

Rusz głową!

Agile



Przyswoisz pojęcie
metodyk zwinnych

Zapanujesz
nad swoim
kodem dzięki XP



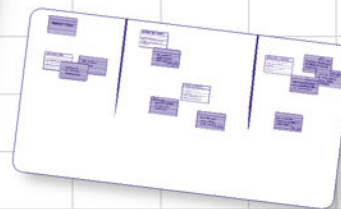
Poznasz tajemnice
Lean i Kanbana



Zrozumiesz podejścia
zwinne i przygotujesz się
do egzaminu PMI-ACP®



Dowiesz się, jak Scrum
pomógł Amy zachwyć
użytkowników



Zobaczysz, jak
zorganizować
pracę przy użyciu
tablic zadań

Tytuł oryginału: Head First Agile

Tłumaczenie: Maksymilian Gutowski

ISBN: 978-83-283-4364-1

© 2018 Helion S.A.

Authorized Polish translation of the English edition of Head First Agile,
9781449314330 © 2017 Andrew Stellman and Jennifer Greene.

This translation is published and sold by permission of O'Reilly Media, Inc.,
which owns or controls all rights to publish and sell the same.

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 Helion SA 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 Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne
szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion SA

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/agiler>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

Spis treści (streszczenie)

Wstęp	xxvii
1 Czym jest metodyka zwinna? <i>Założenia i praktyka</i>	1
2 Wartości i zasady metodyk zwinnych. <i>Nastawienie a metoda</i>	23
3 Zarządzanie projektami w Scrumie. <i>Zasady Scruma</i>	71
4 Planowanie i estymacja w podejściu zwinnym. <i>Ogólnie przyjęte praktyki scrumowe</i>	117
5 XP (programowanie ekstremalne). <i>Otwartość na zmiany</i>	177
6 Lean i Kanban. <i>Walka z marnotrawstwem i zarządzanie obiegiem pracy</i>	245
7 Przygotowanie do egzaminu PMI-ACP®. <i>Sprawdź swoją wiedzę</i>	307
8 Odpowiedzialność profesjonalisty. <i>Podjęcie słusznych decyzji</i>	377
9 Praktyka czyni mistrza. <i>Próbny egzamin PMI-ACP®</i>	391

Spis treści (część właściwa)



Wstęp

Zwinność zaczyna się od głowy.

Kiedy próbujesz się czegoś *nauczyć*, Twój *mózg* stara się wyświadczyć Ci przysługę, robiąc wszystko, co może, żebyś niczego *nie zapamiętał*. Twój mózg myśli sobie: „Lepiej zostawić w głowie miejsce na ważniejsze rzeczy, na przykład wiedzę o tym, przed jakimi dzikimi zwierzętami uciekać, albo o tym, że nie powinno się jeździć na desce snowboardowej w samych slipach”. Jak zatem *przechytryć* mózg, żeby zaczął myśleć, że Twoje życie rzeczywiście zależy od umiejętności zastosowania podejść zwinnych — a może nawet od zaliczenia egzaminu certyfikacyjnego PMI-ACP?

Dla kogo jest ta książka?	xx
Wiemy, co myślisz	xxi
Wiemy też, co czuje Twój mózg	xxi
Metapoznanie: myślenie o myśleniu	xxiii
Oto co MY zrobiliśmy	xxiv
Przeczytaj to	xxvi
Korektorzy merytoryczni	xxvii
Podziękowania	xxviii

Czym jest metodyka zwinna?

Założenia i praktyka

1

Zwinność jest dziś modna jak nigdy dotąd!

Naszej branży udało się po raz pierwszy odkryć praktyczny, zrównoważony sposób rozwiązywania problemów, z którymi musiały się męczyć całe pokolenia zespołów programistycznych. Zespoły zwinne posługują się jasnymi, prostymi **praktykami**, które sprawdziły się już w codziennej pracy nad rzeczywistymi projektami. Ale zaraz... Jeśli metodyka zwinna (ang. *agile*) jest tak wspaniała, to dlaczego nie korzystają z niej wszyscy? Wygląda na to, że w prawdziwym świecie te praktyki, które sprawdzają się w jednych zespołach, innym sprawiają trudności, a o takich rozbieżnościach decydują różnice w **nastawieniu**. Przygotuj się więc na zupełnie inny sposób myślenia o swoich projektach!



Nowe możliwości brzmią nęcąco...	2
...ale nie wszystko idzie zgodnie z planem	3
Metodyka zwinna na ratunek!	4
Kate próbuje przeprowadzić codzienny stand-up	5
Różni członkowie zespołu mają różne podejścia	6
Praktyki stosuje się sprawniej dzięki odpowiedniemu nastawieniu	8
Czym właściwie jest ta cała metodyka zwinna?	10
Scrum jest najpopularniejszym podejściem do wytwarzania zwinnego	12
Certyfikacja PMI-ACP pomoże Ci wyćwiczyć zwinność	18

W trakcie codziennego stand-upu wszyscy członkowie zespołu stoją. Dzięki temu spotkanie jest krótkie, rzeczowe i przebiega w swobodnej atmosferze.



Ale czy ten koleś rzeczywiście słucha, co jego współpracownicy mają do powiedzenia?

Wartości i zasady metodyk zwinnych

2

Nastawienie a metoda

Nie ma czegoś takiego jak „idealny” proces tworzenia udanego oprogramowania.

Niektóre zespoły odniosły niemałe sukcesy i odnotowały znaczącą poprawę po wdrożeniu zwinnych praktyk, metod i metodyk, podczas gdy inne miały z tym poważne problemy. Powiedzieliśmy już, że decydujące jest nastawienie członków zespołu. Co możesz zatem zrobić, żeby Twój własny zespół uzyskał pożądane rezultaty? Jak sprawić, aby zespół przyjął odpowiednie nastawienie? Właśnie w tym miejscu warto sięgnąć po **Manifest programowania zwinnego**. Omówienie z zespołem **wartości i zasad** pozwala zacząć nieco inaczej myśleć o praktykach zwinnych i ich zastosowaniu, co z kolei umożliwia *skuteczniejsze* korzystanie z nich.

Przełom w Snowbird	24
Manifest programowania zwinnego	25
Wprowadzanie nowych praktyk bywa niemałym wyzwaniem	26
Klinika Naukowa: pytania typu „które jest najlepsze?”	36
Szykował się hit...	38
...a wyszła katastrofa!	39
Zasady przyświecające Manifestowi programowania zwinnego	40
Zasady zwinnego programowania, które pomogą Ci w dostarczeniu Twojego produktu	42
Zasady metodyk zwinnych ułatwiają zespołom komunikację i współpracę	52
Nowy produkt to hit!	56



Zarządzanie projektami w Scrumie

Zasady Scruma

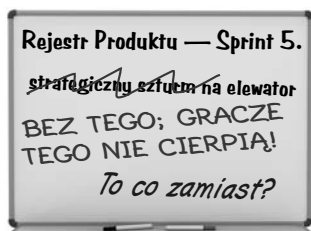
3

Zasady Scruma są proste. Efektywne ich stosowanie już niekoniecznie.

Scrum jest najbardziej rozpowszechnionym podejściem do programowania zwinnego, zresztą nie bez powodu — **zasady Scruma** są proste i łatwe do przyswojenia. Większość zespołów potrafi się szybko zaznajomić ze **zdarzeniami, z rolami i artefaktami** składającymi się na reguły Scruma. Aby jednak Scrum był prawdziwie skuteczny, pracownicy muszą naprawdę **pojąć jego wartości**, a także zasady *Manifestu programowania zwinnego*, które pomogą im przyjąć efektywne nastawienie. Choć sam Scrum wydaje się prosty, ciągłe prowadzenie **inspekcji i adaptacji** wymusza na zespołach przestawienie się na nowy sposób myślenia o projektach.



Poznaj zespół Ranch Hand Games	73
Zdarzenia Scruma, które pomagają w realizacji projektów	74
Role scrumowe pomagają zrozumieć, kto za co odpowiada	75
Artefakty scrumowe pomagają zespołowi być na czasie	76
Wartości Scruma zwiększają efektywność zespołu	82
Klinika Naukowa: pytania typu „co dalej?”	90
Zadanie nie jest wykonane, dopóki nie ma statusu „Ukończone”	92
Zespoły scrumowe dostosowują się do zmian w trakcie Sprintu	93
Manifest programowania zwinnego pomaga zrozumieć metodykę Scrum	96
Gorzej już było	102



↑
Wspólnie z nowym Właścicielem Produktu zespół powinien być w stanie wskazać najbardziej wartościowe funkcjonalności do opracowania w kolejnym Sprincie.



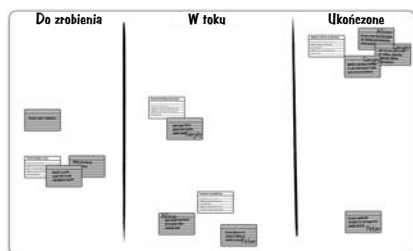
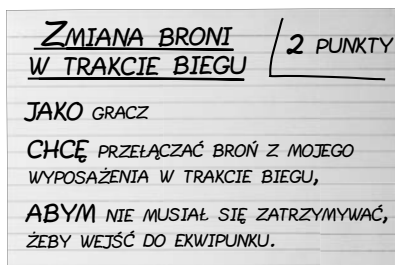
4

Planowanie i estymacja w podejściu zwinnym

Ogólnie przyjęte praktyki scrumowe

Zespoły zwinne używają prostych narzędzi planowania do sprawniejszego zarządzania swoimi projektami.

Zespoły scrumowe planują swoje projekty wspólnie, tak aby każdy członek zespołu zaangażował się w realizację celu każdego sprintu. W trosce o **zbiorowe zaangażowanie** zespołu trzeba zadbać o to, aby wszelkie planowanie, oszacowywanie i monitorowanie było na tyle proste, żeby można je było prowadzić grupowo. Zespoły scrumowe korzystające z najrozmaitszych narzędzi, od historyjek użytkownika, przez planistycznego pokera, po szybkość i wykresy spalania, zawsze wiedzą, co już zrobili i co jeszcze im zostało do zrobienia. W tym rozdziale zapoznasz się z narzędziami, które pozwalają zespołom na przeglądanie aktualnych informacji i utrzymywanie kontroli nad tym, co tworzą!



Co dalej?	121
Ogólnie przyjęte praktyki scrumowe	122
A można bez 300-stronicowej specyfikacji?	124
Historyjki użytkownika pomagają zespołom zrozumieć potrzeby użytkowników	125
Punkty historyjek pozwalają zespołowi skoncentrować się na względnej wielkości każdej historyjki	126
Cały zespół oszacowuje wspólnie	132
Koniec ze szczegółowymi planami projektów	134
Tablice zadań przedstawiają aktualne informacje	136
Klinika Naukowa: mylny trop	140
Wykresy spalania pomagają się zorientować, ile pracy zostało jeszcze do zrobienia	143
Szybkość określa, ile zespół może zrobić w ramach sprintu	144
Wzrastające wykresy spalania oddzielają postępy od zakresu prac	147
Skąd wiadomo, co stworzyć?	148
Mapy historyjek ułatwiają priorytetyzację elementów rejestru	149
Persony pomagają zaznajomić się z użytkownikami	150
Wieści mogłyby być lepsze...	152
Retrospektywy pomagają zespołowi usprawnić pracę	154
Narzędzia do wykorzystania w retrospektywach	156
Gratulacje! Jesteś już gotów!	174

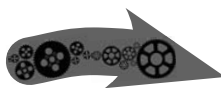
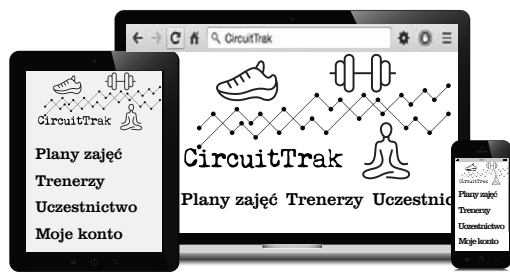
5

XP (programowanie ekstremalne)

Otwartość na zmiany

Zespoły zwinne odnoszą największe sukcesy, kiedy tworzą świetny kod.

Nawet najlepsze zespoły, składające się z samych utalentowanych programistów, muszą sobie radzić z problemami w swoim kodzie. Kiedy drobne zmiany w kodzie zaczynają „pączkować” i przekształcają się w ciąg nakładających się na siebie kombinacji bądź kiedy codzienne commity kodu prowadzą do trwającego kilka godzin rozwiązywania konfliktów złączenia gałęzi, praca, która *miała być przyjemna*, staje się **irytująca, nużąca i męcząca**. W tym miejscu do gry wchodzi XP. Jest to metodyka zwinna skoncentrowana na budowaniu spójnych zespołów, które sprawnie się komunikują i tworzą **swobodne, a zarazem pobudzające środowisko**. Gdy zespoły opracowują **prosty** kod zamiast złożonego, mogą **się otworzyć na zmiany**, zamiast się ich obawiać.



Poznaj zespół twórców CircuitTrak	178
Praca nocą i w weekendy jest źródłem problemów	180
XP działa na korzyść i zespołu, i kodu	181
Programowanie iteracyjne pomaga być na bieżąco ze zmianami	182
Odwaga i szacunek odpędzają strach	184
Zespoły tworzą lepszy kod, kiedy wszyscy ze sobą współpracują	190
Najlepsze zespoły siedzą razem	192
Zespoły XP cenią sobie komunikację	194
Zespoły pracują najlepiej, gdy wszyscy są zrelaksowani	196
Klinika Naukowa: pytania typu „które NIE jest?”	200
Zespoły XP są otwarte na zmiany	204
Częste zgłaszanie uwag pozwala zminimalizować zmiany	205
Złe doświadczenia powodują racjonalną obawę przed zmianą	206
Praktyki XP pozwalają gromadzić informacje zwrotne o kodzie	208
Zespoły XP stosują szybką automatyzację kompilacji	210
Ciągła integracja pozwala uniknąć przykrych niespodzianek	211
Cykl tygodniowy zaczyna się od pisania testów	212
Zespoły zwinne gromadzą informacje z projektów i testów	214
Programowanie w parach	216
Konserwacja złożonego kodu jest naprawdę trudna	223
Gdy prostota jest w cenie, kod może na tym jedynie zyskać	224
Prostota jest fundamentalną zasadą metodyk zwinnych	225
Każdy zespół zaciąga dług techniczny	226
Zespoły XP „spłacają” dług techniczny w każdym cyklu tygodniowym	227
Projektowanie przyrostowe wymaga prostego kodu	228

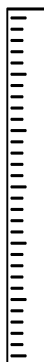
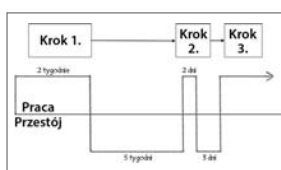
Lean i Kanban

6

Walka z marnotrawstwem i zarządzanie obiegiem pracy

Zespoły zwinne wiedzą, że niemal zawsze mogą jakoś usprawnić swój tryb pracy.

Pracownicy stosujący zasady **szczipłej produkcji** (ang. *lean production*) potrafią bez trudu odkryć, że robią coś, co nie pomaga im w **dostarczeniu wartości**. W takich sytuacjach eliminują **marnotrawstwo**, które ich spowalnia. Wiele zespołów ze szczipłym nastawieniem stosuje metodykę **Kanban** w ramach ustalania **limitów prac w toku** i tworzenia **systemów ciągnionych**, aby dopilnować, by pracownicy nie zajmowali się zadaniami, które niewiele wnoszą w rozwój produktu. Już wkrótce dowiesz się, jak traktowanie procesu rozwoju oprogramowania jako **spójnego systemu** pomoże Ci w tworzeniu lepszego oprogramowania!



Kłopoty z Analizatorem Odbiorców 2.5	246
Szczipła produkcja jest nastawieniem (a nie metodyką)	248
Zasady podejścia Lean pomagają spojrzeć na wszystko inaczej	249
Więcej zasad Lean	250
Narzędzia myślowe, których jeszcze nie widziałeś	254
Więcej narzędzi myślowych Lean	256
Rozmowa przy biurku	258
Klasyfikacja rodzajów marnotrawstwa ułatwia jego dostrzeżenie	260
Mapy strumienia wartości pozwalają dostrzec marnotrawstwo	264
Łapanie zbyt wielu srok za ogon	267
Anatomia opcji	270
Myślenie systemowe pomaga zespołom szczipłym spojrzeć na całość	272
Niektóre „ulepszenia” się nie sprawdziły	273
Zespoły Lean za pomocą systemów ciągnionych pilnują, by zawsze wykonywać najbardziej wartościowe zadania	274
Klinika Naukowa: najmniejsze zło	278
Kanban usprawnia procesy poprzez wykorzystanie systemu ciągnionego	280
Używaj tablic kanbanowych do wizualizacji obiegu pracy	281
Jak wykorzystać metodę Kanban do usprawnienia procesu roboczego	282
Zespół określa przepływ pracy	284
Rozmowa przy biurku	287
Zespół dostarcza produkty częściej	291
Kumulacyjny diagram przepływu pomaga w zarządzaniu	292
Zespoły kanbanowe dyskutują o swoich politykach	293
Sprzężenie zwrotne pokazuje skuteczność	294
Teraz cały zespół wspólnie wymyśla sposoby na sprawniejszą pracę!	295

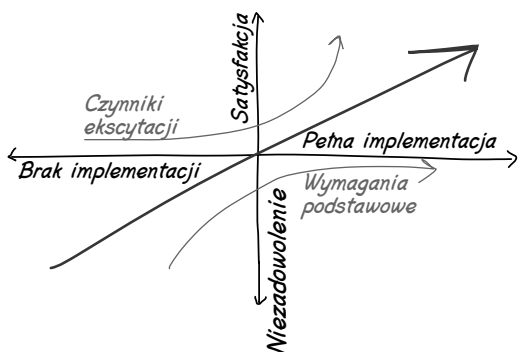
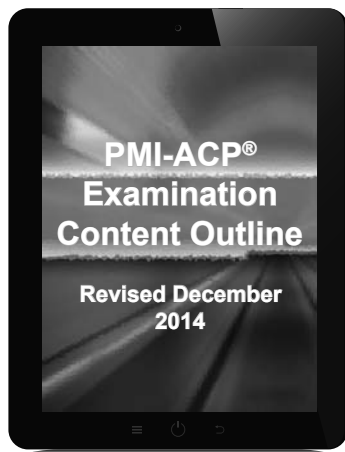
7

Przygotowanie do egzaminu PMI-ACP®

Sprawdź swoją wiedzę

Podczas lektury ostatnich sześciu rozdziałów musiałeś sobie przyswoić całkiem sporo wiedzy!

Poznałeś wartości i zasady *Manifestu programowania zwinnego*. Dowiedziałeś się, jak pomagają one w rozwijaniu zwinnego nastawienia zespołu. Zobaczyłeś, jak zespoły używają Scruma do zarządzania projektami, odkryłeś, jak wygląda zaawansowana inżynieria w XP, i rozeznałeś się w tym, jak zespoły doskonalą własne procesy przy użyciu Lean i Kanbana. Czas **to wszystko podsumować** i utrwalić najważniejsze informacje. **Egzamin PMI-ACP®** wymaga jednak nie tylko znajomości narzędzi, technik i pojęć zwinnych. Aby zdać egzamin wzorowo, będziesz się musiał jeszcze dowiedzieć, jak wykorzystać je w prawdziwym świecie. Spójrzmy zatem *świeżym okiem na te pojęcia*, wspomagając się **pełnym zestawem ćwiczeń, łamigłówek i pytań próbnych** (a także nowym materiałem), skonstruowanych tak, aby łatwiej Ci było przygotować się do egzaminu PMI-ACP®.



Certyfikacja PMI-ACP® jest wartościowa...	308
Egzamin PMI-ACP® oparty jest na oficjalnym sylabusie	309
„Jesteś specjalistą w zakresie podejść zwinnych...”	310
Maraton dla umysłu	313
Dziedzina 1. Zwinne zasady i nastawienie	314
Dziedzina 1. — pytania egzaminacyjne	316
Dziedzina 2. Dostawa oparta na wartości	322
Zespoły zwinne posługują się wartością dla klienta w priorytetyzacji wymagań	325
Obliczenia wartości pomagają określić, które projekty należy realizować	326
Dziedzina 2. — pytania egzaminacyjne	330
Dziedzina 3. Zaangażowanie interesariuszy	336
Dziedzina 4. Wydajność zespołu	337
Dziedzina 3. — pytania egzaminacyjne	338
Dziedzina 4. — pytania egzaminacyjne	339
Dziedzina 5. Planowanie adaptacyjne	348
Dostosowuj swój styl przywództwa do rozwoju zespołu	349
Kilka dodatkowych narzędzi i technik na koniec	351
Dziedzina 6. Wykrywanie i rozwiązywanie problemów	360
Dziedzina 7. Ciągłe doskonalenie	361
Dziedzina 5. — pytania egzaminacyjne	372
Dziedzina 6. — pytania egzaminacyjne	373
Dziedzina 7. — pytania egzaminacyjne	374
Czy jesteś gotów na egzamin końcowy?	376

8

Odpowiedzialność profesjonalisty

Podejmowanie słusznych decyzji

Nie wystarczy znać się na rzeczy. Trzeba umieć podejmować słuszne decyzje, aby radzić sobie w pracy.

Wszystkie osoby, które uzyskują certyfikację PMI-ACP, zgadzają się na przestrzeganie *Kodeksu etyki i profesjonalnego postępowania PMI*. Kodeks pomaga w podejmowaniu **decyzji etycznych**, które nie są szczególnie dobrze omówione we właściwym korpusie wiedzy, ale mogą się pojawić na egzaminie PMI-ACP®. Wiadomości z tego zakresu są w większości **dość łatwe do przyswojenia**, więc wystarczy krótki ich przegląd, żebyś je sobie bez problemu utrwalił.

Właściwe postępowanie	378
Główne założenia	378
Kasa do kieszeni?	380
Może klasa biznesowa?	381
Nowe oprogramowanie	382
Droga na skróty	383
Niskie ceny czy czyste rzeki?	384
Nie jesteście święci	385



Praktyka czyni mistrza

Próbny egzamin PMI-ACP®

9

Pewnie nie sądziłeś, że zajdziesz aż tak daleko!

To była długa podróż, ale w końcu jesteś gotowy na powtórzenie wiadomości i przygotowanie się do egzaminu. Przyswoiłeś sobie dużo nowej wiedzy o podejściach zwinnych i przyszła pora, by sprawdzić, ile z tych informacji zostało Ci w głowie. Dlatego właśnie przygotowaliśmy ten pełnoprawny, składający się ze 120 pytań egzamin próbny PMI-ACP®. **Opieraliśmy się na tym samym sylabusie PMI-ACP®**, którego używają eksperci z PMI; nasz egzamin *wygląda więc tak samo jak ten właściwy*, który przyjdzie Ci zdawać. Czas zatem wysilić szare komórki. Weź głęboki oddech, zakasaj rękawy i do dzieła!

Zanim sprawdzisz odpowiedzi...

423

S

Skorowidz

449

1. Czym jest metodyka zwinna?

Założenia i praktyka

ZALÓŻMY ZATEM,
ŻE PROJEKT ZREALIZUJEMY
BEZBŁĘDNIE,
I ZAPISZMY TO W PLANIE.



Zwinność jest dziś modna jak nigdy dotąd! Naszej branży udało się po raz pierwszy odkryć praktyczny, zrównoważony sposób rozwiązywania problemów, z którymi musiały się męczyć całe pokolenia zespołów programistycznych. Zespoły zwinne posługują się jasnymi, prostymi **praktykami**, które sprawdziły się już w codziennej pracy nad rzeczywistymi projektami. Ale zaraz... Jeśli metodyka zwinna (ang. *agile*) jest tak wspaniała, to dlaczego nie korzystają z niej wszyscy? Wygląda na to, że w prawdziwym świecie te praktyki, które sprawdzają się w jednych zespołach, innym sprawiają trudności, a o takich rozbieżnościach decydują różnice w **nastawieniu**. Przygotuj się więc na zupełnie inny sposób myślenia o swoich projektach!

Ta premia chyba nam jednak nie skapnie

Nowe możliwości brzmią nęcąco...

Poznaj Kate. Kate jest menedżerką projektu w cieszącym się powodzeniem start-upie w Dolinie Krzemowej. Jej firma tworzy oprogramowanie, które daje serwisom streamingowym z filmami i muzyką oraz radiom internetowym możliwość analizowania danych o odbiorcach na żywo i proponowania im treści dostosowanych do ich gustów. Zespół Kate ma teraz szansę na stworzenie czegoś niezwykle korzystnego dla firmy.

ODEZWĘ
SIĘ DO ZESPOŁU
W SPRAWIE WPROWADZENIA
TYCH ZAAWANSOWANYCH FUNKCJI
ANALIZY DANYCH ODBIORCÓW
W KOLEJNYM WYDANIU.

Kate jest menedżerką
projektu w zespole
programistycznym.



DZIĘKI, KATE. JEŚLI
ZESPÓŁ ZROBI TO SZYBKO, NASZ
GŁÓWNY KLIENT WYKUPI PIĘĆDZIESIĄT
DODATKOWYCH LICENCJI I WSZYSCY
DOSTANIEMY PREMIĘ!

Ben jest właścicielem
produktu. Do jego obowiązków
należy komunikowanie się
z klientami, poznawanie ich
potrzeb i wymyślanie nowych
funkcji, z których będą
chętnie korzystać.



...ale nie wszystko idzie zgodnie z planem

Rozmowa Kate z zespołem projektowym nie poszła tak dobrze, jak Kate się spodziewała. Co teraz powie Benowi?

Mike jest głównym programistą i architektem.



TO **BRZMI** JAK
NIEZŁA OKAZJA, ŻEBY
ZADOWOLIĆ NASZYCH
KLIENTÓW.

Kate: Jeśli uda nam się wprowadzić te nowe, zaawansowane funkcje analizy danych odbiorców w kolejnym wydaniu, to wszyscy dostaniemy sporą premię.

Mike: Rzeczywiście brzmi to jak niezła gratka.

Kate: Świetnie! Czyli mogę na was liczyć?

Mike: Zaraz, chwila! Jasne, to *brzmi niezłe* — tylko że nic takiego się nie stanie.

Kate: Co?! Mike, nawet sobie tak nie żartuj.

Mike: Gdybyśmy wiedzieli o tym pomyśle cztery miesiące temu, kiedy projektowaliśmy usługę analizy danych odbiorców, to nie byłoby problemu. Sęk w tym, że teraz musielibyśmy wyciąć spory kawał kodu i zastąpić go... Aj, nawet nie chce mi się rozwodzić o technikaliach.

Kate: I bardzo dobrze, bo mało mnie to interesuje.

Mike: Dobrze, czyli skończyliśmy tę rozmowę? Naprawdę muszę wracać do pracy.

Metodyka zwinna na ratunek!

Kate czytała ostatnio o zwinnym programowaniu i sądzi, że ta metodyka może pomóc we wprowadzeniu pożądaných funkcji do kolejnego wydania. Metodyka zwinna zdobyła niemałą popularność wśród zespołów programistycznych, ponieważ te, które już się przekonały do „zwinności”, często się chwalą znakomitymi wynikami. Ich oprogramowanie jest sprawniejsze, co ma znaczenie zarówno dla nich samych, jak i dla *użytkowników*. Co więcej, zespołom efektywnie stosującym metodykę zwinną po prostu przyjemniej się pracuje! Wszyscy są o wiele bardziej wyluzowani, a środowisko pracy jest przyjaźniejsze.

Dlaczego zatem zwinne wytwarzanie zyskało taką popularność? Oto kilka powodów:

- ★ Zespołom zwinnym łatwiej przychodzi dotrzymanie terminów.
- ★ Można sprawnie zredukować liczbę defektów oprogramowania.
- ★ Konserwacja kodu jest łatwiejsza — dodawanie, rozszerzanie i modyfikowanie nie sprawia już problemów.
- ★ Użytkownicy są o wiele bardziej zadowoleni, co ułatwia życie wszystkim zainteresowanym.
- ★ Największym plusem efektywnych zespołów zwinnych jest to, że pracownikom żyje się lepiej, ponieważ kończą pracę o rozsądnej porze i rzadko muszą pracować w weekendy (co jest dla programisty nie lada gratką!).

Codziennie spotkania to dobry punkt wyjścia

Jedną z najbardziej standardowych praktyk zwinnych wykorzystywanych przez zespoły jest **codzienny stand-up**, czyli spotkanie, podczas którego członkowie zespołu rozmawiają o tym, nad czym pracują i z jakimi wyzwaniem się to wiąże. Dzięki temu, że wszyscy muszą w trakcie spotkania stać, nie jest ono nadmiernie przeciągane. Wprowadzenie codziennego spotkania do pracy nad projektami okazało się pomocne dla wielu zespołów i często jest to pierwszy krok na drodze do wdrożenia metodyki zwinnej.

W trakcie codziennego stand-upu wszyscy członkowie zespołu stoją. Dzięki temu spotkanie jest krótkie, rzeczowe i przebiega w swobodnej atmosferze.



Ale czy ten koleś rzeczywiście słucha, co jego współpracownicy mają do powiedzenia?

Kate próbuje przeprowadzić codzienny stand-up

Ku zdumieniu Kate nie wszyscy członkowie zespołu Mike'a podzielają jej zainteresowanie tą nową praktyką. Co więcej, jeden z programistów jest oburzony propozycją wprowadzenia kolejnego zebrania i nie podoba mu się wizja odpowiadania codziennie na natrętne pytania dotyczące jego pracy.

NOWE FUNKCJE SĄ BARDZO WAŻNE.
ZORGANIZUJMY CODZIENNE SPOTKANIA, ŻEBYŚCIE MOGLI
NA BIEŻĄCO MÓWIĆ MI O POSTĘPACH. TO ŚWIETNA PRAKTYKA
ZWINNA, KTÓRA PRZYDA SIĘ NAM WSZYSTKIM!

JUŻ I TAK MAMY ZA DUŻO
ZEBRAŃ! JEŚLI NIE WIERZYSZ, ŻE
POTRAFIMY WYKONAĆ SWOJĄ PRACĘ,
TO MOŻE OD RAZU ODEZWIJ SIĘ
DO INNEGO ZESPOŁU.

↑
Kate sądzi, że Mike wraz ze swoim
zespołem zachowuje się irracjonalnie
— może jednak mają oni w jakimś
sensie rację? A co Ty o tym sądzisz?



WYSIŁ SZARE KOMÓRKI

Co tu się dzieje? Czy Mike zachowuje się irracjonalnie? Czy Kate jest zbyt wymagająca? Dlaczego ta prosta, powszechnie akceptowana praktyka stała się przyczyną konfliktu?

Różni członkowie zespołu mają różne podejścia

Kate natrafiła na problemy z wdrażaniem metodyki zwinnej już na samym początku. Nie jest w tym jednak odosobniona.

Prawda jest taka, że wiele zespołów zwyczajnie nie odniosło takich sukcesów z wykorzystaniem metodyki zwinnej, jakich się spodziewano. Czy wiedziałaś, że ponad połowa firm zajmujących się produkcją oprogramowania eksperymentowała z metodami zwinnymi? Pomimo opowieści o sukcesach — których jest niemało! — wiele zespołów wypróbowało metodykę zwinną, ale uzyskuje mało zadowalające rezultaty. W rzeczywistości ludzie często czują się nabici w butelkę! Metodyka zwinna zdawała się obiecywać im olbrzymie korzyści, ale wdrożenie jej w ramach ich własnych projektów nie przynosiło odpowiednich efektów.

Kate mierzy się z taką właśnie sytuacją. Samodzielnie opracowała plan i teraz chce otrzymywać aktualne dane od zespołu. Zaczęła wobec tego od zaciągania niechętnych członków zespołu na codzienne stand-upy. Oczywiście jest w stanie wepchnąć ich na siłę do sali, ale co z tego? Kate przejmuje się tym, że ludzie nie działają zgodnie z jej planem, więc zaczyna się koncentrować na wyciąganiu aktualnych danych od każdego z pracowników. Mike wraz z innymi programistami chce natomiast, żeby spotkanie skończyło się szybko i żeby wszyscy mogli wrócić do „prawdziwej” pracy.

DOWIADUJĘ SIĘ
O PROBLEMACH, KIEDY JEST
JUŻ ZA PÓŹNO, ŻEBY COŚ
Z TYM ZROBIĆ.

W trakcie mało skutecznego codziennego spotkania, które Kate zarządziła, każdy z uczestników wyłącza się, gdy inny coś mówi, a wszyscy starają się powiedzieć jak najmniej. Kate udaje się wprawdzie zdobyć jakieś przydatne informacje, ale ceną tego jest konflikt i ogólne wrażenie nudy, przy czym żaden z uczestników spotkania nie jest zainteresowany nawiązaniem współpracy.

KIEDY JESTEŚMY CIĄGANI
PO SPOTKANIACH, NIE MAMY CZASU
NA PISANIE KODU.



Jeśli mamy dwie osoby, to każda z nich może postrzegać tę samą praktykę bardzo odmiennie. Jeżeli obie nie mają poczucia, że jakoś na tym korzystają, współpraca może się okazać mało skuteczna.



CZY NIE TAK WŁAŚNIE WYGLĄDAJĄ
PROJEKTY ROZWOJU OPROGRAMOWANIA? COŚ,
CO PREZENTUJE SIĘ DOBRZE W PODRĘCZNIKU, NIEKONIECZNIE
MUSI SIĘ SPRAWDZAĆ W PRAKTYCE. NIE DA SIĘ Z TYM
NICZEGO ZROBIĆ, PRAWDA?

A właśnie że się da! Właściwe nastawienie sprawia, że takie praktyki są skuteczniejsze.

Powiedzmy sobie jasno: sposób, w jaki Kate prowadzi stand-upy, jest dość standardowy. Choć nie jest to optymalne rozwiązanie, codzienne spotkania przeprowadzane w ten sposób *mimo wszystko przynoszą jakieś rezultaty*. Kate dowiaduje się, co jest nie tak z jej planem, a zespół Mike'a na dłuższą metę na tym korzysta, ponieważ może dzięki temu rozwiązywać ewentualne problemy na wcześniejszym etapie prac. Same spotkania nie zajmują dużo czasu, więc — tak czy inaczej — warto je organizować.

Istnieje jednak spora różnica między zespołem zwinnym, który jedynie wykonuje swoje obowiązki, a takim, który uzyskuje świetne wyniki. Decyduje o niej **nastawienie**, z jakim zespół podchodzi do każdego projektu. Może się to wydawać nieprawdopodobne, ale nastawienie osób zaangażowanych w stosowanie danej praktyki wpływa na jej efektywność!



Nastawienie każdego z członków zespołu wobec praktyki takiej jak codzienny stand-up istotnie wpływa na jej efektywność. Spotkanie takie może jednak być wystarczająco skuteczne, nawet jeśli jest nudne i nikt nie jest w nie zaangażowany.

Praktyki stosuje się sprawniej dzięki odpowiedniemu nastawieniu

Co by się stało, gdyby Kate i Mike mieli inne nastawienie? Co by było, gdyby każdy członek zespołu miał *zupełnie inne podejście* do codziennego stand-upu?

Co by się na przykład stało, gdyby Kate miała poczucie, że wszyscy członkowie zespołu *współpracują* przy planowaniu projektu? W takiej sytuacji byłaby ona skłonna rzeczywiście wysłuchać każdego programisty. Gdyby Kate zmieniła swoje nastawienie do stand-upu, przestałaby zwracać uwagę na to, jak bardzo praktyka odbiega od *jej* planów, i nie skupiałaby się na ciągłym poprawianiu innych. Zaczęłaby się jednocześnie bardziej koncentrować na zrozumieniu planu będącego owocem wspólnej pracy, a jej zadaniem byłoby pomaganie całemu zespołowi w efektywniejszym wykonywaniu obowiązków.

Jest to takie podejście do planowania, z jakim Kate nigdy nie miała styczności w ramach nauki zarządzania projektami, bardzo odmienne od tego, które było jej znane. Zawsze uczono ją, że jej zadaniem jest opracowanie planu projektu i narzucenie go zespołowi. Dano jej do dyspozycji narzędzia pozwalające na sprawdzenie tego, jak dokładnie zespół przestrzega jej planu, jak również sztywne procesy służące do wprowadzania zmian w sposobie działania zespołu.

Teraz wszystko wygląda inaczej. Kate zrozumiała, że sprawne przeprowadzenie codziennego stand-upu wymaga **włożenia pewnego wysiłku w nawiązanie współpracy z zespołem**, tak aby wszyscy jego członkowie mogli wspólnie sformułować najlepsze podejście do projektu. Dzięki temu spotkanie stanie się dla całego zespołu sposobnością do współdziałania i upewnienia się, czy wszyscy podejmują właściwe decyzje i czy projekt rozwija się we właściwym kierunku.

NIE ZNAM
WSZYSTKICH ODPOWIEDZI.
POTRZEBUJEMY TAKICH SPOTKAŃ,
ABY MÓC ZAPLANOWAĆ PROJEKT
RAZEM!

↑
Kate bardzo się irytowała, kiedy trafiała na zmiany wprowadzone do jej planu projektowego, ponieważ na ogół było wtedy już za późno, żeby skierować projekt na właściwe tory.

↑
Ponieważ wprowadzono codzienny stand-up do planu dnia, cały zespół codziennie pomaga Kate w wykrywaniu takich zmian, tak aby można było poczynić właściwe kroki odpowiednio wcześnie. Takie rozwiązanie jest o wiele sprawniejsze!



CZYLI CODZIENNY STAND-UP POLEGA NA TYM, ŻE **SŁUCHASZ, CO JA I MÓJ ZESPÓŁ MAMY DO POWIEDZENIA**, I RZECZYWIŚCIE DOSTOSOWUJESZ DO TEGO PRZEBIEG REALIZACJI PROJEKTU?



Co by było, gdyby Mike doszedł do wniosku, że takie spotkanie nie polega jedynie na informowaniu o stanie prac, tylko na *omawianiu przebiegu projektu* i wymyślaniu sposobów na usprawnienie pracy? Taki codzienny stand-up prawdopodobnie stałby się dla niego ważnym punktem dnia.

Dobry programista niemal zawsze ma wyrobione zdanie nie tylko na temat własnego kodu, ale także na temat ogólnego kierunku rozwoju całego projektu. Codzienny stand-up jest dla niego okazją, aby upewnić się, czy projekt realizowany jest rozsądnie i wydajnie. Mike wie, że jeśli cały projekt jest sprawnie realizowany, to jego kod będzie miał na dłuższą metę większą wartość dla zespołu. Wie też, że jeśli zgłosi na spotkaniu problem związany z planem, *wszyscy go wysłuchają*, a dzięki temu realizacja projektu będzie efektywniejsza.

Wszystko działa najlepiej, kiedy Mike (wraz z resztą zespołu) jest świadom tego, że codzienne spotkanie pomaga w rozplanowaniu prac na następny dzień, a każdy członek zespołu uczestniczy w planowaniu.

TO MA SENS! PLANOWANIE PRZEBIEGA O WIELE SPRAWNIEJ, KIEDY WSZYSCY CZŁONKOWIE ZESPOŁU BIORĄ W NIM UDZIAŁ. TYLE TYLKO, ŻE CHYBA WSZYSCY UCZESTNICY MUSZĄ BYĆ ZAANGAŻOWANI W ROZMOWĘ PRZEZ CAŁY CZAS, ŻEBY TO MOGŁO ZADZIAŁAĆ.



WYSIL SZARE KOMÓRKI

Jak zmienić nastawienie całego zespołu lub poszczególnych jego członków? Czy przychodzą Ci do głowy jakieś przykłady zmiany czyjegoś nastawienia — może nawet Twojego własnego — z którymi zetknąłeś się w pracy nad własnymi projektami?

Czym właściwie jest ta cała metodyka zwinna?

Metodyki zwinne są zoptymalizowane pod kątem ułatwienia rozwiązywania konkretnych problemów, z jakimi zwykle borykają się zespoły programistyczne, a jednocześnie odznaczają się one pewną prostotą, dzięki której wdrożenie ich jest stosunkowo łatwe.

Takie metodyki odnoszą się do wszystkich obszarów tradycyjnej produkcji oprogramowania, w tym do zarządzania projektem, projektowania oprogramowania i formułowania jego architektury oraz usprawniania procesów. Każda z nich składa się z ustandaryzowanych **praktyk**, zoptymalizowanych pod kątem ułatwienia ich wdrażania.



POŚWIĘCIŁAM TYLE CZASU NA WYMYŚLENIE WŁASNEGO PLANU, ALE ZESPÓŁ CIĄGLE GO ZMIENIA. NA CODZIENNYCH SPOTKANIACH BĘDĘ MOGŁA DOPILNOWAĆ, ŻEBY CZŁONKOWIE ZESPOŁU ROBILI **WSZYSTKO TO, CO IM POWIEM.**

Nastawienie kontra metodyka

Wytwarzanie zwinne wiąże się jednak *również* z pewnym **nastawieniem**, co jest zupełnie nowym pojęciem dla wielu osób, które wcześniej nie miały z nim styczności. Wychodzi na to, że nastawienie każdego członka zespołu do stosowanych praktyk *może znacząco wpłynąć* na efektywność całego zespołu. Nastawienie zwinne skupia się na tym, aby ułatwić ludziom dzielenie się informacjami, co pomaga im w podejmowaniu ważnych decyzji projektowych (i nie muszą zdawać się przy tym na szefa lub menedżera projektu). Polega to na pozostawieniu decyzyjności dotyczącej planowania, projektowania i ulepszania procesów *całemu* zespołowi. Aby wszystkim łatwiej było przyjąć skuteczne nastawienie, każda metodyka zwinna ma własny zestaw **wartości**, którymi członkowie zespołu mogą się kierować.

JEŚLI WSZYSCY BĘDZIEMY **WSPÓLNIE PRACOWAĆ NAD PLANOWANIEM PROJEKTU, TO BĘDZIEMY MOGLI WPROWADZAĆ KOREKTY NA BIEŻĄCO W TRAKCIE CODZIENNYCH SPOTKAŃ.**



WYSIL SZARE KOMÓRKI

Co się dzieje, kiedy jeden z członków zespołu wyłącza się w trakcie codziennego stand-upu i nie słucha tego, co mówią inni?

Zaostrz ołówek



Oto kilka problemów, na które Kate, Ben i Mike zwrócili uwagę w trakcie codziennego stand-upu. Obok nich widnieją nazwy różnych praktyk, które są często wykorzystywane przez zespoły zwinne. Nie martw się, jeśli po raz pierwszy stykasz się z niektórymi z nich — przeczytasz o nich więcej w dalszej części książki, dlatego tutaj zamieszczone zostały jedynie krótkie opisy. Sprawdź, czy możesz dopasować poszczególne problemy do praktyk ułatwiających ich rozwiązanie.



„ZMARNOWALIŚMY BARDZO DUŻO CZASU NA PRZEKOPYWANIE SIĘ PRZEZ KOD SPAGHETTI, ŻEBY ZNALEZĆ TEN BŁĄD!”.



„DOBRA, OMÓWILIŚMY HISTORYJKI UŻYTKOWNIKÓW. ZASTANÓWMY SIĘ TERAZ, JAK SKŁADAJĄ SIĘ ONE W CAŁOŚĆ, ŻEBYŚMY MOGLI ROZPLANOWAĆ KOLEJNE TYGODNIE PRACY”.



„PRZY KAŻDYM NOWYM WYDANIU CIĄGLE MAMY TE SAME PROBLEMY”.



„WŁAŚNIE PRZEPROWADZIŁEM DEMONSTRACJĘ NOWEJ FUNKCJI DLA JEDNEGO Z NASZYCH UŻYTKOWNIKÓW. POWIEDZIAŁ, ŻE WCALE NIE ROZWIĄDUJE ONA PROBLEMU, JAKI MIAŁ”.



„MYŚLAŁAM, ŻE ZDAŻYMY ZAKTUALIZOWAĆ KOD BAZY DANYCH Z PIOSENKAMI DO PIĄTKU. TERAZ MI MÓWICIE, ŻE POTRWA TO JESZCZE TRZY TYGODNIE?”.

Retrospektywa jest spotkaniem, podczas którego wszyscy dyskutują o przebiegu końcowego etapu projektu i wyciągają wnioski.

Historyjka użytkownika jest sposobem na wyrażenie jednej, konkretnej potrzeby użytkownika, zazwyczaj w formie kilku zdań zapisanych na karteczce samoprzylepnej lub fiszce.

Tablica zadań jest narzędziem do planowania. Korzystanie z niej polega na przyklejaniu karteczek z historyjkami do tablicy i układaniu ich w kolumny na podstawie ich statusu.

Wykres spalania jest aktualizowanym codziennie wykresem liniowym, wskazującym ilość pracy, jaką trzeba jeszcze wykonać, by zrealizować projekt, „spalającą się” do zera, kiedy następuje zakończenie wszystkich zadań.

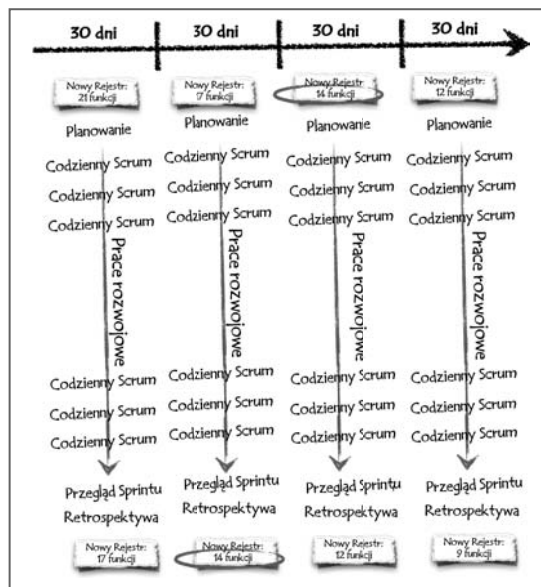
Programiści rozwiązują problemy z kodem poprzez jego ciągłą **refaktoryzację**, czyli poprawianie struktury bez zmiany zachowania.

Nie przejmuj się, jeśli nie zetknąłeś się jeszcze z powyższymi praktykami. Więcej na ich temat przeczytasz w kolejnych rozdziałach.

Scrum jest najpopularniejszym podejściem do wytwarzania zwinnego

Zespoły mogą stosować różnorakie metodyki zwinne, a jest ich co niemiara. Przez lata przeprowadzono jednak wiele badań, z których wynika, że najczęściej korzysta się ze **Scruma**, czyli ram postępowania koncentrujących się na zarządzaniu projektem i rozwoju produktu.

Kiedy zespół korzysta ze Scruma, każdy projekt przebiega według jednakowego wzorca. W projekcie scrumowym występują trzy główne role: **Właściciel Produktu** (jest nim Ben), który pracuje z zespołem nad utrzymaniem Rejestru Produktu, **Scrum Master** (dosł. Mistrz Młyna), który zajmuje się pomaganiem zespołowi w pokonywaniu różnych przeszkód, oraz **członkowie zespołu programistycznego** (czyli wszyscy pozostali). Projekt podzielony jest na **sprinty**, czyli cykle o jednakowym czasie trwania (często dwutygodniowym lub miesięcznym), które przebiegają zgodnie ze wzorcem Scruma. Na początku sprintu zespół przeprowadza **planowanie sprintu**, aby określić, które funkcje z rejestru produktu zostaną w ramach niego opracowane. Zespół przez cały czas trwania sprintu pracuje nad wszystkimi funkcjami, które zawarto w **Rejestrze Sprintu**. Zespół każdego dnia urządza krótkie spotkanie, zwane **codziennym Scrumem**. Pod koniec sprintu działające oprogramowanie zostaje zademonstrowane Właścicielowi Produktu i interesariuszom w ramach **przeglądu sprintu**, a zespół urządza **retrospektywę**, aby omówić wnioski, jakie wyciągnięto w trakcie pracy.

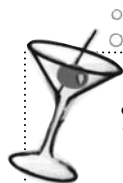


Metodykę Scrum omówimy szczegółowo w rozdziałach 3. i 4.

Dowiedz się, w jaki sposób Scrum pomaga w tworzeniu lepszego programowania i sprawniejszej realizacji projektów, a także wykorzystasz go do zbadania pojęć i koncepcji istotnych dla zespołów stosujących metodykę zwinną.

XP i Lean/Kanban

Choć Scrum jest najpopularniejszą metodyką zwinną, wiele zespołów korzysta z innych podejść. Drugą metodyką pod względem popularności jest **XP**, które koncentruje się na rozwoju oprogramowania i programowaniu, a często używane jest razem ze Scrumem. Inne zespoły stosują metodyki **Lean** i **Kanban** —dają one możliwość zorientowania się, jak wytworzyć oprogramowanie teraz, a ulepszyć je w przyszłości. Więcej o podejściach XP i Lean/Kanban przeczytasz w rozdziałach 5. i 6.



Spokojnie

Za dużo nowych pojęć?

Nowe pojęcia są w tej książce oznaczone **pogrubionym** pismem, kiedy pojawiają się po raz pierwszy. Na tej stronie pojawiło się ich niemało, więc jeżeli stykasz się z niektórymi z nich pierwszy raz, nie przejmuj się! Zapoznanie się teraz z nowymi pojęciami w odpowiednim kontekście pomoże Ci, kiedy później będziesz czytał ich bardziej szczegółowe omówienie. Jest to jeden ze stałych elementów serii *Rusz głową!*, dzięki którym nauka jest łatwiejsza!

Zaostrz ołówki



Rozwiązanie

Zespoły mogą uniknąć popełniania ciągle tych samych błędów, wracając do projektu i omawiając to, co poszło dobrze, oraz to, co można usprawnić.



„ZMARNOWALIŚMY BARDZO DUŻO CZASU NA PRZEKOPYWANIE SIĘ PRZEZ KOD SPAGHETTI, ŻEBY ZNALEZĆ TEN BŁĄD!”.

Tablica zadań jest świetnym sposobem na przedstawienie wszystkim członkom zespołu tego samego oglądu projektu.



„DOBRA, OMÓWILIŚMY HISTORYJKI UŻYTKOWNIKÓW. ZASTANÓWMY SIĘ TERAZ, JAK SKŁADAJĄ SIĘ ONE W CAŁOŚĆ, ŻEBYŚMY MOGLI ROZPLANOWAĆ KOLEJNE TYGODNIE PRACY”.



„PRZY KAŻDYM NOWYM WYDANIU CIĄGLE MAMY TE SAME PROBLEMY”.



„WŁAŚNIE PRZEPROWADZIŁEM DEMONSTRACJĘ NOWEJ FUNKCJI DLA JEDNEGO Z NASZYCH UŻYTKOWNIKÓW. POWIEDZIAŁ, ŻE WCALE NIE ROZWIĄDUJE ONA PROBLEMU, JAKI MIAŁ”.

Kiedy wszyscy członkowie zespołu rozumieją użytkowników i znają ich potrzeby, o wiele sprawniej idzie im tworzenie oprogramowania, które odpowiada użytkownikom.



„MYŚLAŁAM, ŻE ZDAŻYMY ZAKTUALIZOWAĆ KOD BAZY DANYCH Z PIOSENKAMI DO PIĄTKU. TERAZ MI MÓWICIE, ŻE POTRWA TO JESZCZE TRZY TYGODNIE?”.

Retrospektywa jest spotkaniem, podczas którego wszyscy dyskutują o przebiegu końcowego etapu projektu i wyciągają wnioski.

Historyjka użytkownika jest sposobem na wyrażenie jednej, konkretnej potrzeby użytkownika, zazwyczaj w formie kilku zdań zapisanych na karteczce samoprzylepnej lub fiszce.

Tablica zadań jest narzędziem do planowania. Korzystanie z niej polega na przyklejaniu karteczek z historyjkami do tablicy i układaniu ich w kolumny na podstawie ich statusu.

Wykres spalania jest aktualizowanym codziennie wykresem liniowym, wskazującym ilość pracy, jaką trzeba jeszcze wykonać, by zrealizować projekt, „spalającą się” do zera, kiedy następuje zakończenie wszystkich zadań.

To jest praktyka zaczerpnięta z XP. Niektórzy menedżerowie projektu podczas pierwszej styczności z praktykami zwinnymi są zaskoczeni, że dotyczą one samego kodu, a nie planowania i realizacji projektu.

Programiści rozwiązują problemy z kodem poprzez jego ciągłą **refaktoryzację**, czyli poprawianie struktury bez zmiany zachowania.

Podójście zwinne to nie tylko zmiana nazewnictwa

Nie istnieją
głupie pytania

P: Wygląda na to, że Scrum, XP i Lean/Kanban bardzo się od siebie różnią. Jak to możliwe, że wszystkie one są metodykami zwinnymi?

U: Scrum, XP i Lean/Kanban koncentrują się na bardzo różnych obszarach. Scrum skupia się głównie na zarządzaniu projektem: na przedmiocie prac i dostosowaniu go do potrzeb użytkowników oraz interesariuszy. XP koncentruje się na rozwoju oprogramowania: tworzeniu wysokiej jakości kodu, który jest dobrze zaprojektowany i łatwy w utrzymaniu. Lean/Kanban jest połączeniem podejścia Lean z metodą Kanban, którego zespoły używają w celu ciągłego doskonalenia sposobu wytwarzania oprogramowania. Innymi słowy, Scrum, XP i Lean/Kanban skupiają się na trzech różnych obszarach wytwarzania oprogramowania: zarządzaniu projektem, projektowaniu i architekturze oraz

doskonaleniu procesów. Nic więc dziwnego, że ich praktyki różnią się od siebie.

W następnym rozdziale przeczytasz zaś o tym, co je łączy: **wspólnych wartościach i zasadach**, które pomagają zespołom przyjąć odpowiednie nastawienie do stosowania metodyki zwinnej.

P: Czy nie mówimy tutaj o rzeczach, które dobrze znam, tyle tylko, że teraz są one oznaczone nowymi etykietkami? Te sprinty scrumowe to po prostu kamienie milowe i fazy projektu, prawda?

U: Przy pierwszym zetknięciu się z metodologią zwinną taką jak Scrum powszechne jest doszukiwanie się znajomo wyglądających elementów — i bardzo dobrze! Jeśli pracujesz już od jakiegoś czasu w zespole programistycznym, wiele aspektów metodyki zwinnej *powinno* wydać Ci się znajomych. Twój zespół w końcu zajmuje się tworzeniem czegoś, a poza tym pewnie wiele rzeczy chcecie nadal robić tak samo jak dotąd.

Łatwo jednak przyjąć błędne założenie, że znajomo wyglądający element metodyki zwinnej jest dokładnie tym czymś, czego już używasz. Przykładowo sprinty scrumowe *nie są tym samym co fazy projektu*. Istnieje wiele różnic pomiędzy fazami i kamieniami w tradycyjnym zarządzaniu projektem a sprintami w Scrumie.

Przy standardowym planowaniu projektu wszystkie fazy projektu są wyznaczane na samym początku; w Scrumie szczegółowo planuje się natomiast jedynie następny sprint. Ta różnica może się wydać nieco dziwna dla zespołu przyzwyczajonego do tradycyjnego zarządzania projektem.

Już wkrótce dowiesz się o wiele więcej o tym, jak działa planowanie w Scrumie oraz jak odbiega ono od podejść, do których jesteś przyzwyczajony. Na razie zachowaj otwarty umysł i staraj się zwracać uwagę na momenty, w których myślisz sobie: „Przecież ja już to wszystko wiem!”.



CELNE SPOSTRZEŻENIA

- Wiele zespołów wdrażających metodykę zwinną zaczyna od prowadzenia **codziennych stand-upów**, czyli spotkań całego zespołu, podczas których wszyscy stoją, żeby rozmowa się nie przedłużała.
- Podejścia zwinne są nie tylko zestawami **metod i metodyk**, ale też wymagają przyjęcia pewnego nastawienia przez wszystkich członków zespołu.
- Codzienny stand-up jest skuteczniejszy, kiedy wszyscy członkowie zespołu mają takie samo **nastawienie** — kiedy wszyscy się wzajemnie słuchają i wspólnie dążą do tego, żeby projekt rozwijał się prawidłowo.
- Każda metodyka zwinna cechuje się pewnymi **wartościami**, które pomagają zespołowi przyjąć odpowiednie nastawienie.
- Kiedy członkowie zespołu przestrzegają **wspólnych zasad i wartości**, wykorzystywana przez nich metoda staje się o wiele efektywniejsza.
- **Scrum**, metodyka koncentrująca się na zarządzaniu projektem i rozwoju produktu, jest najczęściej stosowanym podejściem zwinnym.
- W projekcie scrumowym zespół dzieli pracę na **sprinty**, czyli cykle o równej długości (często są to okresy miesięczne) oparte na wzorcu Scruma.
- Każdy sprint zaczyna się od sesji **planowania sprintu**, określającej, jakie prace mają zostać wykonane.
- W trakcie sprintu zespół pracuje nad projektem i codziennie organizuje krótkie spotkanie, nazywane **codziennym Scrumem**.
- Pod koniec sprintu zespół urządza **przeгляд sprintu** z udziałem interesariuszy, aby zademonstrować działające oprogramowanie.
- Na zakończenie sprintu zespół organizuje **retrospektywę**, aby omówić przebieg sprintu i możliwości poprawy dla całego zespołu.



Nie lekceważ znaczenia nastawienia

Obejrzyj to!

Wielu ludzi — zwłaszcza zatwardziałych programistów — przewraca oczami, kiedy tylko słyszy pojęcia takie jak nastawienie, wartości i zasady. Największą skłonnością do wyłączenia się w takich momentach mają koderzy, którzy wolą siedzieć sami w swojej kanciapie i do nikogo się nie odzywać. Jeśli sam zaczynasz tak reagować, to postaraj się jednak dać tym pojęciom szansę. Bądź co bądź niemało świetnego oprogramowania powstało dzięki zastosowaniu tej metodyki, więc coś musi być na rzeczy... prawda?



Zaostrz ołówek

Które z poniższych scenariuszy są przykładami zastosowania praktyk, a które wykorzystania zasad? Nie martw się, jeśli pierwszy raz stykasz się z niektórymi z tych praktyk — postaraj się wywnioskować właściwe odpowiedzi z kontekstu. (Takie podejście przydaje się zresztą na egzaminach certyfikacyjnych!).

1. Kate wie, że najskuteczniejszym sposobem na przekazywanie swojemu zespołowi ważnych informacji o projekcie jest **rozmowa twarzą w twarz**.

Zasada

Praktyka

2. Mike i jego zespół wiedzą, że użytkownicy prawdopodobnie zmienią później swoje zdanie, a wprowadzanie zmian może się okazać katastrofalne dla kodu, wobec czego zespół zapobiegawczo posługuje się **projektowaniem przyrostowym**, aby wytworzony kod można było z łatwością zmienić na późniejszym etapie.

Zasada

Praktyka

3. Ben używa **person** do modelowania typowych użytkowników, ponieważ wie, że im lepiej zespół ich rozumie, tym sprawniej będzie sobie radzić z tworzeniem oprogramowania.

Zasada

Praktyka

4. Mike zawsze pilnuje, by jego zespół pracował nad czymś, co będzie mógł później zaprezentować Kate i Benowi, ponieważ wie, że **działające oprogramowanie** świadczy o postępach poczynionych przez zespół.

Zasada

Praktyka

5. Kate chce usprawnić sposób, w jaki zespół wytwarza oprogramowanie, więc zachęca go do **wspólnego doskonalenia się i eksperymentalnego rozwoju**, wymyślając zmiany, jakie może on wprowadzić do swoich procesów, a także używając danych do stwierdzenia, czy zmiany te rzeczywiście okazały się korzystne.

Zasada

Praktyka

6. Mike i jego zespół **aktywnie ustosunkowują się do zmian**, tworząc kod, który będzie można łatwo zmienić na późniejszym etapie.

Zasada

Praktyka

→ Odpowiedzi na s. 20

Większa widoczność to lepsze wyniki (prawda?)



NIGDY DOTĄD NAM SIĘ TAK
DOBRZE NIE WSPÓŁPRACOWAŁO!
TEN CODZIENNY STAND-UP ZMIENIŁ
WSZYSTKO!

Kate: Ten projekt idzie nam dużo lepiej niż wcześniejsze. A wszystko dzięki temu, że codziennie urządzamy jedno krótkie spotkanie!

Mike: Nie powiedziałbym.

Kate: A możesz przestać być takim malkontentem?

Mike: Serio mówię. Chyba nie myślisz, że nikt przed tobą nie wpadł na to, żeby rozwiązywać nasze problemy z projektami poprzez urządzenie dodatkowych spotkań?

Kate: E... No...

Mike: No dobra, udało nam się uzyskać naprawdę niezłe wyniki, więc będę z tobą szczerzy. Kiedy zaczęłaś prowadzić te codzienne stand-upy, prawie cały zespół był niezadowolony z tego pomysłu.

Kate: Naprawdę?

Mike: Nie pamiętasz, jak przez pierwsze półtora tygodnia większość z nas po prostu gapiała się w swoje telefony?

Kate: No tak, wtedy rzeczywiście niewiele z tego wynikało. Też muszę szczerze powiedzieć, że myślałam wówczas o odwołaniu tego wszystkiego.

Mike: W końcu jednak jedna z programistek zwróciła uwagę na poważny problem z architekturą. Wszyscy jej wysłuchali, bo jest naprawdę dobra w tym, co robi, i wszyscy szanują jej zdanie.

Kate: Rzeczywiście. Musieliśmy wprowadzić większą zmianę, więc wycięłam dwie funkcje z kolejnego wydania, żebyśmy z tym wszystkim zdążyli.

Mike: Tak! I to właśnie okazało się naprawdę istotne. Zazwyczaj, kiedy wykrywamy takie problemy, musimy pracować po nocach, żeby sobie z nimi jakoś poradzić. Tak jak wtedy, kiedy dowiedzieliśmy się, że algorytm analizy danych o słuchaczach zawierał poważny błąd.

Kate: Jeju, to było straszne. Zwykle dowiaduję się o takich problemach po tym, kiedy już obiecuję dostarczenie czegoś, czego nie możemy zapewnić. Tym razem wykryliśmy problem wcześniej i mogłam razem z Benem wpłynąć na oczekiwania użytkowników, a wam dać czas na wymyślenie nowego rozwiązania.

Mike: A my na pewno będziemy informować o takich problemach, kiedy tylko się pojawią.

Kate: Czekaż, co? Takie problemy zdarzają się często?!

Mike: Żartujesz? Nigdy jeszcze nie realizowałem projektu, w którym nie trafiłbym przynajmniej raz na taką niespodziankę w trakcie kodowania. Tak wygląda realizacja projektów programistycznych w prawdziwym świecie.

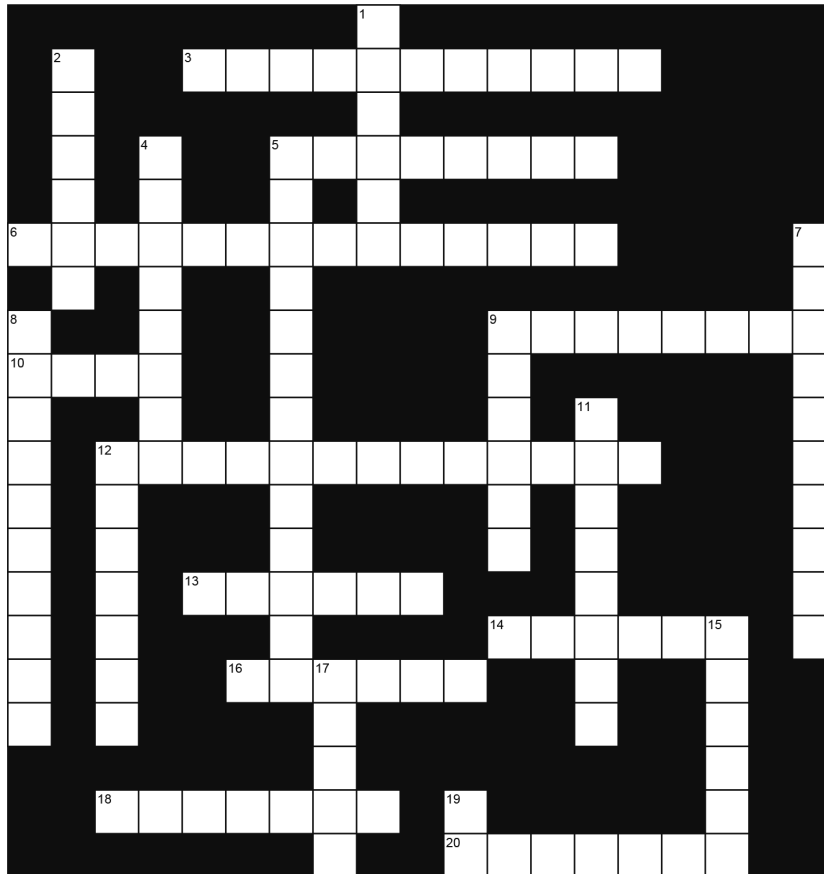
Kate właśnie uświadomiła sobie, że realizacja projektów programistycznych jest o wiele bardziej złożona i chaotyczna w praktyce niż na papierze. Wcześniej mogła opracowywać plany samodzielnie i narzucać je zespołowi, a kiedy coś szło nie tak, mogła zaktadać, że to z winy zespołu, a nie jej.

Jednakże ostatni projekt był o wiele bardziej udany od poprzednich. Kate musiała dotożyć starań, żeby poradzić sobie z problemami, ale w zamian uzyskała lepsze wyniki!



Krzyżówka

Rozwiąż krzyżówkę i zapoznaj się z poniższymi terminami dotyczącymi metodyk zwinnych! Ile poprawnych odpowiedzi podasz bez przeglądania tego rozdziału?



Poziomo

3. Ma duże znaczenie dla wdrażania praktyk
5. Narzędzie lub technika stosowane przez zespół
6. Zespoły XP robią to w celu ciągłego doskonalenia struktury kodu
9. Scrum lub XP
10. Jedna z metodyk, stosowana często w połączeniu z Kanbanem
12. Spotkanie, na którym zespół omawia wnioski wyciągnięte z pracy
13. Metodyka koncentrująca się na doskonaleniu procesu
14. Na jego doskonaleniu koncentruje się Kanban
16. Pracuje wspólnie nad projektem
18. Codzienny _____ bywa przydatny, ale sprawdza się najlepiej, kiedy wszyscy członkowie zespołu mają właściwe nastawienie
20. Jego Właściciel w zespole scrumowym zarządza rejestrem

Pionowo

1. Obok wartości pomagają zespołom przyjąć odpowiednie nastawienie do stosowania metodyki zwinnej

2. _____ spalania — wskazuje ilość pracy, jaka pozostała do wykonania
4. Wykres wskazujący ilość pracy, jaką trzeba jeszcze wykonać, by zrealizować projekt
5. _____ przyrostowe — praktyka XP, w ramach której zespół rozwija bazę kodu przyrostowo wraz z upływem czasu
7. _____ Produktu — zarządza rejestrem w zespole scrumowym
8. Zespoły scrumowe robią to wspólnie na początku prac nad projektem
9. Scrum _____ pomaga zespołowi w pokonywaniu przeszkód i we wdrożeniu metodyki Scruma
11. Cecha takich metodyk jak Scrum czy XP
12. Znajdują się w nim funkcje, których jeszcze nie stworzono
15. Chronologiczna jednostka podziału projektu scrumowego
17. Ramy działania koncentrujące się na zarządzaniu projektem i rozwoju produktu
19. Metodyka koncentrująca się na kodzie i projektowaniu oprogramowania

Certyfikacja PMI-ACP pomoże Ci wyćwiczyć zwinność

Certyfikacja Agile Certified Practitioner (PMI-ACP)[®] została przygotowana przez Project Management Institute w odpowiedzi na potrzeby menedżerów projektów, którzy coraz częściej muszą pracować z metodami, metodykami, praktykami i technikami zwinnymi. Podobnie jak w przypadku certyfikacji PMP, PMI stworzyło egzamin odnoszący się do zadań, narzędzi i praktyk stosowanych na co dzień przez zespoły zwinne.

Certyfikacja PMI-ACP przeznaczona jest dla wszystkich osób pracujących w zespołach zwinnych lub w organizacjach przechodzących na metodyki zwinne.

Pani doktor pomoże Ci zadbać o kondycję i przygotować się do egzaminu.

JESLI ZAMIERZASZ PODEJŚĆ DO EGZAMINU, PRZYGOTUJESZ SIĘ DO NIEGO, ODPOWIADAJĄC NA PYTANIA PRÓBNE, KTÓRE POMOGĄ CI POWTÓRZYĆ MATERIAŁ Z KAŻDEGO ROZDZIAŁU.

Egzamin sprawdza praktyczną wiedzę o wytwarzaniu zwinnym.

Egzamin PMI-ACP odzwierciedla sposób, w jaki zespoły rzeczywiście pracują. Obejmuje on najczęściej stosowane metody i metodyki, takie jak Scrum, XP i Lean/Kanban. Pytania egzaminacyjne nawiązują do zadań, które zespoły wykonują na co dzień.

Niniejsza książka ma za zadanie przede wszystkim **przekazać Ci wiedzę o podejściach zwinnych** właśnie dlatego, że zrozumienie metod, metodyk, praktyk, wartości i koncepcji zwinnych jest najskuteczniejszym sposobem przygotowania się do egzaminu PMI-ACP.

Poza przedstawieniem Ci szerokiego omówienia metodyk zwinnych zajmiemy się też konkretnie materiałami egzaminacyjnymi. Książka ta prezentuje **pełny zakres egzaminacji PMI-ACP**. Znajduje się w niej wiele pytań próbnych, wskazówek testowych i ćwiczeń przygotowujących, w tym pełny egzamin próbny, dobrze odzwierciedlający ten właściwy.



NAWET JEŚLI NIE ZAMIERZASZ ZDAWAĆ EGZAMINU PMI-ACP, ZAWARTE W TEJ KSIĄŻCE PYTANIA PRÓBNE DADZĄ CI MOŻLIWOŚĆ SPOJRZENIA NA MATERIAŁ Z INNEJ STRONY. TO ŚWIETNY SPOSÓB NA SKUTECZNE PRZYSWOJENIE SOBIE WIEDZY!



Rozdziały 2. – 7. kończą się zestawami próbnych pytań egzaminacyjnych. Znajdziesz w nich też podrozdziały „Klinika naukowa”, w których omówimy różne rodzaje pytań, z jakimi zetkniesz się na egzaminie.

Umiejętność odróżnienia różnych typów pytań na egzaminie jest przydatna, ponieważ widok czegoś znajomego pomoże Ci się odprężyć, a tym samym szybciej udzielać odpowiedzi. Poniższy przykład ukazuje pytanie typu „suche fakty”. Na takie pytanie należy odpowiedzieć wskazaniem jakiejś prostej informacji, ale należy pamiętać o *ważnym przeczytaniu wszystkich odpowiedzi!* Często bowiem pojawiają się odpowiedzi, które wydają się poprawne, ale wcale takie nie są.

39. Czego zespoły używają do analizowania postępów prac nad projektem?

A. Refaktoryzacji

Niektóre odpowiedzi są w oczywisty sposób błędne. Refaktoryzacja dotyczy ulepszania kodu, a nie oceny postępów.

B. Retrospektyw

Niektóre odpowiedzi bywają mylące! Retrospektywa pomaga zespołowi lepiej zrozumieć projekt, ale nie pozwala na śledzenie postępów, jako że obejmuje ona przede wszystkim tę część pracy, którą już wykonano.

C. Wykresów spalania

Poprawna odpowiedź! Wykres spalania jest narzędziem ukazującym dotychczasowe postępy prac oraz ilość pracy, którą trzeba jeszcze wykonać.

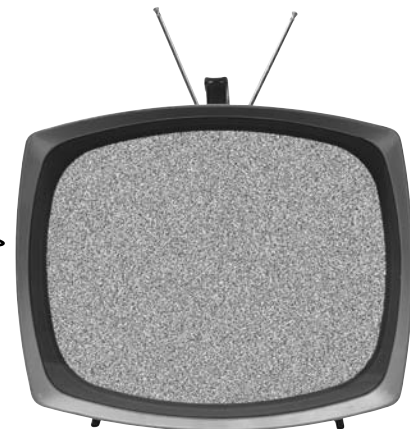
D. Ciągłej integracji

Pytania próbne dostarczą Ci wskazówek i pomogą opracować strategię...



Nie spotkałeś się jeszcze z tym terminem — jest to praktyka stosowana przez zespoły XP. W niektórych pytaniach egzaminacyjnych trafisz na niezrozumiałe odpowiedzi. Wszystko w porządku! Odpręż się i skoncentruj na pozostałych. W tym przypadku jedna z nich jest poprawna. Jeśli jednak zdarzy się tak, że żadna z innych odpowiedzi nie będzie trafna, to będziesz mógł je kolejno eliminować i podjąć racjonalną próbę odgadnięcia poprawnej!

...A PONADTO W PODROZDZIAŁACH „KLINIKA NAUKOWA” POMOŻEMY CI SIĘ PRZYGOTOWAĆ DO EGZAMINU PMI-ACP, OMAWIAJĄC RÓŻNE RODZAJE PYTAŃ, A TAKŻE DAJĄC CI OKAZJĘ DO SAMODZIELNEGO NAPISANIA WŁASNYCH!





Zaostrz ołówek Rozwiązanie

Jest to jedna z zasad metodyk zwinnych: rozmowa twarzą w twarz jest najskuteczniejszym sposobem na przekazywanie informacji zespołowi programistycznemu, jak również w jego obrębie.

1. Kate wie, że najskuteczniejszym sposobem na przekazywanie swojemu zespołowi ważnych informacji o projekcie jest **rozmowa twarzą w twarz**.



Zasada



Praktyka

Projektowanie przyrostowe jest praktyką XP, w ramach której zespół rozwija bazę kodu przyrostowo wraz z upływem czasu.

2. Mike i jego zespół wiedzą, że użytkownicy prawdopodobnie zmienią później swoje zdanie, a wprowadzanie zmian może się okazać katastrofalne dla kodu, wobec czego zespół zapobiegawczo posługuje się **projektowaniem przyrostowym**, aby wytworzony kod można było z łatwością zmienić na późniejszym etapie.



Zasada



Praktyka

Stosowanie person jest praktyką polegającą na tworzeniu fikcyjnych użytkowników z własnymi imionami (i często też fikcyjnymi zdjęciami portretowymi) w celu lepszego zrozumienia ludzi, którzy będą korzystać z oprogramowania.

3. Ben używa **person** do modelowania typowych użytkowników, ponieważ wie, że im lepiej zespół ich rozumie, tym sprawniej będzie sobie radzić z tworzeniem oprogramowania.



Zasada



Praktyka

Istotna zasada głosi, że działające oprogramowanie jest główną miarą postępu realizacji projektu, jako że jest to najskuteczniejszy sposób na oszacowanie, co właściwie udało się zespołowi osiągnąć.

4. Mike zawsze pilnuje, by jego zespół pracował nad czymś, co będzie mógł później zaprezentować Kate i Benowi, ponieważ wie, że **działające oprogramowanie** świadczy o postępach poczynionych przez zespół.



Zasada



Praktyka

5. Kate chce usprawnić sposób, w jaki zespół wytwarza oprogramowanie, więc zachęca go do **wspólnego doskonalenia się i eksperymentalnego rozwoju**, wymyślając zmiany, jakie może on wprowadzić do swoich procesów, a także używając danych do stwierdzenia, czy zmiany te rzeczywiście okazały się korzystne.



Zasada



Praktyka

Jest to jedna z fundamentalnych praktyk Kanbana. Zespół posługuje się metodą naukową, aby określić, czy jego ulepszenia sprawdzają się w praktyce.

6. Mike i jego zespół **aktywnie ustosunkowują się do zmian**, tworząc kod, który będzie można łatwo zmienić na późniejszym etapie.

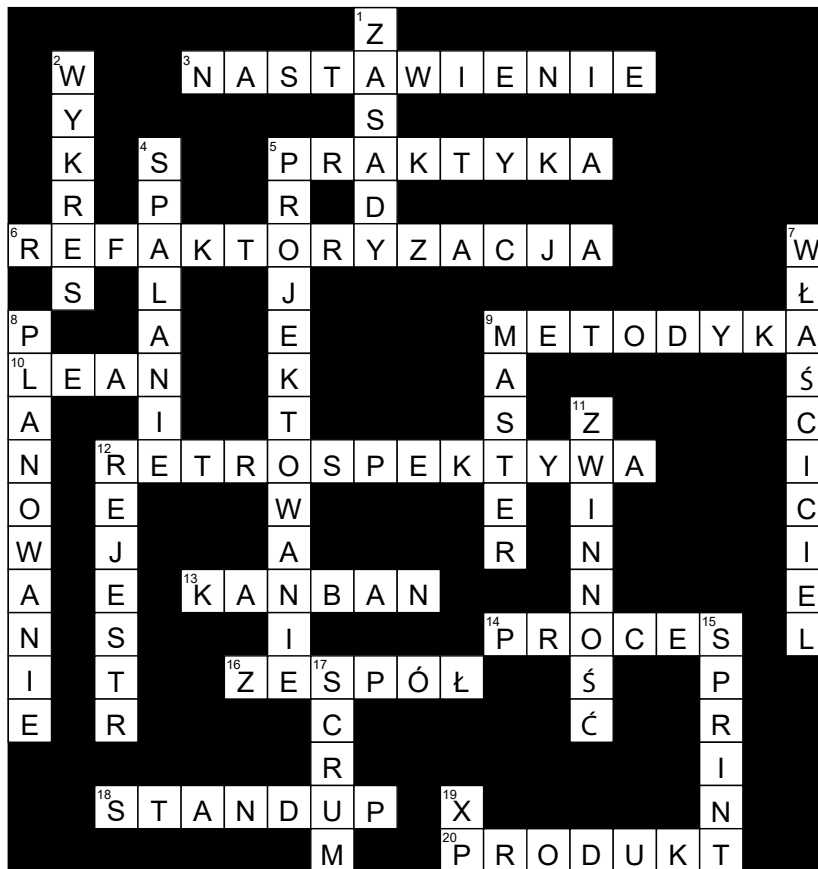


Zasada



Praktyka

Istotną wartością dla wszystkich skutecznych zespołów XP jest aktywne ustosunkowywanie się do zmian, w odróżnieniu od unikania ich lub opierania się im.



o o

HEJ, POPATRZ! W ROZWIĄZANIU „ZAOSTRZ OŁÓWEK” NA POPRZEDNIEJ STRONIE ZNAJDUJĄ SIĘ DOPISKI Z PRZYDATNYMI OBJAŚNIENIAMI! JESTEM PEWIEN, ŻE POMOGĄ CI W NAUCE.

Skorowidz

A

adaptacja, 93, 95, 99
analiza
 Kano, 325
 pre-mortem, 351
 sytuacji, 155
Analizator Odbiorców, 246, 266, 289, 293
anatomia opcji, 270
API, 205
artefakty, 76, 79, 112, 114
automatyzacja kompilacji, 210

B

basen, 315, 317
bezpieczeństwo, 428
bramka zatwierdzania ryzyka, 363

C

certyfikacja PMI-ACP®, 18, 307–376, 391
CFD, cumulative flow diagrams, 292, 344
ciągła integracja, 211, 219, 239, 241, 329
ciągłe doskonalenie, 361
CircuitTrak, 178
codzienne
 Scrumy, 74, 78, 95, 99, 110, 112
 stand-upy, 5, 9, 32, 64
cykl
 kwartalny, 193
 tygodniowy, 193, 212
czas
 idealny, 135
 realizacji, 305
częstotliwość, 257
Członek Zespołu, 6, 62, 64, 193

D

Definicja Ukończenia, 92, 329, 342
demonstracja, 429
diagram Ishikawy, 155, 172, 434, 448
dług techniczny, 226, 227, 323, 331
dokumentacja, 28
dokumenty, 54
doprecyzowywanie, 32
doskonalenie
 procesów, 280
 rejstru, 148
dostarczanie działającego oprogramowania, 44, 46, 54, 131
dostawa oparta na wartości, 322
dostosowanie oszacowań, 133
dostrzeganie marnotrawstwa, 254
droga na skróty, 383
dyskryminacja, 388
działające oprogramowanie, 28, 55, 62, 65
działania operacyjne, 323, 331

E

efektywność zespołu, 82
egzamin PMI-ACP®
 certyfikacyjny, 18, 58, 307–376
 próbny, 391
ekspert, 437
eksperymentowanie, 439, 442
element rejestru, 79
estymacja, 117–177
ESVP, 156, 172
etap
 docierania się, 363, 426
 formowania, 363
 normowania, 363, 435
 realizacji, 363
EVM, earn value management, 326

Skorowidz

F

fiszka, 131

G

GASP, 122, 123

głosowanie

kropkami, 352

na 100 punktów, 352

palcami, 352

gra karciana, 355

gromadzenie danych, 154, 156

gry oparte na współpracy, 352

H

harmonogram dostaw, 447

historyjka o świnie i niosce, 86

historyjki użytkownika, 11, 13, 122–131, 151, 335

I

ilość pracy w toku, 274

informacje

o stanie kompilacji, 241

z projektów i testów, 214

zwrotne o kodzie, 208, 433, 436

inicjatywa, 427

inspekcja, 93, 95, 99

planu, 432

integracja kodu, 205

interfejs programistyczny aplikacji, 205

INVEST, 170

inżynier, 179

IRR, internal rate of return, 326

iteracje, 46, 54, 65, 205

tygodniowe, 183

K

Kanban, 12, 245–306, 432

przepływ pracy, 284

karta sygnału, 288

klasa biznesowa, 381

klasyfikacja rodzajów marnotrawstwa, 260

kod wielokrotnego użytku, 225, 229

kodeks etyki, 377

kolejka, 255

kolejność praktyk, 305

kolokacja, 448

kolorowe kropki, 156

komunikacja, 52, 194

osmotyczna, 217, 218, 445

koncentracja, 83, 88

koniec sprintu, 137, 335

konserwacja kodu, 223

konstruktywne spory, 435

konstytucja zespołu, 342

kontakt z zespołami biznesowymi, 54

kontrola procesu, 110

konwersacja, 131

kopiowanie materiałów chronionych, 388

koszt opóźnienia, 256, 304

kropkowana linia, 171

kumulacyjny diagram przepływu, 292

L

Lean, 12, 245–306

myślenie opcjami, 270

myślenie systemowe, 272

narzędzia myślowe, 253, 256

lider, 24

zespołu, 425

limit

pracowników zespołu scrumowego, 436

WIP, 274, 275, 283

lokalne oddziały PMI, 423

Ł

łapówka, 380, 388

M

maksymalizacja

rezultatów, 321

wartości dodanej, 437

wartości elementów dostawy, 435

Manifest programowania zwinnego, 25, 33, 40, 63, 94

mapa

historyjek, 149, 151, 172, 434

strumienia wartości, 254, 264, 267, 304

marnotrawstwo, 260, 304
 częściowo wykonana praca, 260
 defekty, 261
 dodatkowe funkcje, 260
 dodatkowe procesy, 260
 muda, 268
 mura, 268
 muri, 268
 nadmierne przetwarzanie, 268
 nadprodukcja, 268
 oczekiwanie, 261, 268
 ruch, 261
 wady i błędy jakościowe, 268
 zapasy, 268
 zbędny ruch, 268
 zbędny transport, 268
 żonglowanie zadaniami, 260
 meldowanie, 156
 menedżer projektu, 2, 62, 109
 metoda
 „Fiszka, konwersacja, potwierdzenie”, 172
 empiryczna, 95
 koszulkowa, 151
 MoSCoW, 325, 344
 wartości wypracowanej, EVM, 326
 metodyka zwinna, 1–23
 zasady, 24–71
 minimalizacja zobowiązań, 321
 minimalna wymagana wartość rynkowa, 329
 MMF, Minimally Marketable Features, 255
 model kaskadowy, 35, 67
 MVP, Minimally Viable Product, 255, 323, 329, 331
 myślenie
 opcjami, 276
 systemowe, 272
 szczupłe, 269

N

narzędzia
 do analizy, 157
 do gromadzenia danych, 156
 do podejmowania decyzji, 157

 do przygotowania retrospektywy, 156
 myślowe Lean, 256
 nastawienie zespołu, 9, 45
 negocjacja umów, 31
 niskie ceny, 384
 nowe oprogramowanie, 382
 NPV, net present value, 326, 345

O

obawa przed zmianą, 206
 obliczanie wartości projektu, 326
 odwaga, 83, 88, 184
 ogólnie przyjęte praktyki scrumowe, 122
 ograniczenie czasowe, 46, 67
 opcje, 270
 opracowanie planu, 98
 oprogramowanie wbudowane, 29
 oszacowania, 133, 135
 oś czasu, 156
 otwartość na zmiany, 82, 88, 177, 229

P

persona, 150, 172, 444
 pielęgnacja rejestru, 151, 323, 331, 334
 planistyczny poker, 123, 133, 138, 151, 170, 347, 431
 planowanie, 117–177
 adaptacyjne, 348
 sprintu, 74, 78, 92, 109, 112, 440
 plany kwartalne, 182
 podejmowanie decyzji, 155
 podejście
 do codziennego stand-upu, 8
 iteracyjne, 47
 Lean, 249, 262
 popełnianie błędów, 190, 442
 porządkowanie elementów rejestru, 324
 postępy, 147
 potwierdzenie, 131
 praca infrastrukturalna, 331
 pracownicy zespołów zwinnych, 444
 praktyki, 50
 scrumowe, 117–177

Skorowidz

priorytetyzacja
 elementów rejestru, 149
 relatywna, 328
 wartości, 329
 wartości dla klienta, 329
 wymagań, 325
 z użyciem kropek, 157
problemy, 360, 424, 429
produkt, 334
programowanie
 ekstremalne, XP, 12, 177–243, 333
 iteracyjne, 182
 w parach, 216, 219, 229, 240, 334
 zespołowe, 195
projekt
 kaskadowy, 63
 przyrostowy, 228, 229
 zwinny, 63
prostota, 225, 230
próbna architektura, 215
próby ryzyka, 358
Przegląd Sprintu, 74, 96, 99, 111, 334, 344
przejrzystość, 99
przepływ, 269
 pracy, 284
przerwy, 443
przyrost, 77, 110
przywództwo adaptacyjne, 350
punkty
 historyjek, 126, 132, 135, 138, 151
 rejestru, 144
pytania egzaminacyjne, 58–65, 104–111, 164–173, 234–
 241, 300–321, 330–347, 356–374, 386–446

R

radiator informacji, 199, 239, 363, 429
reagowanie na zmiany, 32, 33
realizacja założonego planu, 32
refaktoryzacja, 11, 13, 209, 227
Rejestr, 47, 54, 67
 porządkowanie elementów, 324
 Produktu, 76, 79, 98, 108, 110, 128, 139

 Sprintu, 79, 90, 108, 139
 uwzględniający ryzyko, 358
relaks, 196
Retrospektywa Sprintu, 11, 13, 74, 78, 86, 101, 109,
 112, 154, 171, 428
 narzędzia, 156
rewizje, 39, 44
rodzaje marnotrawstwa, 261
ROI, return on investment, 326
role scrumowe, 75, 112
 Członek Zespołu, 62
 Scrum Master, 62, 75
 Właściciel Produktu, 62, 75
 Zespół Deweloperski, 75
rozkład
 caves and commons, 193
 przestrzeni pracy, 193
 przestrzeni roboczej, 435
rozmowa o politykach, 293
rozplanowanie
 projektu, 32
 zależności, 134
rozważanie opcji, 256
rozwiązania próbne, 445
rozwiązywanie problemów, 360
rozwój oparty na zestawach, 257
ryzyko, 241, 351, 358, 366, 440
ryzykowne zadanie, 334

S

samoorganizacja, 97, 109, 321, 426
Scrum, 12, 86, 304
 zarządzanie projektami, 71–117
Scrum Master, 62, 75, 160, 428
slack, 183
Snowbird, 24
spadek szybkości, 431, 433
specjalista w zakresie podejść zwinnych, 426
specyfikacja, 124
spotkania z użytkownikami, 52
spotkanie komitetu sterującego, 363

spójność

konceptualna, 256

postrzegana, 256

Sprint, 14, 47, 78, 112

sprzężenia zwrotne, 294, 297, 305

stopniowe doprecyzowywanie, 67

strumień wartości, 264, 267

Sylabus, 309

system

ciągniony, 255, 274, 280, 306

kontroli wersji, 207

szacunek, 82, 88, 184

szczegółowe plany projektów, 134

szczupła produkcja, *Patrz* Lean

szczupłe myślenie, 268

szczupłość, 262

szkielety, 214, 240

szybka kompilacja, 218

szybkość, 135, 138, 144, 151, 170

Ś

środek sprintu, 137

T

tablica

kanbanowa, 281, 286, 305, 424

zadań, 11, 122, 136, 151, 171, 281, 329

TDD, 219, 228, 229

teoria kolejek, 255

tester, 219

testowanie eksploracyjne, 329

testy

jednostkowe, 213, 239

użyteczności, 214, 329, 424

Toyota Production System, 268

tworzenie

jakości i spójności, 250

map myśli, 352

systemu ciągnionego, 274

tygodniowy przegląd ryzyka, 363

U

ukończenie elementu roboczego, 441

unikanie rewizji, 44

upodmiotowienie zespołu, 250

usprawnienie procesu roboczego, 282

ustalenia robocze, 342

ustalenie

nieprzekraczalnego terminu, 306

oszacowania, 133

W

walidacja, 329

wartości metodyk zwinnych, 23

wartość

bieżąca netto, NPV, 326, 328, 345

biznesowa elementów, 447

wczesne dostarczanie wartości dodanej, 426

wewnętrzna stopa zwrotu, IRR, 326, 328

Wideband Delphi, 431

WIP, work in progress, 274

wizualizacja obiegu pracy, 281, 282

Właściciel Produktu, 62, 75, 86, 96, 99, 108, 344

właściwe postępowanie, 378

wodospad, 35

wskazywanie zmian na wczesnym etapie, 43

wskaźnik szybkości, 143

wskaźniki pomiarowe, 257

współpraca, 52, 190

z klientem, 31, 62

wykorzystanie systemu ciągnionego, 280

wykres spalania, 11, 123, 143, 147, 151, 345, 351

wykrywanie

koniecznych zmian, 54

problemów, 360

wymagania, 138

dotyczące bezpieczeństwa, 428

niefunkcjonalne, 334, 445

wytyczne administracyjne, 342

względna priorytetyzacja, 325

wzór historyjki użytkownika, 171

wzrastający wykres spalania, 171

X

XP, Extreme Programming, 12, 177–243, 333

Z

zaangażowanie, 84

 interesariuszy, 336

zadania właściciela produktu, 437

zadanie ukończone, 92

zadowolenie klienta, 41

zakres prac, 147

założenia manifestu, 41

zarządzanie

 obiegiem pracy, 245

 oczekiwaniami interesariuszy, 438

 projektami, 71–117

 przepływem, 268

 wymaganiami, 425

zasady

 Kanbana, 280

 Lean, 250

 Manifestu programowania zwinnego, 46

 metodyk zwinnych, 23, 42, 57

 pracy, 425

 Scruma, 71–117

zaufanie, 190, 447

zdarzenia Scruma, 74

zespoły

 biznesowe, 52

 deweloperskie, 52, 75

 kanbanowe, 294, 427

 sprzężenia zwrotne, 294

 kaskadowe, 53

 Lean, 274, 304

 systemy ciągnięte, 274

 scrumowe, 63, 75, 99, 111

 szczupłe, 253

 testowe, 267

 XP, 177–243

 automatyzacja kompilacji, 210

 ciągła integracja, 211, 239, 241

 cykl tygodniowy, 212

 dług techniczny, 227

 historijki użytkownika, 182, 200

 informacje zwrotne o kodzie, 208

 iteracje tygodniowe, 183

 kod wielokrotnego użytku, 229

 komunikacja, 194, 199

 konserwacja kodu, 223

 otwartość na zmiany, 204, 229

 plany kwartalne, 182

 programowanie w parach, 216, 219, 229, 240

 projektowanie przyrostowe, 228

 próbna architektura, 215

 radiator informacji, 239

 slack, 183

 tempo pracy, 199

 testy jednostkowe, 239

 współpraca, 190

 zarządzanie projektem, 238

 zaufanie, 190

 zgłaszanie uwag, 205

 zwinne, 33, 64, 99, 320, 334

złożoność

 elementów, 432

 projektu, 225

zmiana

 nastawienia, 9, 175

 wymagań, 43, 54, 55, 65

zrównoważony rozwój, 240

zwięzłe tematy, 157

zwinne

 myślenie, 40

 wytwarzanie, 4

zwinność, 1, 18

zwrot z inwestycji, ROI, 326, 328

Ź

źródła

 marnotrawstwa, 268

 ryzyka, 440

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**

Rusz głową! Agile



Metodyki zwinne różnią się od standardowego prowadzenia projektów. Jeśli Twój zespół programistów nigdy nie pracował zgodnie z ideą Agile, najwyższy czas to zmienić, zwłaszcza że zwinność jest teraz modna! Dzięki tej nietuzinkowej, innowacyjnej książce błyskawicznie zrozumiesz, co się kryje pod tą nazwą i jak to się dzieje, że podejścia zwinne nie tylko zapewniają świetne rezultaty, ale też sprawiają, że praca w grupie staje się dużo przyjemniejsza. Jeśli jednak praktyka Agile sprawia trudności w Twoim zespole, być może trzeba zmienić jego nastawienie. Jak to osiągnąć? Tego także nauczysz się dzięki tej książce!

Podobnie jak inne książki z serii „Rusz głową!”, również ta została napisana z wykorzystaniem najnowszych zdobyczy nauki o poznawaniu i uczeniu się. Dzięki temu Twój mózg zaangażuje się w poznawanie nowych koncepcji i ochoczo je przyswoi. Zrozumiesz, na czym polegają i jakie zalety mają najpopularniejsze metodyki Agile: Scrum, XP, Lean i Kanban. Nauczysz się korzystać z dobrodziejstw metodyk zwinnych i sprawisz, że Twój zespół zacznie lepiej planować pracę, będzie pisać lepszy kod — w efekcie poprawi się współpraca w grupie. Niepostrzeżenie idea Agile zwinnie zadomowi się w Twoim umyśle i w Twoim zespole. Co więcej, jeśli zechcesz podjąć wyzwanie i zdobyć certyfikat PMI-ACP®, dzięki tej książce świetnie przygotujesz się do egzaminu!

W książce między innymi:

- założenia i praktyka metodyk zwinnych
- wartości i zasady metodyk zwinnych
- zarządzanie projektami i praktyki w Scrum
- XP, czyli programowanie ekstremalne
- Lean i Kanban, czyli poprawianie obiegu pracy
- przygotowanie do egzaminu i próbny egzamin PMI-ACP®

**Scrum, XP,
Lean i Kanban:
poznaj, zastosuj,
zarządzaj!**

Jennifer Greene i *Andrew Stellman* zajmują się tworzeniem oprogramowania i piszą o tym od 1998 roku. Od czasu założenia agencji Stellman & Greene Consulting w 2003 roku, każdego dnia współpracują z zespołami programistycznymi, pomagając im tworzyć oprogramowanie i dostarczać je użytkownikom. Wśród książek, które wydali, są między innymi: *Agile. Przewodnik po zwinnych metodykach programowania*, *Beautiful Teams*, *C#. Rusz głową!*, *Head First PMP* i *Applied Software Project Management*.

	<p>Sprawdź nasze szkolenia!</p>  <p>AKADEMIA IT & BUSINESS</p> <p>WWW.SZKOLENIA.HELION.PL</p>
<p>helion.pl</p> <p>HELION SA ul. Kościuszki 1c 44-100 Gliwice tel.: 32 230 98 63 helion@helion.pl</p>	
<p>INFORMATYKA W NAJLEPSZYM WYDANIU</p>	

KOD KORZYŚCI
Śięgnij po więcej! ▶



ISBN 978-83-283-4364-1



Cena: 89,00 zł