
idea, 60
szablon, 62
tworzenie, 60

L

Lean, 25
Liker, Jeffrey, 59

M

Made to stick, książka, 60
Manifest Agile, 259
master story list, *Patrz* główna lista historii
McConnel, Steve, 120
menedżer projektu, 43, 91
metody zwinne, 21, 22, 23
 role w projektach, 36
minimal marketable feature set, *Patrz*
 minimalny sprzedawalny zestaw funkcjonalności
minimalny sprzedawalny zestaw funkcjonalności, 144
miniprzegląd, prowadzenie, 190
mistrz scruma, 45
MMF, *Patrz* minimalny sprzedawalny zestaw funkcjonalności
Moore, Geoffrey, 62
Mott, Randy, 82, 83
myśl przewodnia, 59, 60

N

negocjowalne historie użytkownika, 106

O

odbudowanie wiarygodności u klienta, 32
on-site customer, *Patrz* klient w zespole
opakowanie, projekt, 63, 64, 65, 66
opis produktu, *Patrz* główna lista historii
oprogramowanie
 dostarczanie, 18, 19, 20

P

personas, *Patrz* role
plan, tworzenie, 138, 139, 143
planistyczny poker, 131
planowanie, 137, 138
 adaptacyjne, 22
 priorytety, 147
 statyczne, 136
 zwinne, 21
PM, *Patrz* menedżer projektu
podsumowania
 codzienne, 192
 prowadzenie, 190, 191, 192
pokaz, 188
 szybkie przygotowanie, 244
poker planistyczny, 131
powieści, 115
product owner, *Patrz* właściciel produktu
programista, 40, 91
programowanie
 oparte na testach, 234, 237, 238
 praca, 177
 w parach, 176
programowanie ekstremalne, 31, 215, 246
 nazewnictwo, 26
projekt
 budżet, 92, 93
 czas trwania, 81, 82, 83
 elastyczność, 140, 142
 klient, 36, 37
 koniec czasu, 161
 myśl przewodnia, 59, 60
 opakowanie, 63, 64, 65, 66
 prezentacja planu, 90
 problemy, 198
 rozmiar, 81, 83

projekt

- ryzyka, 77, 78, 79, 80
- śmierć, 52
- trzy proste prawdy, 24, 25
- ustalanie terminów, 149
- wpadki, 69
- zakres, 66, 67, 144
- zespół programistyczny, 38
- zmiana w projekt zwinny, 155
- zwinny, 27, 28, 29
- źródło śmieci, 146
- projektant interakcji z użytkownikiem, 44, 91
- publikacja kodu, 249

R

- refaktoryzacja, 221, 224, 225, 226, 227, 229, 230, 231
- release, *Patrz* wydanie
- repozytoria, 249
- role, 174
- rozmiar projektu, 81, 83
- ryzyka projektowe, 77
- rozmowa, 78, 79, 80

S

- samoorganizacja zespołu, 32, 33
- Scrum, 25, 31
 - mistrz scruma, 45
 - nazewnictwo, 26
- scrum master, *Patrz* Scrum, mistrz scruma
- slogany, 65
 - Acura, 65
 - FedEx, 65
 - Starbucks, 65
- słownik, opracowanie, 204

SPH, *Patrz* spotkanie planowania historii

SPI, *Patrz* spotkanie planowania iteracji

spike, *Patrz* szpilka

sposób spartańskiego wojownika, 159

spotkanie planowania historii, 186, 187

spotkanie planowania iteracji, 189

sprint, *Patrz* iteracja

Starbucks, slogan, 65

statyczne planowanie, 136

stożek niepewności, 120, 121

Subversion, 249

Surowiecki, James, 130

system punktowy, 124, 125, 126

szacowanie, 120, 121

- bezwzględne, 124

- dokładne, 121

- planistyczny poker, 131

- problemy, 120

- system punktowy, 124, 125, 126

- triangulacja, 127

- względne, 123, 124

szpilka, 130

szybkość zespołu, 21, 138, 148

- szacowanie, 147, 149

T

tablica historii, 199, 200, 201, 202

tablica koncepcyjna, 51, 54, 55, 57, 199, 201

- idea, 54

- podsumowanie, 93

- w pigułce, 55

tablica wskaźników, 87, 88, 89

tablica wydania, 199

TDD, *Patrz* programowanie oparte na testach

team velocity, *Patrz* szybkość zespołu
test-driven development, *Patrz*
programowanie oparte na testach
tester, 41, 42, 91
testowalne historie użytkownika, 106
testowanie, 178, 179
testy akceptacyjne użytkownika, 179
testy jednostkowe, 214, 215, 216, 217, 218
The Pixar Touch, film, 31
ThoughtWorks, 52, 54, 69, 87
triangulacja, 127
trudne pytania, 52

U

UAT, *Patrz* testy akceptacyjne
użytkownika
user stories, *Patrz* historie użytkownika
UX, *Patrz* projektant interakcji
z użytkownikiem

V

visual workspace, *Patrz* wizualizacja
przestrzeni roboczej

W

Wake, Bill, 107
warsztaty zbierania historii, 112
wizualizacja przestrzeni roboczej, 197, 201
tworzenie, 201
wizualizacja rozwiązania, 76
właściciel produktu, *Patrz* klient
właściciel projektu, 37
wpadki projektowe, 69
współ własność kodu, 179

Wściekła Czwórka, 85, 86
budżet, 86
czas, 86
jakość, 86
zakres, 87
wydanie, 144
wykres malejący, 151, 152, 153, 200
wykres rosnący, 153, 154
wymagania, 102

X

XP, 25

Z

zasady zwinności, 20, 24, 31, 33, 34, 102, 141, 144, 190, 226
wszystkie, 260
zespół, 90
cross-functional, *Patrz* zespół,
wszechstronność
miejsce pracy, 30
odbudowanie wiarygodności
u klienta, 32
odpowiedzialność i samodzielność,
33, 34
rozproszony, 30
samoorganizacja, 32, 33
spotkania, 192, 193
szybkość, 21, 138, 147, 148, 149
szybkość mniejsza od planowanej,
158
typowe role, 36
utrata członka, 160
wspólny język z klientem, 204
wszechstronność, 34, 35, 46
zaangażowani klienci, 30, 31
zwinny, 27, 28, 29, 38

zespół programistyczny, 36, 38
 analityk, 39, 91
 klient, 91
 menedżer projektu, 43, 91
 programista, 40, 91
 projektant interakcji z
 użytkownikiem, 44, 91
 tester, 41, 42, 91

zwinna analiza, 171, 172
zwinne metody, 21, 22, 23
zwinne planowanie, 21
zwinności, zasady, 20, 24, 31, 33, 34,
 102, 141, 144, 190, 226
 wszystkie, 260
zwinny projekt, 27, 28, 29
 zwinny zespół, 27, 28, 29, 38

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



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

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

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

GRUPA WYDAWNICZA

 **Helion SA**

Poznaj nowoczesne metodologie wytwarzania oprogramowania!

Konkurencja na rynku oprogramowania, oczekiwania klientów oraz tempo życia wymagają zmiany podejścia do wytwarzania aplikacji. Klienci nie chcą czekać miesiącami na pierwszą wersję zamówionego produktu — chcą zobaczyć cokolwiek już za tydzień! Niemożliwe? A jednak! Jeśli zastosujesz zwinne praktyki, masz szansę błyskawicznie pokazać klientowi działające zrzęby aplikacji, a w kolejnych (krótkich!) iteracjach kolejne efekty. Zobacz, jaki wpływ na efektywność może mieć zwinność. Sprawdź, jak dobrać ludzi do zwinnego zespołu oraz jak nim zarządzać.

Jonathan w swojej książce zaprezentuje Ci wszystko, co musisz wiedzieć na temat zwinnych praktyk. Dowiesz się, jak ważny jest zespół, co go napędza i jakie role pełnią jego członkowie. Najpierw poznasz największe zagrożenia dla projektu i zalety tablic koncepcyjnych oraz przygotujesz się do rozpoczęcia fazy realizacji. Kolejne rozdziały zawierają niezbędne informacje poświęcone planowaniu, szacowaniu oraz zarządzaniu iteracjami. Dodatkowo na własne oczy zobaczysz, jak ważne są testy jednostkowe, refaktoryzacja oraz ciągła integracja w procesie wytwarzania oprogramowania. Książka ta jest idealnym źródłem informacji dla członków zwinnych zespołów oraz osób, które kolejny projekt chciałyby zrealizować z wykorzystaniem właśnie tej metodologii.

- Cechy charakterystyczne zwinnych projektów
- Rola w projekcie
- Wskazówki w kwestii tworzenia zwinnego zespołu
- Tablice koncepcyjne
- Identyfikacja istotnego ryzyka
- Definiowanie celu
- Zbieranie historii użytkowników
- Szacowanie
- Planowanie
- Zarządzanie iteracjami
- Plan komunikacji
- Codzienny miniprzegląd
- Testy jednostkowe
- Refaktoryzacja kodu
- Programowanie oparte na testach
- Potencjał ciągłej integracji

Jonathan Rasmusson

ma ponad dziesięć lat doświadczenia w branży. Jest przedsiębiorcą, międzynarodowym konsultantem i zwinnym trenerem związanym z firmami ThoughtWorks oraz Cortex Business Solutions. Od dawna pomaga innym znaleźć najlepsze rozwiązania i sposoby pracy. Brał także udział w tworzeniu zaawansowanych systemów dla dużych firm, takich jak Microsoft czy British Petroleum.



helion.pl
księgarnia internetowa

Nr katalogowy: 7501

Księgarnia internetowa
<http://helion.pl>

Zamówienia telefoniczne:
0 801 339900
0 601 339900



Helion

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

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

Cena: 49,00 zł

ISBN 978-83-246-3483-5



sięgnij po WIĘCEJ



KOD KORZYŚCI

Informatyka w najlepszym wydaniu