Henning Wolf

Zwinne projekty w klasycznej organizacji Scrum, Kanban, XP

→ Klasyczna organizacja też może być zwinna!

Helion

Tytuł oryginału: Agile Projekte mit Scrum, XP und Kanban im Unternehmen durchführen

Tłumaczenie: Dawid Zieliński (wstęp, 1 – 6, dodatki), Sławomir Kupisz (rozdz. 7 – 9),
Bartosz Sałbut (rozdz. 10 – 13)

ISBN: 978-83-246-8843-2

Title of the German original: Agile Projekte mit Scrum, XP und Kanban im Unternehmen durchführen
ISBN 978-3-89864-752-6

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/zwipro
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

Spis treści

2 Zwinny projekt to nie bułka z masłem

Holger Koschek

W znaczącym przedsiębiorstwie średniej wielkości dział IT i eksperci dziedzinowi otrzymują szansę na zbudowanie systemu aplikacji na dziewiczym gruncie. Zewnętrzni doradcy do spraw architektury polecają Scruma jako zwinne podejście do zarządzania projektem, a przedsiębiorstwo na to przystaje. Wraz ze wzrostem doświadczenia w Scrumie przedsiębiorstwo stwierdza również, że zwinny projekt to nie bułka z masłem, ale poważny wysiłek — który jednak się opłaca, co obrazują wyniki projektu i motywacja pracowników.

2.1. Pobudka

Na spotkanie inaugurujące projekt przybyli wszyscy: zleceniodawca, analitycy biznesowi, odpowiedni eksperci dziedzinowi oraz wyznaczony Zespół Deweloperski z bezpośrednim przełożonym. Wszyscy w napięciu oczekiwali pomysłów pozwalających osiągnąć ambitny cel, który wielu postrzegało jako niemożliwy do zrealizowania. Rozeszła się również pogłoska, że projekt ma zostać przeprowadzony z wykorzystaniem zwinnych metod zarządzania projektem, a dokładniej z użyciem Scruma. Ale czym był ten Scrum? Jakie korzyści mógłby przynieść? Naszym zadaniem było udzielenie odpowiedzi na te pytania.

Razem z moim współpracownikiem, Jo, mieliśmy w bagażu standardowy zestaw slajdów wprowadzających do Scruma. Jako formę prezentacji wybraliśmy wielokrotnie sprawdzoną zwinną wersję klasyka „dobry glina, zły glina". To zawsze frajda doświadczać reakcji publiczności, gdy Jo wtrąca swoje pierwsze pytanie. Dopiero co zaprezentowałem Rejestr Produktu i wyjaśniłem, jak wymagania funkcjonalne zostaną w nim opisane oraz oszacowane na podstawie wartości biznesowej i innych kryteriów, gdy Jo przerwał: „A gdzie są pojedyncze zadania konieczne do implementacji tych wymagań? Za nimi kryje się przecież wiele różnych szczegółów technicznych. Bez ich znajomości nie jestem w stanie oszacować nakładów potrzebnych do realizacji wymagania funkcjonalnego!". „To dobre pytanie — odpowiedziałem z wielkim spokojem, a niektórzy słuchacze w ostatnim rzędzie zaskoczeni zwrócili uwagę na nasz kiełkujący dyskurs. — Chciałbym odpowiedzieć na to pytanie, czyniąc dwa spostrzeżenia. Po pierwsze: nie szacujemy bynajmniej dokładnego kosztu wymagań funkcjonalnych, ale ich koszt względny (wobec siebie nawzajem). Na tak wczesnym etapie nie jesteśmy w stanie zrobić nic więcej. Po drugie: poszczególne zadania ustala zespół dopiero na początku Sprintu, w którym dane wymagania funkcjonalne powinny zostać zrealizowane". „Ale czy to nie jest o wiele za późno? Jak rozsądnie planować projekt na podstawie tak skromnych informacji?" — ekscytował się Jo. Następnie zarysowałem temat planowania zwinnego projektu i kierowania nim. To rozbudziło ostatnich śpiochów. Po raz kolejny mieliśmy wrażenie, że Jo trafił w czuły punkt sceptyków.

Przyszedł czas na przekonanie ich — nie tu i teraz, ale w toku projektu. Dlatego zaprosiliśmy wszystkich do przeżycia zwinnego zarządzania projektem na żywo w naszym projekcie. Było ku temu wiele sposobności: Przegląd Sprintu oraz (po konsultacjach z zespołem) Codzienny Młyn stały otworem dla każdego zainteresowanego, a przejrzysta kontrola projektu powinna była całkowicie ujawniać rzeczywisty stan projektu — widoczny dla każdego w dowolnym momencie. To była nowość — nie tylko w tym przedsiębiorstwie. Wiedzieliśmy już z innych treningów zwinności, że przejrzystość i uczciwość przysporzą nam nowych przeciwników. Dowiedzieliśmy się jednak również, że u większości z nich wygrywa ciekawość i zanim zechcą puścić projekt z dymem (do czego tak naprawdę nigdy nie doszło), najpierw go poobserwują.

Nasza zwinna pobudka zadziałała po raz kolejny. Wszyscy byli zgodni co do tego, że ten projekt będzie inny. Chcieliśmy obudzić pozytywną ciekawość i złagodzić lęki. Największe obawy miał Zespół Deweloperski — to było dla nas zrozumiałe. Jak mogliśmy przypuszczać, tej grupie nasze zajęcia wprowadzające dostarczyły więcej pytań niż odpowiedzi. Dlatego umówiliśmy się z zespołem na następny dzień, aby przyjrzeć się Scrumowi z ich perspektywy i wspólnie opracować rozkład jazdy na wystartowanie projektu.

Czy Twoje wprowadzenie do Scruma powinno zostawić długotrwałe wrażenie? Jeśli tak, rozpraw się jeszcze z klasycznymi uprzedzeniami wobec zwinnych metod pracy i sposobów postępowania. W ten sposób, mówiąc żartobliwie, pozbawisz wiatru żagle sceptyków.

2.2. Zespół się formuje

Spotkaliśmy się w małym kręgu. Celowo nieformalny nastrój sprzyjał budowaniu atmosfery, w której mogliśmy w pełni poświęcić się odpowiadaniu na pytania o Scruma. Aby powoli zabrać się za trudne tematy, jak na przykład odpowiedzialność zbiorowa, zaczęliśmy od opisu idealnej przestrzeni projektowej — i natychmiast zostaliśmy skonfrontowani z pierwszym problemem. Programiści pochodzili z wielu różnych działów, a każdy dział posiadał własne pomieszczenia. Zespół Deweloperski był więc rozproszony po różnych lokalizacjach. Teoretycznie wszyscy działali w tym samym budynku, ale to nam nie wystarczało. Co robić? Zleceniodawca zrozumiał nasz problem i obiecał, że wyruszy na poszukiwanie odpowiedniej przestrzeni dla projektu. Zespół Deweloperski skorzystał z okazji i dał mu jeszcze jedno zadanie na drogę: zleceniodawca powinien dopilnować, aby dotychczasowe stanowiska pracy zostały na czas trwania projektu zarezerwowane. Podejrzewaliśmy, że stoi za tym strach przed utratą ukochanego biurka i „współlokatorów" oraz otrzymaniem przydziału na gorsze stanowisko po zakończeniu projektu. Gdy ostrożnie wspomnieliśmy o tym jednemu z członków zespołu, dowiedzieliśmy się o jeszcze jednej przyczynie: jego przełożony oddelegował go do tego projektu jedynie na 70 procent czasu pracy i oczekiwał, że pozostałe 30 procent spędzi on na wykonywaniu innych zadań, w dotychczasowej lokalizacji. Gdy wypytaliśmy o to pozostałych deweloperów, okazało się, że poza jednym wyjątkiem wszyscy zostali skierowani do pracy w projekcie jedynie w części etatu. Biorąc pod uwagę strategiczne znaczenie i napięty harmonogram tego projektu, była to nietypowa decyzja, którą mimo wszystko tymczasowo zaakceptowaliśmy. Chcieliśmy w końcu osiągnąć poczucie bezpieczeństwa, a nie wywoływać dalszy niepokój. Dlatego kontynuowaliśmy opis przestrzeni zespołowej. Objaśniliśmy rolę tablicy z zadaniami, na której

uwidaczniano (dla wszystkich zainteresowanych) prace wykonywane w bieżącym Sprincie i odnotowywano postępy. Przedstawiliśmy ponownie Codzienny Młyn, podczas którego zespół powinien się codziennie synchronizować, i podkreśliliśmy ogromne znaczenie komunikacji w zespole.

Zespół zasługiwał na początku swojej zwinnej podróży na szczególną uwagę. Certyfikowany kurs na Mistrza Młyna ujawnił kolejne pytania — ponieważ kurs jest dopasowany do Mistrza Młyna, jedynie pośrednio rozpatruje interesy samego Zespołu Deweloperskiego. Postaw się w sytuacji debiutującego w Scrumie zespołu, a stwierdzisz, że zwinne wartości, działanie na własną odpowiedzialność, czynności związane z planowaniem i umiejętności miękkie niezbędne do skutecznej pracy zespołowej są na początku raczej obciążeniem niż radością. Na początkowym etapie zespół nie wykształcił jeszcze zdolności rzemieślniczych, których oczekuje się od zwinnych deweloperów.

Prawie wszyscy członkowie zespołu znali się już wcześniej i zdawali się dobrze wzajemnie rozumieć. Cieszyli się na czekające ich zadania i pasjonujące nowe technologie, które chcieliśmy wypróbować i wykorzystać w tym projekcie. Gdy niezobowiązująco dyskutowaliśmy na temat tych technologii i możliwych architektur oprogramowania, dokonaliśmy zdumiewającego odkrycia: członkowie zespołu oczekiwali od nas — swoich doradców — gotowych szkiców architektonicznych oraz zaleceń odnośnie do technologii, które miały zostać wykorzystane w projekcie. Nieco zaskoczeni, rozejrzeliśmy się dokoła. Ci, którzy wyrazili to pragnienie, nie byli wcale świeżo upieczonymi absolwentami uniwersytetu, lecz doświadczonymi inżynierami oprogramowania z wieloletnim doświadczeniem. „Jak można się dobrowolnie wykluczyć z tak interesującej dyskusji na tematy techniczne?" — pomyślałem. Nie musiałem zadawać tego pytania na głos, by otrzymać odpowiedź. Właściwie były to dwie odpowiedzi. „Powszechną tutaj praktyką jest instruowanie deweloperów przez architekta, który podejmuje techniczne decyzje projektowe" — powiedział jeden z deweloperów, a drugi dodał: „A poza tym zostaliście zapowiedziani jako architekci oprogramowania dla naszego projektu. Czy coś się nie zgadza?". Ależ owszem, wszystko się zgadzało. Natomiast pomysł poprowadzenia projektu zgodnie z zasadami Scruma powstał, ściśle mówiąc, podczas pierwszej dyskusji o architekturze. To wtedy ustaliliśmy, że mamy dwie możliwości: albo zagubimy się w teoretycznych dyskusjach, dopełnianych przez nieskończone ewaluacje oprogramowania i technologii, albo zabierzemy się ostro do pracy, aby w krótkich iteracjach osiągnąć pierwsze rezultaty. Postawiliśmy więc wyznaczonego kierownika projektu przed wyborem pomiędzy podejściem bardziej klasycznym a podejściem zwinnym, mimo że nie wpisywało się to w nasze pierwotne zamówienie na konsultacje.

2.3. Właściwe zlecenie

Zacznijmy jednak od samego początku: naszą wizytę rzeczywiście rozpoczęliśmy jako architekci oprogramowania. Wyposażeni w niezbędną wiedzę technologiczną i architektoniczną oraz stosowną porcję doświadczenia, mieliśmy stworzyć wspólnie z klientem nową platformę oprogramowania, przeznaczoną do realizacji kluczowych procesów przedsiębiorstwa, korzystając przy tym z możliwości pracy na dziewiczym gruncie. Czy nie jest to marzenie każdego doradcy? Niemal nie zważając na istniejące aplikacje i infrastrukturę, mieliśmy opracować świeże pomysły na nadchodzące lata oraz zaimplementować je, wykorzystując nowoczesne technologie.

Rozmowy z architektami oprogramowania klienta były interesujące i zazwyczaj spełniały swój cel. Mimo to mieliśmy poczucie, że nie jesteśmy dostatecznie szybcy. Wyraźną oznaką nadmiaru teoretyzowania była relacja czasu poświęconego projektowaniu do czasu spędzanego na kodowaniu: po dwóch tygodniach badań nie powstała ani jedna linia kodu oparta na dokumentacji. Znajdowaliśmy się w samym środku drugiej fazy modelu kaskadowego — projektowania. Na domiar złego nie mieliśmy zielonego pojęcia o tym, co powinno zostać wykonane w celu funkcjonalnego rozwinięcia platformy. Specyfikacja funkcjonalna projektu (model kaskadowy, faza pierwsza: wymagania) była bowiem wynikiem pracy innego działu, którego żadnego przedstawiciela dotychczas nie poznaliśmy i od którego nie otrzymaliśmy żadnego dokumentu.

Dzień, w którym poznaliśmy ekspertów dziedzinowych i porozmawialiśmy o wymaganiach na podstawie specyfikacji funkcjonalnej projektu, wyznaczył pozytywny punkt zwrotny. Dowiedzieliśmy się wreszcie, czemu miało służyć zbudowanie nowej platformy technicznej. Technologia nie była dla nas celem samym w sobie. Właśnie dlatego ostatecznie ucieszyliśmy się z przekazania wizji produktu, która zadziałała na naszą nieśmiało kiełkującą architekturę jak zastrzyk energii. Na podstawie przypadków użycia o wiele łatwiej było nam oceniać i wybierać technologiczne alternatywy. Realizacja pewnych istotnych wymagań funkcjonalnych nieuchronnie się opóźniała, a wymagania niefunkcjonalne, takie jak bezpieczeństwo i wydajność, mogły zostać zdefiniowane dokładniej. Niemniej jednak byliśmy wciąż w drodze. Wprawdzie mogliśmy opisać swoje pomysły ekspertom dziedzinowym, ale nie mogliśmy ich pokazać. Im bardziej abstrakcyjny był pomysł, tym trudniej było zachwycić ekspertów. A kto nie odczuwa wobec danego rozwiązania zachwytu, ten nie opowiada się za nim bezdyskusyjnie. Chcieliśmy wytworzyć coś, co nadawało się do prezentacji, podczas której można by zetrzeć się merytorycznie, wywołując dyskusję lub burzę pomysłów. Zamiast tego mieliśmy jedynie stertę papieru. To powinno było, a nawet musiało, się zmienić.

Zamiast długo analizować, planować i projektować, zespół projektowy powinien raczej zabrać się do pracy, gdy tylko uzyska pierwsze wiarygodne wyobrażenie na temat oczekującego zadania. Wizja produktu ma wtedy stanowić przydatną wskazówkę. Opisuje ona „nadrzędny cel", jednakże bez wskazywania gotowej drogi i bez narzucania zbyt dużych ograniczeń. W końcu świat wokół nas zmienia się z dnia na dzień. Pomysły i wymagania pojawiają się i znikają, są modyfikowane, odrzucane i ponownie przyjmowane. Wszystko to ma wpływ na kształt końcowego produktu. Istota tego produktu, wyrażona w wizji, przetrwa jednak z reguły dynamikę codzienności projektu. Im bardziej szczegółowy i daleko idący jest plan projektu, tym więcej wymagań trzeba później dostosować do nowych realiów projektu. Początkowe planowanie było więc ostatecznie chybioną inwestycją. Pieniądze lepiej należało zainwestować w jak najszybsze pozyskanie informacji zwrotnej, otrzymanej w wyniku bezzwłocznego zabrania się do pracy.

2.4. Od partyzantki do zwinności

Aby szybko osiągnąć rzeczywiste wyniki i zminimalizować ryzyko popłynięcia w złym kierunku, Jo i ja zadecydowaliśmy, że przekonamy klientów do wyższości zwinnego sposobu postępowania. W tym celu zabraliśmy się do pracy bardzo ostrożnie. Zaczęliśmy od czegoś nieszkodliwego, co osobie nieobeznanej z warsztatem zwinnych narzędzi nie od razu rzuca się w oczy: przygotowaliśmy Rejestr Produktu i wypełniliśmy go wymaganiami funkcjonalnymi i technicznymi,

zapisanymi w postaci Historyjek Użytkownika. Od razu dostarczyliśmy również uzasadnienie wprowadzenia Rejestru Produktu. Zwróciliśmy w nim uwagę na dużą liczbę pomysłów, które wykreowaliśmy w ubiegłych tygodniach. Aby nie stracić tych pomysłów i jednocześnie nie musieć od razu ich wszystkich dokładnie wyceniać, powinniśmy najpierw dodać je do Rejestru Produktu i jedynie zgrubnie oszacować. Naszym odpowiednikiem po stronie klienta był klasyczny kierownik projektu. Wprawdzie rozumiał podstawy użycia Rejestru Produktu, ale skonfrontowany z szacowaniem względnych wielkości zadań, w przeciwieństwie do konkretnych wartości liczonych w roboczodniach dewelopera, stawał się coraz bardziej niepewny. Być może nie wyjaśniliśmy mu dostatecznie zrozumiale przewagi względnego szacowania i różnicy pomiędzy oszacowaniem a dokładnym planowaniem nakładu prac. W każdym razie w odwzorowaniu ukonstytuowanym przez nasz Rejestr Produktu brakowało mu dwóch informacji istotnych przy planowaniu projektu: konkretnych zadań i nakładów koniecznych do ich realizacji. Nasza wskazówka, że są one dostarczane przez zespół, była uderzeniem głową w mur klasycznego zarządzania projektem: speszony kierownik projektu zapytał nas, jak w takim razie ma planować projekt bez znajomości tak istotnych parametrów. Ponadto zespół nie został jeszcze skompletowany. Deweloperzy nie byli tak czy owak przyzwyczajeni do samodzielnego definiowania swoich zadań — nie powinni zresztą tego w ogóle wykonywać, ponieważ jest to obowiązek kierowników projektów i głównych architektów. Nasze anteny, ustawione na wykrywanie kompetencji miękkich, odnotowały mieszankę braku zrozumienia, strachu przed nowością i bliżej nieokreślonego egzystencjalnego lęku, wywołanego poprzez podanie w wątpliwość roli zarządzania projektem. Nadszedł czas na zwolnienie obrotów.

Nie tylko zespół jest początkowo sceptycznie nastawiony do niekonwencjonalnych wartości, zasad i praktyk. Również świeżo upieczony Mistrz Młyna potrzebuje czasu na odnalezienie się w swojej roli. Rola ta jest często obsadzana kierownikiem projektu lub głównym architektem. Nie jest to szkodliwe, ale nie zawsze bywa pomocne. Dla wielu kierowników projektów zamiana ról wiąże się z natychmiastową utratą władzy. Zamiast kierować pewną grupą deweloperów, stają się „jedynie" arbitrami samoorganizującego się zespołu, dbającymi o to, by zespół mógł pracować bez żadnych zakłóceń. Należy o tym pamiętać, jeśli zamierzamy pomóc niedoświadczonemu Mistrzowi Młyna w osiągnięciu sukcesu w jego nowej funkcji.

2.5. Kompleks Scruma

Poszliśmy na ustępstwa. Rozszerzyliśmy Rejestr Produktu o jedną kolumnę, w której dołożyliśmy do Historyjek Użytkownika pierwsze prace do zrealizowania. Razem z kierownikiem projektu spotkaliśmy się ze współpracownikami, którzy sporządzili koncept projektu, aby przejrzeć ten dokument w poszukiwaniu materii, z której mogłyby powstać kolejne Historyjki Użytkownika. W ten sposób wykreowaliśmy nowy problem, ponieważ Rejestr Produktu zawierał zarówno funkcjonalne, jak i techniczne Historyjki Użytkownika. Jak można było je rozróżnić? Dlaczego w ogóle powstały techniczne Historyjki Użytkownika? Czy nie sprzeciwia się to zwinnej ideologii, według której liczy się jedynie funkcjonalność? Pytania zadane przez jednego ze współpracowników możemy obalić stwierdzeniem, że otrzymaliśmy dwa zupełnie różne zlecenia: po pierwsze, powinna powstać techniczna platforma do zarządzania procesami biznesowymi; po drugie, na tej platformie powinien zostać osadzony wybrany proces biznesowy

— przykładowy, lecz produktywny. W rezultacie mieliśmy do czynienia z „wymaganiem technicznym" (platforma) i „wymaganiem funkcjonalnym" (proces biznesowy). Dlatego w celu rozróżnienia pomiędzy obydwoma projektami zafundowaliśmy Rejestrowi Produktu kolejną kolumnę. Nasz pierwotny cel, czyli zorganizowanie Rejestru Produktu w najprostszy możliwy sposób, aby pozbawić scrumowych nowicjuszy obaw, nie został przez nas osiągnięty. Fakt, że to nie Scrum był temu winny, lecz złożona natura projektu, wcale nie pomagał nam w dalszej pracy. Nadal byliśmy zdania, że z takimi skomplikowanymi projektami można sobie poradzić jedynie, stosując podejście zwinne.

Coraz wyraźniej uwidaczniał się brak możliwości dalszej pracy bez uprzedniego przekazania podstaw Scruma wszystkim osobom, które były zaangażowane w projekt. Tak czy owak, nadszedł czas na ustalenie składu zespołu i zacieśnienie kontaktu z ekspertami dziedzinowymi. Szczęśliwym trafem zleceniodawca zwołał trwające pół dnia spotkanie inauguracyjne. Wyciągnęliśmy z szuflady nasz standardowy zestaw folii wprowadzających do Scruma i cieszyliśmy się na ponowne przedstawienie naszego klasyka „dobry glina, zły glina", który pojawił się już na początku tej historii. A jak potoczyło się to dalej?

Rzadko można dowolnie wybrać swój pierwszy projekt realizowany zgodnie z wytycznymi Scruma. Jeśli rzeczywiście będziesz mieć wybór, zdecyduj się na przejrzysty projekt o strategicznym znaczeniu. Złożony projekt, taki jak opisany w tej książce, niepotrzebnie podniesie próg wejścia do świata zwinności. Z drugiej strony, bardzo łatwy projekt-zabawka jest pozbawiony znaczenia. Nawet jeśli później zostanie oceniony jako sukces, krytycy nadal będą utrzymywać, że podejście zwinne zadziałało jedynie dlatego, iż projekt był tak mały i przewidywalny. W najgorszym przypadku komunikat o sukcesie zostanie zakłócony przez wieści docierające z dużych, krytycznych projektów.

2.6. Zwycięstwa i porażki

Jak już wspomniano, zespół domagał się zarówno zachowania dotychczasowych stanowisk pracy, jak i nowych pomieszczeń dla zespołu. Zleceniodawca rzeczywiście spełnił oba życzenia. Byliśmy dumni, gdy pokazywał nam nową przestrzeń biurową, która znajdowała się w przyległym budynku — właśnie zdobyliśmy pierwsze punkty dla zwinności. Fizyczne oddzielenie od reszty przedsiębiorstwa postrzegaliśmy jako zaletę. Zrządzeniem losu „mieszkaliśmy" teraz w bezpośrednim sąsiedztwie ekspertów dziedzinowych, dla których powinniśmy rozwijać pierwszą aplikację na naszej nowej platformie. Było to praktyczne, ponieważ krótsza droga powinna — zgodnie z naszym planem — zacieśnić współpracę. Uzupełniając: pokoje nie były tak nowoczesne jak w centrali firmy, ale mieliśmy dostatecznie dużo miejsca i potrzebny spokój. Co prawda nasze małe imperium składało się z dwóch pomieszczeń (oraz sali konferencyjnej), ale kierownik projektu natychmiast je podzielił: jeden pokój przypisał sobie wraz z analitykami biznesowymi, a drugi przypadł w udziale nam, deweloperom i architektom. „To nic strasznego" — pomyśleliśmy i rozpoczęliśmy urządzanie naszego nowego biura w duchu zwinności: zbudowaliśmy ścianę dedykowaną metodzie metaplanu[1] dla tablicy z zadaniami oraz przygotowaliśmy ją do pierwszego

[1] Metaplan — metoda dyskusji, podczas której uczestnicy wspólnie tworzą plakat obrazujący jej graficzny skrót — *przyp. tłum.*

Sprintu. Zespół i kierownik projektu z zaciekawieniem obserwowali nasze wybryki. Objaśniliśmy im funkcjonowanie tablicy z zadaniami, powtórzyliśmy raz jeszcze opis wewnętrznej struktury Sprintu i zaprosiliśmy wszystkich do sali konferencyjnej na spotkanie planujące pracę w Sprincie.

2.7. Planowanie Sprintu (część I)

Podczas Planowania Sprintu ostatecznie zemściło się to, że zespół nie brał udziału w sporządzaniu Rejestru Produktu. Elementy Rejestru Produktu zostały już wcześniej spriorytetyzowane, przy czym uwzględniono zarówno funkcjonalne, jak i techniczne zależności. Mogliśmy zatem bez przeszkód włączyć się do wspólnego szacowania wielkości poszczególnych elementów. Kierownik projektu, Jo i ja byliśmy w pełni zaznajomieni z tematem — nic dziwnego, skoro razem zbudowaliśmy Rejestr Produktu. Pięciu pozostałym członkom zespołu musieliśmy wyjaśnić historie stojące za konkretnymi elementami. Jeden z członków zespołu został do niego ściągnięty jako wewnętrzny architekt i powinien być naszym partnerem sparingowym przy opracowywaniu architektury systemu. Zadawał wiele dobrych pytań, chciał zrozumieć szczególnie decyzje podjęte w kontekście podstawowej architektury i wniósł sporo świeżych pomysłów, które stawiały wiele elementów Rejestru Produktu w zupełnie nowym świetle. Niestety, jego żądza wiedzy przedłużała Planowanie Sprintu, a poza tym odczuwaliśmy, że pozostali członkowie zespołu rzadko chcieli lub mogli podążać za naszymi dyskusjami (co odpowiadało ich zapotrzebowaniu na posiadanie architekta w zespole). Aby całkowicie nie zgubić zespołu, ograniczyliśmy się do wyjaśnienia szczegółów funkcjonalnych. Samo w sobie było to wystarczająco wymagające, ponieważ niektórzy współpracownicy nie byli przyzwyczajeni do zastanawiania się nad zakresem funkcjonalnym. Dotąd otrzymywali jasne techniczne zadania i w razie wątpliwości zwracali się do swoich architektów oprogramowania lub kierowników projektu. Teraz mieli nagle podejmować decyzje dotyczące funkcjonalności albo przynajmniej mieć w tym udział, a na dodatek szacować wielkość elementów Rejestru Produktu, zarówno technicznych, jak i funkcjonalnych. „Wielkość? Dlaczego nie pracochłonność?" — zapytał nas jeden z deweloperów. Ponownie objaśniliśmy względne szacowanie zadań i przeszliśmy do gry w Planowanie Pokerowe[2].

W momencie, gdy rozdawaliśmy karty, wzbudziliśmy pośród sceptyków prawdziwe zainteresowanie: gry karciane stanowiły urozmaicenie szarej codzienności projektu. Przytoczyliśmy reguły gry i od razu przeszliśmy do rzeczy. Szybko znaleźliśmy punkt odniesienia pośród elementów Rejestru Produktu. Otrzymał on wartość równą 2. W związku z tym kolejny wybrany element Rejestru Produktu powinien zostać oszacowany względem wyłonionego właśnie punktu odniesienia. Chociaż wszyscy zrozumieli zasady gry, nadal trudno było zdecydować się na konkretną wartość wielkości zadania. Kto wie, jak duże mogły być pozostałe elementy Rejestru Produktu. Na podstawie wątpliwości niektórych członków zespołu ustaliliśmy również, że nie zrozumieli oni jeszcze w pełni elementów Rejestru Produktu. Pierwsza tura Planowania Pokerowego wydobyła na światło dzienne wiele różnych wartości szacunkowych. Gdy zachęciliśmy graczy z najwyższą i najniższą wartością do uzasadnienia swojego oszacowania, otrzymaliśmy odpowiedzi, które z trudem można było odróżnić. Kartę z najwyższą

[2] *Planning Poker* (Planowanie Pokerowe) to znak towarowy Mountain Goat Software, Inc.

wartością wyłożył architekt. Wyjaśnił swój wybór, wskazując na techniczne koszty produkcji i realizację pozostałych niefunkcjonalnych wymagań dotyczących bezpieczeństwa, skalowalności, rozszerzalności i możliwości ponownego wykorzystania.

Projekt techniczny, który nakreślił dla realizacji tego elementu Rejestru Produktu, byłby wspaniałym uzupełnieniem każdego podręcznika o projektowaniu zorientowanym obiektowo. Dla naszego konkretnego problemu projekt ów był jednakże zbyt skomplikowany. „Czy słyszałeś o regule KISS?" — zapytałem. Pokręcił głową. „*Keep it simple, stupid*[3]; oznacza to tyle co: utrzymaj to tak prostym, jak to tylko możliwe — wyjaśniłem akronim. — Nasz plan działania powinien być jednocześnie tak prosty, jak to możliwe, i zarazem tak złożony, jak to potrzebne, ale zdecydowanie nie bardziej skomplikowany". „Ale co, jeśli wymagania się zmieniają i oczekuje się nowych funkcji, których prosty szkic projektu nie jest w stanie spełnić?" — odparł architekt. „Wtedy rozszerzamy nasz prosty system, przy czym refaktoryzujemy projekt oraz kod. Dzięki testom jednostkowym znajdujemy się w korzystnym położeniu, pozwalającym na sprawdzenie, czy zrefaktoryzowany system zachowuje się tak samo jak przed refaktoryzacją. Testy minimalizują ryzyko niesione przez refaktoryzację". Pozornie przyjął tę argumentację — ale przypuszczalnie tylko dlatego, że nalegał na to kierownik projektu. Wiedziałem teraz, że powinienem mieć architekta na oku. W rzeczywistości miał on tendencję do konstruowania bardzo złożonych, generycznych szkiców projektowych, przygotowanych na wszelkie ewentualności — nawet jeśli były one jedynie domniemane, a nie wskazane — przez co czynił kod niepotrzebnie rozdmuchanym i trudniejszym do zrozumienia.

Gra w Planowanie Pokerowe przyniosła całemu zespołowi wiele przyjemności. Tej atmosferze zabawy dali się ponieść również co bardziej nieśmiali koledzy i odważyli się zadawać własne pytania. Najbardziej podobał się wszystkim fakt, że szacowanie zostało przeprowadzone wspólnie. W przeszłości pracochłonność zadań musieli szacować indywidualnie. Ryzyko błędu oszacowania było więc wyższe, a na dodatek, w najgorszym przypadku, można było zostać osobiście pociągniętym do odpowiedzialności. Mając za plecami zespół, wszyscy byli odważniejsi i pełni wiary. I w ten oto sposób jeszcze przed południem dokonaliśmy wyboru elementów Rejestru Produktu, które zamierzaliśmy zrealizować w pierwszym Sprincie. Elementy były dobrze dopasowane do siebie pod kątem funkcjonalnym, dzięki czemu byliśmy w stanie szybko sformułować Cel Sprintu.

2.8. Planowanie Sprintu (część II)

Po południu doszliśmy do punktu, na który czekał kierownik projektu. Definiowaliśmy prace (zadania) dla wszystkich elementów Rejestru Produktu, które zostały wybrane do danego Sprintu. Kierownik projektu zaskoczył nas wypowiedzią, że ma już w zanadrzu gotowe odpowiednie zadania — w dodatku dla wszystkich elementów Rejestru Produktu. Zdumieni, pozostawiliśmy mu pole do popisu i zostaliśmy nagrodzeni generyczną mapą zadań, która ubierała pojedyncze elementy Rejestru Produktu w płaszcz miniaturowego modelu kaskadowego: „analiza", „projektowanie architektury", „implementacja", „testy integracyjne", „testy funkcjonalne" — tak (albo przynajmniej analogicznie) brzmiały zadania. Nie podobało się to ani nam, ani (na szczęście) architektom oprogramowania obecnym w zespole. Kierownik zespołu przygotował

[3] Na język polski zazwyczaj bywa to tłumaczone jako „nie komplikuj, głupku" — *przyp. tłum.*

bowiem w rzeczywistości szczegółowe pomysły na implementację poszczególnych elementów. Naturalnie musieliśmy go co rusz powstrzymywać przed przeholowaniem z regułą KISS. Jo i ja dołożyliśmy do tego pomysły ze wstępnej dyskusji na temat architektury i w ten sposób z pomocą architekta zdefiniowaliśmy zadania do wyznaczonych elementów Rejestru Produktu, podczas gdy pozostała część zespołu (w tym kierownik projektu) przyglądała się naszym zabiegom i próbowała naśladować nasz tok myślenia. Być może dobrym pomysłem byłoby rozbudzenie w pozostałych członkach zespołu aktywności wobec biegu wydarzeń, ale czas nie był naszym sprzymierzeńcem. Zdecydowaliśmy się na punktualne rozpoczęcie Sprintu kosztem wyrównania poziomu wiedzy w zespole, w nadziei na pomyślne przekazanie informacji w trakcie samego Sprintu. Zobowiązanie się zespołu do osiągnięcia Celu Sprintu w zasadzie odeszło wieczorem w zapomnienie, ale Jo i ja pominęliśmy to wspaniałomyślnym milczeniem.

Tuż przed zasłużonym fajrantem ogłosiliśmy zespołowi, że od jutra „…będziemy zajmować się właściwą pracą deweloperską!". Współpracownicy bardzo się na to ucieszyli. W końcu będą mogli zająć się tym, co sprawia im największą frajdę. Bez zastrzeżeń przyjęli naszą chęć złożenia im porannej wizyty podczas Codziennego Młyna.

Być może zadałeś już sobie pytanie, dlaczego w tej opowieści nie pojawił się Właściciel Produktu. Odpowiedź jest zarazem tak prosta, jak i niezadowalająca: ta rola nie została optymalnie obsadzona. Niektórzy odnieśli wrażenie, że do dwóch typów zadań (zbudowania platformy technicznej oraz osadzenia w niej pierwszej funkcjonalności realizującej proces biznesowy) potrzebowaliśmy dwóch Właścicieli Produktu. Rolę prekursora technicznego objął tradycyjnie architekt oprogramowania, brakowało mu jednak wiedzy dotyczącej dostosowania platformy do wymagań funkcjonalnych. Eksperci dziedzinowi byli zajęci definiowaniem pierwszego nowego procesu biznesowego. W tym celu przyglądali się obecnemu procesowi, standaryzując i optymalizując tenże proces oraz dostosowując go do postaci umożliwiającej wsparcie techniczne. Pomoc okazana Zespołowi Deweloperskiemu przez ekspertów dziedzinowych była wzorowa: zawsze znajdowaliśmy partnera do rozmowy, który był w stanie natychmiast odpowiedzieć na nasze pytania. Natomiast decyzje techniczne były często odwlekane w czasie, ponieważ konieczne było oczekiwanie na opinie trudno dostępnych specjalistów. Po części było to osadzone w kulturze pracy obowiązującej w przedsiębiorstwie — ciężar podjęcia decyzji był regularnie przekazywany „wyżej". Patrząc z perspektywy czasu, byłoby lepiej, gdyby równoległe badania w zakresie domen — funkcjonalnej i technicznej — odbywały się na przedpolu projektu. W tej konkretnej sytuacji zespół projektujący proces czekał wciąż na Zespół Deweloperski i odwrotnie.

> Wielu deweloperów musi na początku swojego pierwszego zwinnego projektu oprzeć się chęci rozwijania najbardziej ogólnego ze wszystkich możliwych rozwiązań. Będą oni wciąż wysłuchiwać argumentów dotyczących zgodności ze sztuką: „Jeśli teraz nie przewidzimy elastyczności rozwiązania, nie będziemy go w stanie rozbudować". Twoją ripostą mogłoby być: „Czy jesteś pewien, że istotnie potrzebujemy tej elastyczności?". Jeden z moich profesorów mawiał zawsze: „Użycie poprzedza ponowne użycie". Jakie to prawdziwe…

2.9. Codzienny Młyn

Dopiero kwadrans po dziesiątej wszyscy deweloperzy zgromadzili się w biurze i mogłem wreszcie wezwać ich na Codzienny Młyn. Na pytanie, dlaczego nie została dotrzymana ustalona na spotkanie godzina dziesiąta (co z mojego punktu widzenia i tak było dosyć późną porą),

otrzymałem różnorakie niezadowalające odpowiedzi. Zobowiązania w innych projektach, zbyt powolna podróż koleją miejską albo po prostu brak pamięci o terminie spotkania. Nie mogłem zrobić nic więcej poza przypomnieniem wszystkim o znaczeniu tego spotkania. Następnie przytoczyłem reguły gry i oddałem głos pierwszemu współpracownikowi. Stał on niepewnie przed tablicą, przeczytał ponownie wszystkie elementy Rejestru Produktu, które wybraliśmy do tego Sprintu, poświęcił się bez reszty kartkom z zadaniami i w końcu obwieścił: „Nie wiem, którego zadania powinienem się podjąć". Zanim niepokój zdążył się rozprzestrzenić, przejąłem moderację, przekierowałem uwagę wszystkich na pierwszy element Rejestru Sprintu i zaproponowałem współpracownikowi jedno z zadań, które uznałem za odpowiednie. „I co dokładnie powinienem zrobić w ramach tego zadania?" — zapytał. Odpowiedź na to pytanie przyszła z ust innego współpracownika. Bez namysłu połączyłem pytającego z odpowiadającym — trudno wprowadzić w zespole programowanie w parach w bardziej elegancki sposób. Obaj byli zadowoleni z możliwości wspólnej pracy nad tym zadaniem.

Przykład znalazł naśladowców i w ten oto sposób po zakończeniu Codziennego Młyna mieliśmy dwie programistyczne pary oraz troje „samotnych jeźdźców". Jedynie kierownik projektu był niezadowolony, ponieważ nie odegrał w tym kręgu żadnej roli. Nie chciał pogodzić się z zadaniem usuwania wszelkich przeszkód stających na drodze zespołu. Dużo bardziej na rękę było mu kierowanie projektem i zespołem oraz wpływanie na kształt architektury. Powiedziałem kierownikowi projektu, że ostatniemu aspektowi nie można w zasadzie nic zarzucić tak długo, jak długo on, kierownik, zagwarantuje, iż zespół będzie w stanie pracować bez zakłóceń, inaczej mówiąc: przeszkody będą przez niego sprawnie usuwane. Jednak kierownik projektu miał jeszcze jedno ważne zastrzeżenie: to jemu miała przypaść kontrola postępów projektu. W tym celu zdążył już wykorzystać wewnętrzne narzędzie do raportowania czasu pracy. Zamiast zwyczajnie zlecić zadania na wykonanie elementów Rejestru Produktu lub Rejestru Sprintu, próbował opracować funkcjonalnie lub technicznie umotywowane pakiety zadań. W konsekwencji nikt nie wiedział dokładnie, której grupie zadań przypisać swój czas. Nie było to jednak wcale takie dramatyczne, ponieważ również w tym przedsiębiorstwie w kontekście rejestracji czasu pracy obowiązywało motto: „Nieważne, dla którego zadania zaksięgujesz czas pracy — ważne, aby suma czasu pracy się zgadzała". Aby móc tej sztuce dla sztuki przeciwstawić podejście zorientowane na cel, wtajemniczyłem kierownika projektu w zagadkowy świat Wykresów Spalania. Idea kontroli stanu projektu z dokładnością co do dnia przypadła mu do gustu. Oczywiście musiał to być automatycznie generowany Wykres Spalania oparty na arkuszu kalkulacyjnym, ale nie chciałem pozbawiać go zamiłowania do elektronicznych narzędzi. Ze względu na to, że kierownik projektu otrzymał godziwe z jego punktu widzenia zadanie, nie wspomniałem o możliwości utrzymywania Wykresów Spalania przez zespół. Zespół był zadowolony, ponieważ ktoś inny podjął się tej rzekomo nudnej powinności, i skoncentrował się na realizacji zadań.

Podczas kolejnych Codziennych Młynów większość członków zespołu coraz odważniej przyjmowała na oczach całego zespołu odpowiedzialność za wykonywanie zadań. Dwojgu współpracownikom sprawiało to wciąż trudności, chociaż nikt ich nie ponaglał ani nie wyśmiewał. Większość wyszukiwała tym ścichapękom partnerów, którzy mogliby ich wesprzeć przy danym zadaniu. Jeśli brakowało zagadnienia odpowiedniego do zrealizowania w parze, wyruszali na poszukiwanie indywidualnego zadania. Gdy jeden z członków zespołu rozwiązywał zadanie sam, jego współpracownicy okazywali mu wiele fachowego wsparcia. Kiedy wreszcie

takie postępowanie stało się obowiązującym zwyczajem, wszyscy mogliśmy o wiele lepiej, a przede wszystkim w sposób dużo bardziej ukierunkowany na osiągnięcie celu, obchodzić się z niepewnością pojawiającą się cyklicznie podczas Codziennego Młyna.

Programowanie w parach sprawdza się, gdy pary są wyważone. Albo obaj partnerzy mają mniej więcej podobne kwalifikacje i wtedy mogą nauczyć się wiele od siebie nawzajem, albo jeden z partnerów obiera rolę mentora drugiego partnera, który przykładowo wdraża się w technologię. W obu przypadkach ważne jest, aby regularnie wymieniali się rolami: aktywną (przy klawiaturze) i pasywną. Jak wiesz, na proces rozwoju oprogramowania składa się wiele rutynowych czynności, które mogą być ćwiczone właśnie podczas programowania w parach.

Punktualność jest jedną z zasad zwinności, z której często rezygnuje się na rzecz promocji kreatywnego otoczenia. Jest to błędem, ponieważ Scrum jest procesem metodycznym. Teza, że określenia „zwinny" oraz „chaotyczny" można ze sobą utożsamiać, jest niepoprawna — jednak często przywołuje się ją jako uzasadnienie braku punktualności. Zmuszanie swoich współpracowników do żmudnego oczekiwania jest wyrazem braku szacunku (a na dodatek bywa kosztowne). Szacunek jest ważną wartością zwinną — ważniejszą niż kreatywność. Poza tym ścisłe przestrzeganie ustalonych ram czasowych (ang. *timebox*) pomaga skupić się na wyznaczonych zadaniach i ułatwia planowanie prac w Sprincie.

Podczas Codziennego Młyna poznaje się bardzo dobrze zarówno mocne, jak i słabe strony poszczególnych członków zespołu. Pomaga to Mistrzowi Młyna wspierać ich indywidualny rozwój i stawiać przed nimi stosowne wyzwania. Jeśli Mistrz Młyna rozpozna problematyczne wzorce zachowań (na przykład opisaną wcześniej skłonność do poszukiwania partnera do programowania, w przeciwieństwie do samodzielnego podjęcia się zadania), musi zadać sobie pytanie o przyczyny takiego stanu rzeczy. Jedynie dysponując wiedzą o przyczynach, może lepiej zintegrować członków zespołu i wyeliminować niepożądane wzorce zachowań.

Dzięki Wykresom Spalania można przekonać do Scruma zarówno klasycznych kierowników projektu, jak i inne osoby nadzorujące wyniki pracy, oraz wyjaśnić, że rozpowszechniony obraz chaotycznego, niekontrolowanego zwinnego projektu jest po prostu błędny. W zasadzie jest dokładnie odwrotnie: zwinne projekty oferują codzienną kontrolę postępów i dlatego pozwalają w dowolnym momencie wykryć aberracje i zastosować środki przeciwdziałające. Ta przewaga nad klasycznymi projektami, które gonią za rzeczywistością poprzez harmonogramy projektów, przekonuje wielu sceptyków.

2.10. Sprint to prędkość względna

Rozwój oprogramowania w tym projekcie stawiał przed nami duże i interesujące wyzwania. Oprócz przygotowania nowej architektury, należało również opracować dokładnie przemyślane środowisko wytwórcze. Opieraliśmy się przede wszystkim na doświadczeniu zdobywanym przez deweloperów z biegiem czasu w istniejących projektach, które to doświadczenie chcieli wykorzystać w nowym projekcie. W zasadzie był to dobry pomysł, który przyświecał zwinnej idei „inspekcji i adaptacji". Wszakże obarczanie nowego projektu kosztami (finansowymi i czasowymi) nie było wskazane — chyba że zostałoby to wyraźnie zaplanowane (co nie miało miejsca w naszym projekcie). W ten sposób przynieśliśmy ze sobą dług z innych projektów. Minęło dużo czasu, zanim pełne środowisko wytwórcze, pozwalające nam na dobrą i sprawną pracę, wyposażone w system kontroli wersji oraz narzędzia do budowania aplikacji i ciągłej integracji, ujrzało światło dzienne. Dlatego z perspektywy dążenia do celu funkcjonalnego pierwszy Sprint sprawiał wrażenie jazdy w rajdzie terenowym.

Ze względu na to, że infrastruktura aplikacji i struktura projektu przenikały się nawzajem, w krótkim czasie staliśmy się ekspertami do spraw refaktoryzacji. Podejście to świetnie się sprawdziło i stanowiło pomocną wprawkę, która opłaciła się później podczas refaktoryzowania kodu aplikacji.

Nie udało nam się wprowadzić techniki rozwoju oprogramowania sterowanego testami. Zdołaliśmy z Jo przemycić do Definicji Ukończenia pewne zabezpieczenie, polegające na konieczności osiągnięcia pozytywnego rezultatu testów automatycznych dla wszystkich elementów Rejestru Produktu. Mieliśmy nadzieję, że regularne posługiwanie się testami podczas programowania wznieci wystarczająco dużo pasji, że droga do rozwoju oprogramowania sterowanego testami nie będzie regułą narzuconą z zewnątrz, a jedynie logiczną konsekwencją. Niestety, to nie zadziałało. Skoro środowisko ciągłej integracji było wykorzystywane jedynie połowicznie, to mimo agitacji niektórych deweloperów nie pojawiła się również presja otoczenia, która mogłaby sprawić, że czyjś kod zostanie lepiej zabezpieczony, a budowanie aplikacji nie będzie skutkowało niepowodzeniem. Chwileczkę — czy ja właśnie napisałem „czyjś kod"? Czy kod źródłowy nie powinien należeć do wszystkich i czy każdy w zespole nie powinien przyjąć odpowiedzialności za wszystkie części systemu? W teorii tak, w praktyce wyglądało to jednak inaczej. Członkowie zespołu byli przyzwyczajeni do pracy w zakresie swoich specjalizacji. Dzięki programowaniu w parach likwidowaliśmy tego rodzaju silosy kompetencyjne, ale wciąż istniało wiele obszarów, w których cała dotycząca ich wiedza zgromadzona była tylko w jednej głowie. Biorąc pod uwagę szeroki wachlarz zagadnień technicznych, z którymi przyszło nam się uporać, idea ogólnego specjalisty, który zna wszystkie obszary systemu przynajmniej tak dobrze, że w przypadku błędu będzie w stanie wkroczyć z poprawką, była niezwykle trudna do realizacji. Ograniczyliśmy się więc jedynie do tego, aby wykształcić przynajmniej dwie osoby rozeznane w każdym z tematów.

Ścisła Definicja Ukończenia stawia wysokie wymagania środowisku rozwoju oprogramowania, chyba że budowanie i testowanie aplikacji ma być wykonywane manualnie. Jeśli odpowiednia architektura nie istnieje, zespół powinien się zastanowić, czy będzie w stanie postawić takie środowisko w Sprincie, pracując równolegle nad elementami Rejestru Produktu. Jeśli nie, proponowanym rozwiązaniem jest wykonanie tego rodzaju prac przygotowawczych w poprzedzającym Sprincie.

W większości świeżo uzwinnionych projektów, które miałem sposobność poznać, zdolności rzemieślnicze wielu deweloperów nie były dostatecznie ukształtowane. Wina nie spoczywała jedynie na deweloperach, ponieważ zdolności te mogły ewoluować jedynie w ramach oferowanych im alternatyw. Gdy brakuje czasu i budżetu na doskonalenie warsztatu, kultura rozwoju oprogramowania nie jest w stanie się wykształcić. Do tego potrzeba zarówno zewnętrznych bodźców (specjalistycznej literatury, szkoleń i wyjazdów na konferencje), jak i możliwości uzyskania w przedsiębiorstwie tymczasowej przestrzeni dla przyswajania i wykorzystywania wiedzy. W moich szkoleniach z programowania sterowanego testami stosowałem ćwiczenia *kata*[4] z wytwarzania kodu, aby wyćwiczyć niezbędne chwyty i uczynić z nich nawyk. Wielu uczestników jest początkowo zirytowanych powtarzającymi się ćwiczeniami — póki nie zrozumieją, że kluczem do nabycia codziennej wprawy jest zdyscyplinowana repetycja. Codzienne ćwiczenie wytwarzania kodu nie jest trwonieniem czasu, lecz wychodzi na dobre jakości rozwijanego oprogramowania.

[4] *Kata* to wysoce sformalizowany rodzaj ćwiczeń stosowany w wielu tradycyjnych sztukach i sportach walki — *przyp. tłum.*

Ten warsztat rzemiosła dewelopera i długa droga do jego opanowania są często opisywane w literaturze jako kunszt programowania (ang. *software craftsmanship*). Bez kunsztu programowania nawet najlepszy proces zwinny nic nie wskóra — ostatecznie, jeśli deweloperzy nie opanują swojego rzemiosła, wyjdzie z tego w najlepszym wypadku oprogramowanie średniej jakości.

2.11. Planowanie wymiarowe

Niektóre wymagania w Rejestrze Produktu były za duże, aby można je było zrealizować w trakcie jednego Sprintu. Nie dało się ich jednak łatwo rozłożyć na mniejsze elementy. Dla przykładu: chcieliśmy rozwijać pojedyncze elementy graficznego interfejsu użytkownika, które były „szyte na miarę" dla naszej aplikacji. W tym celu potrzebowaliśmy wielu podstawowych elementów graficznych. Pierwsze implementacje aplikacji musiały poradzić sobie bez specjalnie przygotowanych kontrolek. Ale jak mieliśmy to opisać w Rejestrze Produktu? Czy nie da się rozstrzygnąć zagadnienia realizacji elementu Rejestru Produktu podczas jednego Sprintu inaczej niż przez podział elementu?

Aby nie utracić z pola widzenia projektu całości wymagań, Koen Van Exem i Walter Hesius wprowadzili do gry nową zmienną kontrolną: *głębokość* elementu Rejestru Produktu. Posługując się metaforą drogi, zdefiniowali dla tego wymiaru cztery fazy: droga gruntowa, droga brukowana, droga asfaltowa i autostrada. Aby dotrzeć z punktu A do punktu B, należy zbudować drogę. Cel można osiągnąć niezależnie od jakości drogi (od tego, czy to droga gruntowa, czy autostrada), choć z różną szybkością czy różnym poczuciem komfortu.

Naszym funkcjonalnym elementom Rejestru Produktu brakowało informacji o głębokości. Droga gruntowa odpowiadała w nim graficznemu interfejsowi użytkownika, w którym zastosowano jedynie standardowe kontrolki. Interfejs użytkownika był w pewnej części trochę skomplikowany w obsłudze, ale dostarczał żądaną funkcjonalność. Wraz z wymianą kontrolek na nasze wyspecjalizowane rozwiązania rozbudowywaliśmy drogę gruntową do autostrady. Funkcjonalność pozostawała bez zmian, ale komfort obsługi wyraźnie wzrastał. Dalsze podnoszenie komfortu zaowocowało docelowo zbudowaniem autostrady.

Planowanie wymiarowe pomaga w iteracyjnej realizacji wymagań funkcjonalnych. Dzięki przyjęciu głębokości w roli nowego wymiaru planowania można podzielić Historyjkę Użytkownika na wiele Sprintów. Niemniej jednak każdy Sprint dostarcza autonomiczne rozwiązanie (z różnym standardem komfortu).

2.12. Przegląd Sprintu

Pod koniec pierwszego Sprintu spojrzeliśmy na naszą tablicę z zadaniami, tak jak czyniliśmy to każdego ranka. Wszyscy mogli z niej wyczytać, że zdołaliśmy osiągnąć dużo — chociaż nie wszystko, co pierwotnie przedsięwzięliśmy. Jest to jednak zupełnie naturalne dla zespołów rozpoczynających pracę zgodnie z wytycznymi Scruma, ponieważ oszacowanie tego, co może zostać wykonane w trakcie jednego Sprintu, wymaga wiele doświadczenia i różni się w zależności od konkretnego zespołu. Teraz, gdy rezultaty były tak widoczne i zrozumiałe, w całym zespole

pojawiły się po raz pierwszy uczucia satysfakcji i dumy. To osiągnięcie było postrzegane jako rezultat pracy całego zespołu. „I właśnie dlatego rezultaty pracy w Sprincie powinny zostać przedstawione przez cały zespół!" — obwieściłem podczas jednego z ostatnich Codziennych Młynów tego Sprintu. Zarezerwowaliśmy czas na przygotowanie Przeglądu Sprintu. Skoro osiągnęliśmy namacalne wyniki w postaci działającego oprogramowania, powinniśmy je zaprezentować na żywo — jak inaczej moglibyśmy zademonstrować, że zastosowaliśmy się do naszej surowej Definicji Ukończenia?

Wynikowym artefaktem pewnych elementów naszego Rejestru Produktu była koncepcja, a nie działający kod. Jak należało to pokazać na żywo? Zaproponowałem zaprezentowanie testów, które przeprowadzili deweloperzy, aby ocenić różnorakie alternatywy. W ten sposób dla uczestników Przeglądu Sprintu od razu stało się jasne, który z wariantów jest najbardziej odpowiedni. I dokładnie tak samo funkcjonowało to później.

Podczas Przeglądu Sprintu większość przyjęła postawę aktywną, chociaż wciąż nie wszyscy cechowali się tym podejściem. Przedstawienie spójnej i płynnej prezentacji było dla nas ważniejsze niż zmuszanie każdego członka zespołu do znalezienia się w centrum uwagi. Najbardziej zaskoczył mnie jeden z deweloperów, który nie należał do grona prelegentów, gdyż kariera estradowa nie przypadła mu szczególnie do gustu. W trakcie dyskusji ze zleceniodawcą przytaczał decydujące argumenty, które prowadziły do uznania poprawności wykonania poszczególnych elementów Rejestru Sprintu. To ucieszyło nas wszystkich, szczególnie zleceniodawcę, który zaobserwował, że do osiągnięcia przedstawianego rezultatu przyczynił się cały zespół. Był również mile zaskoczony, że zdołaliśmy faktycznie zaprezentować coś w terminie, do którego zobowiązaliśmy się, rozpoczynając Sprint — bez przekładania, bez wymówek, bez żadnych „ale". Na jedno pytanie nie byliśmy mu jednak w stanie odpowiedzieć: czy będziemy w stanie dotrzymać ambitnego terminu dostarczenia całego systemu. Wskazywaliśmy na dynamikę Rejestru Produktu, ciągłe doskonalenie wiedzy i wynikające z tego rewaloryzacje naszych zgrubnych oszacowań dla elementów Rejestru Produktu. „Ale czy nie powiedzieliście na początku projektu, że dzięki zastosowaniu zwinnego sposobu pracy możliwe będzie dotrzymanie terminu realizacji projektu?". Nie, tego nie powiedzieliśmy. Natomiast niewątpliwie nie omieszkaliśmy bacznie obserwować oczekiwań poszczególnych interesariuszy, a w szczególności zleceniodawcy, i reagować na wszelkie zapotrzebowanie.

2.13. Instrukcja pielęgnacji oczekiwań

Opuśćmy na chwilę nasz projekt na rzecz omówienia podstawowych zagadnień dotyczących zarządzania oczekiwaniami.

Według moich doświadczeń ludzie, którzy po raz pierwszy mają styczność ze zwinnymi metodami postępowania, albo wiążą z nimi bardzo wysokie oczekiwania, albo nie wiążą z nimi żadnych oczekiwań. Oba ekstrema wymagają adekwatnego podejścia. Poniżej powołuję się na konkretne oczekiwania, na które natknąłem się w opisywanym tu projekcie. Na pewno spotkasz się z nimi w podobnej postaci w innych projektach.

Zacznijmy od uprzednio wspomnianych oczekiwań zleceniodawcy. „Scrum powoduje, że wszystko przebiega szybciej" — tak brzmi od dawna wyrażana z nadzieją opinia. To stwierdzenie może okazać się słuszne, jeśli zwinny zespół zdołał już się dotrzeć. Rzadko ma to jed-

nak miejsce na początku zwinnego projektu. Główną odpowiedzialność ponosi tu Definicja Ukończenia, która nakłada bardzo wysokie wymagania na poziom warunków niezbędnych do ukończenia dowolnego elementu Rejestru Produktu. Im wyższe są wymagania, tym większy jest koszt ich spełnienia. Dlatego też element Rejestru Produktu jest docelowo faktycznie gotowy, a nie jedynie „wykonany w 90 procentach", z czym można się spotkać w przypadku innych metod prowadzenia projektu. Największą zaletą metod zwinnych jest nie szybkość, lecz duża przejrzystość postępów projektu. Umożliwia to ustalenie z dokładnością co do dnia, jak daleko na swojej drodze do osiągnięcia Celu Sprintu znajduje się zespół. Problemy pojawiające się podczas przemierzania tej drogi są wcześnie odnotowywane, rozpoznawane i — jeśli to możliwe — likwidowane. Projekty realizowane w Scrumie nie borykają się z mniejszą liczbą przeszkód niż projekty prowadzone w inny sposób, a jedynie częściej i wcześniej podejmowane są próby usunięcia tych przeszkód. W ten sposób projekty są znacznie łatwiejsze do kontrolowania z perspektywy zleceniodawcy oraz innych interesariuszy, ponieważ decyzje dotyczące projektu są podejmowane na podstawie twardych faktów, w przeciwieństwie do opierania się na domysłach i podrasowanych półprawdach.

Inną nadzieją, którą żywią często zleceniodawcy i kierownicy projektów, jest to, że Scrum wkomponuje się w tradycyjny model ról hierarchicznie zorganizowanego przedsiębiorstwa. W rzeczywistości Scrum ma własny model ról, dopasowany do organizacji projektu. Wprawdzie Scruma da się zintegrować ze strukturą liniową, ale stawia to wysokie wymagania kierownikom liniowym, którzy muszą stworzyć swoim współpracownikom środowisko promujące odpowiedzialność zbiorową i kreatywność. Sprawia to problem zarówno wielu kierownikom liniowym, jak i ich współpracownikom[5]. Pierwsi z nich czują się zbędni („Moi współpracownicy i tak decydują o wszystkim sami"), a ich władza zostaje rozproszona, podczas gdy drudzy są często przytłoczeni („Dlaczego nikt mi już nie mówi, co dalej?").

Wielu kierowników liniowych jest przyzwyczajonych do tego, że przy podziale obowiązków wśród swoich współpracowników w projekcie dysponują możliwością współdecydowania, a nawet prawem weta. W zwinnych projektach prawo to przechodzi na zespół. W efekcie oczekujące elementy Rejestru Produktu są wykonywane płynnie i z najwyższą możliwą jakością, przy czym nie powstają silosy kompetencyjne.

Kolejny problem pojawił się, gdy kierownicy liniowi podjęli się dysponowania swoimi współpracownikami bez wcześniejszego uwzględnienia potrzeb poszczególnych projektów. Zwinne projekty zakładają obecność stabilnego zespołu, który, jeśli to tylko możliwe, spędza 100 procent swojego czasu pracy w pojedynczym projekcie (z wyjątkiem tymczasowo powołanych ekspertów). Nawet jeśli nie zawsze ma to miejsce, skład i dostępność zespołu muszą być stałe na czas każdego pojedynczego Sprintu. Niestety, niejednokrotnie przeżyłem sytuację, w której kierownik liniowy spontanicznie „wyjmował" niektórych współpracowników z projektu w celu realizacji zadań specjalnych. To może stawiać pod znakiem zapytania osiągnięcie Celu Sprintu — choć pozostaje ono możliwe, ponieważ pozostali członkowie zespołu są w stanie rekompensować absencje nadgodzinami, co jednak można zalecić jedynie warunkowo. Oczywiście

[5] Mówiąc „współpracowników", mam na myśli osoby będące podwładnymi kierownika liniowego. Termin „zespół" jest umiejscowiony za blisko zdarzeń określających projekt i dlatego wprowadza w błąd. Wszystkie organizacyjne pojęcia, jak „dział", „grupa" i tak dalej, są zbyt konkretne. W jednej z firm, którym miałem kiedyś przyjemność doradzać, przełożeni mówili zawsze o swoich „ludziach", ale wydawało mi się to komiczne.

analogiczna sytuacja może również powstać wskutek choroby jednego z członków zespołu. Jest to jednak sytuacja nieprzewidziana, podczas gdy przenosiny w celu realizacji zadań specjalnych zazwyczaj da się zaplanować lub przełożyć. W zasadzie chodzi o to, aby skutecznie przekonać kierownika liniowego, że Sprint znajduje się pod ochroną. Sprawdza się to najlepiej, jeśli kierownik doświadczy i nauczy się mechanizmów zwinnego prowadzenia projektu na praktycznym przykładzie. Oprócz korzyści płynących ze zwinnych praktyk, muszą również zostać wyjaśnione wymogi wstępne i reguły gry dotyczące zwinnego projektu. Do nich należy również ochrona Sprintu. Z tych wymogów można następnie wyprowadzić i wspólnie uzgodnić oczekiwania wobec przełożonych liniowych.

Na konkurencji między zwierzchnictwem a projektem cierpi z reguły ten, kto znalazł się między młotem a kowadłem: współpracownik. Nie jest wcale łatwo sprostać jednocześnie wymaganiom stawianym przez kierowników liniowych i przez projekt. Członkowie zespołu oczekują w takich przypadkach w pierwszej kolejności pomocy ze strony Mistrza Młyna. Musi on koniecznie spełnić te oczekiwania, ponieważ jest to jego najważniejsze zadanie. To również Mistrz Młyna wspiera zespół w przestrzeganiu oraz ustawicznym rozwoju zwinnych wartości, zasad i praktyk, poprzez świecenie możliwie wyraźnym przykładem oraz zarządzanie sytuacją w projekcie i rozsądne pośredniczenie. Można oczekiwać wsparcia także od pozostałych członków zespołu. Do tego wlicza się również Właściciel Produktu, który przykładowo może wpłynąć na zrozumienie potrzeb zwinnych zespołów na szczeblu kierowniczym.

Niektórzy członkowie zespołu życzą sobie jasnego podziału ról w zespole. Jak to często bywa w życiu, nie istnieje uniwersalne rozwiązanie spełniające ten postulat. Zgodnie z tym, co rozważyliśmy już powyżej, nie każdy musi absolutnie wszystko wiedzieć i potrafić — tak długo, dopóki w zespole nie pojawią się silosy kompetencyjne.

Łatwiej zająć się innymi życzeniami członków zespołu, a mianowicie chęcią uzyskania większej ilości czasu na codzienne czynności i pozostałe projekty. Codzienne czynności muszą tak czy owak zostać uwzględnione w odpowiednim nakładzie czasu (w czym zawierają się z góry wiadome nieobecności) podczas Planowania Sprintu. Jeśli członek zespołu jest w tym samym czasie zajęty wieloma projektami, powinno się go zapytać, czy będzie w stanie w pełni sprostać oczekiwaniom stawianym przed nim we wszystkich toczących się jednocześnie projektach. W końcu każdy członek zespołu ponosi częściową odpowiedzialność za spełnienie obietnic składanych w każdym z projektów, w których uczestniczy. Często lepszym rozwiązaniem jest ustalenie nowych zespołów, tak aby mniejsza liczba osób pracowała jednocześnie w wielu różnych projektach.

Aby złagodzić strach kierowników liniowych przed odczuwaną utratą władzy, należy przygotować ich na stojące przed nimi nowe zadania. Powinni oni wdrożyć i zmotywować swoich współpracowników do działania propagującego odpowiedzialność zbiorową. Z klasycznych przełożonych powinni stać się partnerami do rozmowy w kontekście przygotowania decyzji (które następnie współpracownicy podejmują samodzielnie), jak i doradcami w zakresie funkcjonalności, trenerami, mentorami oraz partnerami do przedyskutowania pomysłów. Te zadania są tak wymagające, że nie powinno, a nawet nie może być więcej mowy o utracie władzy.

Konsekwentne wprowadzanie zwinnego sposobu myślenia ma w większości przedsiębiorstw fundamentalne skutki: nowe role organizacyjne w projekcie, zwiększenie ciężaru odpowiedzialności w zespole i stosowny wpływ na liderską rolę kierowników liniowych, którzy wyzbyli się złudnego przekonania, że projekt prowadzony Scrumem jest sam w sobie szybszy i bardziej skuteczny. Co więcej, połączyli to z przeświadczeniem, że zwinne projekty opierają się na

przejrzystości i codziennej kontroli, co stanowi idealny (a zarazem konieczny) wymóg do stosowania podejścia „inspekcji i adaptacji", znanego również jako cykl Deminga („zaplanuj – wykonaj – sprawdź – działaj").

Scrum oraz inne zwinne modele zmieniają organizację. W tym celu muszą zaangażować się wszyscy: od kadry zarządzającej do członków poszczególnych zespołów, a być może nawet klienci, dostawcy i partnerzy. Kto nie tylko zaakceptuje owe zmiany, ale potraktuje je jako szansę, ten odniesie korzyści z wprowadzonych zmian organizacyjnych, gdyż nowo powstała zwinna organizacja będzie lepiej przygotowana na nieuniknioną przyszłość i na teraźniejszość.

Na tym zakończę rozważania podejmujące temat ważny nie tylko w zwinnym środowisku. Kontynuujmy naszą opowieść, spoglądając ponownie w przeszłość.

2.14. Retrospekcja Sprintu

Podczas pierwszej Retrospekcji Jo i ja byliśmy bardzo spięci. Proszenie scrumowych nowicjuszy o ocenę ich pierwszego podejścia do zwinności jest zawsze pouczające.

Zaczęliśmy Retrospekcję od „pierwszej dyrektywy" Normana Kerthsa.

> Niezależnie od naszych odkryć rozumiemy i szczerze wierzymy, że każdy wykonał pracę najlepiej jak mógł, biorąc pod uwagę wiedzę, jaką w tym momencie dysponował, swoje zdolności i umiejętności, dostępne zasoby oraz aktualną sytuację.

Następnie narysowałem linię czasu, która symbolizowała ubiegły Sprint. Wszyscy członkowie zespołu mieli za zadanie nanieść na nią doświadczenia, które uważali za godne uwagi. Mogły być to sprawy zarówno merytoryczne, jak i osobiste. Po pięciu minutach koncentracji (zgodnie z przyjętymi ramami czasowymi!) w ciszy przykleiliśmy kolejno nasze kartki na linii czasu. Dominującą rolę odgrywały wydarzenia, które były dla współpracowników odmianą od codzienności, w tym przenosiny do nowego biura zespołu, różnorodne warsztaty i dobrze przeprowadzony Przegląd Sprintu. Niektórzy członkowie zespołu przyznawali otwarcie, że nie są w stanie przypomnieć sobie żadnego zdarzenia, które zasługiwałoby na ich uwagę — było to stwierdzenie, któremu pozwoliliśmy zawisnąć w przestrzeni bez odpowiedzi. Gdy wszystkie doświadczenia zostały już rozmieszczone, linia czasu oddawała dokładny, a w niektórych aspektach również zaskakujący obraz minionego Sprintu. Wcześniej byłem do tej metody nastawiony sceptycznie, ale muszę przyznać, że w tym zespole zadziałała ona wyśmienicie.

Pytanie: „Co funkcjonowało dobrze w minionym Sprincie?" dostarczyło ze strony niektórych członków zespołu sporo informacji zwrotnej, a pozostali wstrzymali się od komentarza. Doceniony został nowy sposób pracy, wyraźnie zorientowany na zespół. Zagadnieniem wartym uwagi dla prawie wszystkich członków zespołu było intensywne wykorzystanie wiki[6] — również dlatego, że poprawiało przejrzystość projektu z zewnątrz. Co więcej, do pozytywnie ocenionych elementów zaliczały się także osiągnięcia w dziedzinie architektury oprogramowania oraz wybór technologii. Nawet utworzenie przestrzeni zespołowej, które oznaczało dla wszystkich przynajmniej tymczasowe pożegnanie się z tradycyjnym stanowiskiem pracy, zostało przez niektórych członków zespołu odebrane pozytywnie. Z upływem czasu zaczęli oni

[6] Wiki — typ serwisu internetowego z treścią tworzoną i zmienianą za pomocą języka znaczników lub edytora WYSIWYG z poziomu przeglądarki internetowej — *przyp. tłum.*

traktować drogę między biurami jako pomocną w umysłowym przełączaniu się pomiędzy codziennymi czynnościami i projektem. Swoje miejsce znalazł tu nawet Przegląd Sprintu, częściowo dlatego, że tak dobrze uwydatniał poczucie zespołowości, a częściowo ponieważ zleceniodawca był pełen podziwu dla zrealizowania Celu Sprintu.

Po krótkiej przerwie odważyliśmy się poszukać odpowiedzi na pytanie, co można by ulepszyć. Ta lista była zdecydowanie dłuższa niż poprzednia. Większość propozycji udoskonaleń dotyczyła Scruma. Dwóch członków zespołu było zdania, że posługujemy się Scrumem zbyt restrykcyjnie. Rzeczywiście, Jo i ja próbowaliśmy rozpocząć od podręcznikowej wersji Scruma, aby zapewnić członkom zespołu punkt odniesienia. Jako że nikt z zespołu nie przeczytał żadnej książki na temat Scruma, nasz plan niewątpliwie spalił na panewce. Retrospekcja Sprintu opłaciła się już choćby dla tego pojedynczego przejawu krytyki. Ale w naszej implementacji Scruma wciąż występowało wiele miejsc wymagających ulepszenia. Na przykład kierownik projektu i wyznaczony Mistrz Młyna nie siedzieli w jednym pokoju z Zespołem Deweloperskim, co zostało odebrane jako utrudnienie. I chociaż Codzienny Młyn stawał się w trakcie Sprintu coraz krótszy, i tak sumarycznie zabierał zbyt wiele czasu. Ciekawy był zarzut, że w Rejestrze Produktu uwzględnione zostały przede wszystkim wymagania funkcjonalne, a wymagania techniczne i cele zostały umieszczone jedynie w tle, względnie nie były jawnie obecne w Rejestrze Produktu. Nasza wskazówka, że Rejestr Produktu, jak sama nazwa wskazuje, stanowi zbiór wszystkich właściwości produktu (które zazwyczaj mają naturę funkcjonalną), rozjaśniła wprawdzie tę kwestię, ale nie wzbudziła zbytniego entuzjazmu. Dla każdego przejawu krytyki staraliśmy się zdefiniować osobę odpowiedzialną i środki zapobiegawcze. Dla przykładu: niewystarczające zapewnienie jakości wykonanego zadania chcieliśmy w nadchodzącym Sprincie poprawić za pomocą rozszerzenia tablicy z zadaniami o kolumnę „przegląd kodu" oraz listy kontrolnej dla cyklicznych zadań. Jednak przy wielu poruszanych punktach nie udało się nam wysupłać propozycji ulepszeń. Konstruktywnej krytyki trzeba się po prostu nauczyć.

Płynnie wprowadziliśmy zaproponowane usprawnienia pełniące funkcję środków zapobiegawczych, co zespół życzliwie przyjął do wiadomości. Pewne problemy nie zostały rozwiązane, ponieważ nie zdołaliśmy ustalić dla nich odpowiednich udoskonaleń. Na przykład kierownik projektu nie chciał (albo nie mógł) przeprowadzić się do pokoju Zespołu Deweloperskiego. Zespół mógł jednak funkcjonować zadziwiająco dobrze pomimo nierozwikłanych problemów. Dlatego nie znaleźliśmy się w sytuacji, w której podjęlibyśmy za dużo środków zapobiegawczych na nadchodzący Sprint i nie moglibyśmy ich w pełni zastosować — było to odkrycie, którego regularnie dokonywałem w innych zespołach pracujących zgodnie z wytycznymi Scruma. Po drugiej stronie medalu znajdowały się wprawdzie zauważone, ale wciąż nierozwiązane problemy.

> Retrospekcja Sprintu to konieczność. Jest ona kluczem do ulepszania procesu — oddaje nastroje panujące w zespole i ujawnia problemy z motywacją. Kto rezygnuje z Retrospekcji, ten wcale nie stosuje Scruma, ponieważ zrzeka się możliwości ciągłego rozwoju i dostosowywania przebiegu swojego projektu do zmieniających się warunków.

2.15. Metaretrospekcja (klasycznie: podsumowanie)

Zwinny projekt to nie bułka z masłem. Z odnoszeniem sukcesu w zwinności związanych jest nieodłącznie wiele elementów. Zwinne projekty nie są łatwiejsze, czy mniej wymagające, niż klasyczne. Również problemy w zwinnych projektach nie rozpływają się w powietrzu, jak się

gdzieniegdzie uważa. Jest raczej tak, że w zwinnych projektach problemy są rozpoznawane szybciej. Dzięki temu zespół projektowy otrzymuje możliwość wspólnego usunięcia przeszkód stojących na jego drodze (albo pozostawienia problemu do rozwiązania przez Mistrza Młyna), zanim dojdzie do kosztownych powikłań. Wymaga to wyraźnego uświadomienia sobie problemu zgodnie z następującą dewizą: nie opłaca się czynić nikogo osobiście odpowiedzialnym za problem. Nie zmniejsza to samego problemu, a jedynie wiąże niepotrzebnie energię, którą lepiej wykorzystać na rozwiązanie problemu.

Ważnym powodem wprowadzenia zwinnego podejścia jest minimalizacja ryzyka w projekcie. Kto wkracza na nieznany teren, temu w dobrej wierze doradza się, aby zapewnił sobie jak najszybszy przypływ informacji zwrotnej dotyczącej wykonywanej pracy. Powyższy argument jest na ogół dobrze przyjmowany przez zarząd, ponieważ termin „minimalizacja ryzyka" odgrywa znaczącą rolę w klasycznym słownictwie z zakresu zarządzania.

Bycie zwinnym jest dostatecznie trudne samo w sobie, pozostanie zwinnym jest jednak nieporównywalnie trudniejsze. Profil wymagań dla członków zwinnego zespołu obejmuje odwagę, dyscyplinę, myślenie kompleksowe oraz kombinację umiejętności miękkich i zdolności rzemieślniczych. Nie jest wcale łatwo znaleźć takich współpracowników.

Aby móc z powodzeniem wprowadzić Scruma do nowego zespołu, który nie pracował wcześniej w sposób zwinny, niezmiernie ważne jest zadbanie o to, by wszystkie osoby przypisane do projektu otrzymały solidne podstawowe wykształcenie w zakresie zwinności, ukierunkowane na grupowe osiąganie celu. W przeciwnym wypadku zbyt dużo czasu w Sprincie pochłonie wyjaśnianie zwinnych wartości, zasad i praktyk. Jeśli przeszkolony zostanie jedynie Mistrz Młyna, to Zespołowi Deweloperskiemu (a często również Właścicielowi Produktu) będzie brakowało metodycznego arsenału naukowego.

Na starcie projektu musi się przede wszystkim zawiązać zespół. Jest to grupa ludzi różniących się wiedzą, rozkładem interesów, a po części nawet podłożem kulturowym, która powinna nagle wspólnie rozwinąć fragment oprogramowania, a do tego również odpowiedzialnie nakreślić ramy pracy — ten ciężar spadał wcześniej na barki kierownika projektu. Nic dziwnego, że wielu członków zespołu z początku pozostaje w cieniu i sonduje swoje położenie. Zwinne praktyki, jak Planowanie Pokerowe oraz znaczenie aspektów komunikacyjnych podczas Planowania Sprintu, stanowią dla wielu osób brzemię, ponieważ skrajnie wysoki jest stopień przejrzystości i co za tym idzie, również strach przed narażeniem się na śmieszność przy wszystkich. Zmotywowanie ludzi do wspólnej pracy oraz delikatne nakłanianie, aby się nie izolowali ani nie płoszyli, jednocześnie ich nie przeciążając, jest niczym spacer po linie. To właśnie Planowanie Pokerowe wraz ze swoją atmosferą zabawy pomaga wyzbyć się lęku i obudzić ducha zespołowości. Przeświadczenie, że pojedyncza osoba nie jest już samotna w odpowiedzialności za „swoje" oszacowanie nakładu pracy, jak dotąd przypadło do gustu wszystkim zespołom. A jeśli Planowanie Pokerowe zajmuje zbyt wiele czasu, to dostępne są również szybsze techniki, na przykład Magiczna Estymata[7].

[7] Magiczna Estymata (ang. *Magic Estimation*) jest techniką, która poddaje szacowaniu nie pojedynczy element, lecz grupę elementów rozkładanych równocześnie przy użyciu analizy porównawczej na pewnej ustalonej skali. Skalę stanowią najczęściej początkowe wartości ciągu Fibonacciego lub wybrane kolejne naturalne potęgi liczby 2 — *przyp. tłum.*

Zobowiązanie zespołu, a dokładniej obietnica osiągnięcia Celu Sprintu, to trudne, ale niezwykle istotne zadanie. Przypomina ono zespołowi o wspólnej odpowiedzialności i wzywa do podejmowania roztropnych decyzji.

Dobre zarządzanie oczekiwaniami jest niezmiernie ważne — szczególnie dlatego, że wiele osób nadal ma błędne wyobrażenie zasad, możliwości i granic zwinnych sposobów postępowania. Tylko wtedy, gdy oczekiwania są znane, można je okiełznać i powiązać z wymaganiami. W ten sposób włącza się do wspólnego rejsu ekskluzywnym statkiem wycieczkowym zarząd i przełożonych liniowych. Jeśli to się nie uda, projekt prowadzony Scrumem będzie prawdopodobnie przemierzał niczym okręt podwodny głębiny oceanów. Nie jest to nic złego, ale nie przybliża do celu, czyli zwinnej organizacji.

W chwili pisania tych słów (czerwiec 2011) omawiany projekt jest nadal realizowany. Stanowi on sukces zarówno pod względem funkcjonalnym, jak i technicznym. Również Scrum jest nadal w użyciu, chociaż zespół nie został w pełni przekonany do zalet zwinnego podejścia. Czasami zwyczajnie potrzeba trochę więcej czasu, by iskra rozbłysła.

2.16. Zwinne wartości w projekcie

Ludzie i interakcje	**ponad**	**procesy i narzędzia**
▲ Komunikacja w zespole miała duże znaczenie już w erze poprzedzającej Scruma. Wiele uwagi poświęcano indywidualnym potrzebom poszczególnych członków zespołu. Klient w ogóle nie ingerował w proces wytwarzania. Dostępny był zestaw narzędzi, które stosowano w innych projektach. Niemniej jednak zespół projektowy dysponował swobodą w zakresie wykorzystania własnych narzędzi.		
Działające oprogramowanie	**ponad**	**obszerną dokumentację**
▲ Wymagania są opisane w Rejestrze Produktu. Dokumentacja składa się z komentarzy osadzonych w kodzie oraz rozrastającej się wiki, która zawiera w istocie koncepcję i konwencje dla całego projektu. Jednak dla niektórych deweloperów koncepcja była ważniejsza niż działające oprogramowanie. Zanim zabrali się oni za implementację, chcieli najpierw wnikliwie poznać zagadnienie teoretyczne. Nie byli w stanie przyswoić sobie naszego pomysłu na konsekwentne programowanie sterowane testami.		
Współpraca z klientem	**ponad**	**formalne ustalenia**
▲ (1) ▲ (2) (1) Naszym celem było (i jest) utworzenie przenośnej platformy we współpracy z klientem. Nasze przedsięwzięcie byłoby wielokrotnie przedłużane, gdyby nie ograniczenie zależności codziennych zajęć od ściśle określonej roli (na przykład „architekta"). Wpływ budżetu projektu na współpracę był zauważalny, co jest jednak w pełni zrozumiałe w relacji klient – zleceniodawca. (2) Współpraca z rzeczywistymi klientami, czyli ekspertami dziedzinowymi, układała się fantastycznie.		
Reagowanie na zmiany	**ponad**	**podążanie za planem**
▲ Proces był wielokrotnie adaptowany, a Rejestr Produktu zmieniał się wyraźnie wraz z upływem czasu, proporcjonalnie do przyrostu wiedzy dotyczącej tego projektu.		

Skorowidz

I

J

K

L

M

N

O

P

R

S

Ś

T

V

W

X

Z

Zarządzaj zwinnie projektami!

Dynamika zmian w dzisiejszym świecie IT wymaga niezwykłej elastyczności i błyskawicznego adaptowania się do nowych warunków. Klasyczne techniki zarządzania projektami nie są w stanie podołać takiemu wyzwaniu, dlatego rynek zdobywają techniki zwinne — Scrum, Kanban i XP. Już dawno udowodniły swoją skuteczność i są świadomie wybierane przy nowych projektach. Zastanawiasz się, czy możesz wprowadzić te techniki w Twojej organizacji? Szukasz skutecznego sposobu realizacji tego zadania? Na te i dziesiątki innych trudnych pytań odpowiada ta niezwykła książka!

Znajdziesz w niej liczne studia przypadków, przygotowane przez ekspertów z bogatym doświadczeniem w codziennej pracy nad zwinnymi projektami. W trakcie lektury poznasz pułapki, jakie na Ciebie czyhają, oraz dowiesz się, jak sprytnie je ominąć. Ponadto przekonasz się, jak wprowadzić Scruma do klasycznej organizacji lub projektu z ustalonym budżetem. Poznasz również takie problemy, w których przypadku Kanban sprawdzi się wyśmienicie. Zwróć także uwagę na rozdział poświęcony przejrzystości w organizacji — to ona leży u podstaw sukcesu zwinnych metodyk. Książka ta jest obowiązkową lekturą dla wszystkich osób interesujących się zarządzaniem projektami, mających ambicję lub obowiązek wprowadzenia zwinnych technik w swojej organizacji.

Dzięki tej książce:

- → poznasz ciekawe przypadki wykorzystania Scruma, Kanbana i XP
- → wybierzesz właściwy sposób zarządzania w startupie
- → poznasz podstawy Scruma, Kanbana i XP
- → wprowadzisz zwinne metodyki w Twojej organizacji

helion.pl
księgarnia internetowa

Nr katalogowy: 23578

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/nowosci

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

Informatyka w najlepszym wydaniu

sięgnij po WIĘCEJ
KOD KORZYŚCI

ISBN 978-83-246-8843-2

cena: 39,90 zł

9 788324 688432