

## SCRUM GUIDE – PRZEWODNIK PO METODYCE SCRUM

Ken Schwaber, Jeff Sutherland

Luty 2010

Tłumaczenie:

Anna Gajewska, Tomasz Włodarek (tomek@poddzrewem.pl)

Konsultacja:

dr Marek Ćwiklicki

---

### Podziękowania

Scrum jest wzorowany na najlepszych praktykach i standardach przemysłowych, będących w zastosowaniu od dziesiątków lat. Oparty jest o teorię empirycznej kontroli procesów. Jak ujął to Jim Coplien w rozmowie z Jeffem Sutherlandem „wszyscy docenią Scruma, bowiem oddaje on dokładnie to, co robimy, gdy zostaniemy przyparci do muru”.

### Ludzie

Z pośród tysięcy osób, które przyczyniły się do powstania Scruma, należy wyróżnić tych, którzy aktywnie uczestniczyli w jego tworzeniu podczas pierwszych dziesięciu lat jego istnienia. Przede wszystkim Jeffa Sutherlanda, współpracującego z Jeffem McKenna, oraz Kena Schwabera, działającego wspólnie z Mikem Smithem i Chrisem Martinem. Po raz pierwszy Scrum został zaprezentowany publicznie podczas konferencji OOPSLA 1995. W trakcie kolejnych pięciu lat, w sposób znaczący przyczynili się do jego rozwoju Mike Beadle i Martine Devos. Ponadto podziękowania należą się także wszystkim innym osobom, bez pomocy których Scrum nie przyjąłby postaci znanej nam dzisiaj.

### Historia

Dziś, kiedy Scrum ma za sobą kilkunastoletnią historię stosowania w produkcji oprogramowania, warto wspomnieć miejsca, w których był początkowo wykorzystywany. Są to – Individual Inc., Fidelity Investments, oraz IDX (obecnie GE Medical).

### Tłumaczenie

Niniejszy przewodnik został przetłumaczony z oryginalnej angielskiej wersji, opracowanej przez Kena Schwabera i Jeffa Sutherlanda. Do jego powstania przyczynili się: Anna Gajewska, Tomasz Włodarek, dr Marek Ćwiklicki.

### Przeznaczenie

Metodyka Scrum jest stosowana od wczesnych lat 90-tych XX w. Niniejszy przewodnik opisuje jak wykorzystać tę metodykę w procesie wytwarzania złożonych produktów. Sam w sobie Scrum nie jest procesem czy techniką wytwórczą, opisuje jedynie ramy metodyczne w obrębie których, możliwe jest zastosowanie różnego rodzaju procesów i technik. Rolą Scruma jest wykazywanie nieefektywności stosowanych procesów tak, aby możliwe było ich ciągłe usprawnianie, **stanowi więc strukturę, która umożliwia wytwarzanie złożonych produktów.**

## Scrum – Teoria

Scrum, osadzony w teorii empirycznej kontroli procesu, wykorzystuje iteracyjne, przyrostowe podejście do procesu produkcji oprogramowania w celu osiągnięcia lepszej przewidywalności i kontroli ryzyka. Każda realizacja empirycznej kontroli procesu opiera się na trzech filarach opisanych poniżej.

### Pierwszy filar to przejrzystość

Reguła przejrzystości gwarantuje, że te aspekty procesu, który wpływają na jego wynik są widoczne dla osób zarządzających rezultatami procesu wytwórczego. Jednak nie tylko te aspekty muszą być widoczne; musi być również jasne, co konkretnie jest rezultatem tego procesu. To oznacza, że gdy osoba nadzorująca proces uważa, iż dana praca została wykonana, musi to być zgodne z jej definicją słowa „wykonane”.

### Drugi filar to kontrola

Wiele aspektów procesu musi podlegać kontroli na tyle często, aby można było wykryć niedopuszczalne odchylenia. Ustalając częstość kontroli, należy uwzględnić fakt, że sam akt kontrolowania wpływa na kontrolowane procesy. Dylemat pojawia się, gdy wymagana częstość inspekcji przekracza poziom tolerancji procesu. Na szczęście, wydaje się, że problem ten nie dotyczy procesu tworzenia oprogramowania. Innym istotnym czynnikiem są umiejętności oraz staranność osób kontrolujących wyniki pracy.

### Filar trzeci to adaptacja

Jeśli inspektor stwierdzi na podstawie przeprowadzonej kontroli, że jeden lub więcej aspektów procesu nie spełnia kryteriów dopuszczalności, i że produkt powstały w wyniku tego procesu będzie nieakceptowalny, musi dostosować proces lub materiał wejściowy podlegający temu procesowi. Dostosowanie to powinno nastąpić jak najszybciej, aby zminimalizować dalsze odchylenia.

W Scrumie występują trzy punkty, w których ma miejsce kontrola i adaptacja. Codzienny Scrum (*Daily Scrum Meeting*) służy kontroli postępu prac ku wytyczonemu Celowi Sprintu (*Sprint Goal*) i optymalizacji wartości każdego kolejnego dnia pracy. Ponadto, przegląd Sprintu (*Sprint Review*) i spotkania planistyczne (*Sprint Planning*) służą kontroli postępu prac nad realizacją wyznaczonego celu danego cyklu produkcyjnego (*Release Goal*) – zazwyczaj jest to osiągnięcie wersji produktu o jakości pozwalającej na wydanie go klientowi – oraz poczynieniu dostosowań optymalizujących wartość kolejnego Sprintu. I wreszcie, retrospektywa Sprintu (*Sprint Retrospective*) służy przeglądowi zakończonemu Sprintu i ustaleniu, jakie modyfikacje w kolejnym Sprincie zwiększą jego produktywność, dadzą poczucie spełnienia i radości z pracy.

## Scrum – Meritum

Struktura metodyki Scrum obejmuje: Zespoły Scrum (*Scrum Teams*) i powiązane z nimi role, ograniczenia czasowe (*time-boxes*), narzędzia (*artifacts*) i zestaw reguł postępowania.

Zespoły Scrum są zaprojektowane tak, aby osiągnąć wysoką elastyczność i wydajność; dlatego są

samoorganizujące się i interdyscyplinarne (międzyfunkcjonalne) oraz pracują w cyklicznych Sprintach. W każdym Zespole Scrum występują trzy role: 1) Mistrz (*Scrum Master*) odpowiedzialny za zrozumienie i spełnienie założeń procesu przez zespół; 2) Właściciel Produktu (*Product Owner*) odpowiedzialny za maksymalizację wartości pracy wykonywanej przez zespół; oraz 3) Zespół (*The Team*), który wykonuje pracę. Zespół składa się ze specjalistów posiadających wszelkie umiejętności konieczne do przekształcenia wymagań Właściciela Produktu w potencjalnie zbywalny fragment produktu w trakcie jednego Sprintu.

Stosowane w Scrumie ograniczenia czasowe zapewniają regularny rytm pracy. Elementy Scruma ograniczone czasowo to: spotkanie planistyczne wydania (*Release Planning Meeting*), spotkanie planistyczne Sprintu (*Sprint Planning Meeting*), Sprint, Codzienny Scrum (*Daily Scrum*), przegląd Sprintu (*Sprint Review*) oraz retrospektywa Sprintu (*Sprint Retrospective*).

Głównym elementem Scruma jest Sprint, czyli cykl pracy obejmujący okres maksymalnie jednego miesiąca (4 tygodnie), którego czas trwania pozostaje niezmienny w całym procesie wytwarzania produktu. We wszystkich Sprintach używa się tej samej struktury scrumowej i wszystkie one mają na celu dostarczenie potencjalnie zbywalnego przyrostu finalnego produktu. Sprints następują bezpośrednio po sobie.

W Scrumie używa się czterech podstawowych narzędzi. Rejestr produktowy (*Product Backlog*) to zhierarchizowana lista wszelkich elementów składających się na ostateczną postać produktu. Rejestr zadaniowy (*Sprint Backlog*) to lista zadań, których wykonanie doprowadzi do przetworzenia wybranych w danym Sprincie pozycji rejestru produktowego w gotowy fragment potencjalnie zbywalnego produktu. Wykres wypalania (*Burndown Chart*) to podstawowe narzędzie oceny postępu prac w projekcie. Pomiar wypalania dla projektu (*Release Burndown*) pokazuje elementy pozostające w rejestrze produktowym w stosunku do czasu zakończenia projektu, natomiast pomiar wypalania dla Sprintu (*Sprint Burndown*) ukazuje zadania pozostające w rejestrze zadaniowym w stosunku do końca Sprintu.

Reguły postępowania spajają ograniczenia czasowe, role i narzędzia. Są one kolejno opisane w tym poradniku. Przykładowo, jedną z zasad obowiązujących w Scrumie jest zabieranie głosu podczas Codziennego Scruma tylko przez członków Zespołu scrumowego, czyli osoby zaangażowane w przekształcanie rejestru produktowego w przyrost produktu. Sposoby wdrożenia metodyki Scrum, które nie są ścisłymi regułami, a jedynie sugestiami, zostały zamieszczone w sekcjach zatytułowanych „Wskazówka”.

#### **Wskazówka**

Gdy nie ma ustalonych reguł, użytkownicy metodyki Scrum powinni sami dojść do tego, co zrobić. Nie należy za wszelką cenę szukać idealnego rozwiązania problemu, ponieważ problemy zwykle szybko ulegają zmianie. Zamiast tego należy wypróbować jakieś rozwiązanie i przekonać się, jak ono działa. Kierować się należy empirycznymi mechanizmami Scruma – zasadami kontroli i adaptacji.

## **Role w Scrumie**

Zespół Scrum (*The Scrum Team*) składa się z Mistrza, Właściciela Produktu i Zespołu. Członkowie Zespołu scrumowego to „świnki”, zaś wszyscy inni to „kurczaki”. Właściciel Produktu jest „świnką” odpowiedzialną za rejestr produktowy. Zespół to „świnki” odpowiedzialne za wykonanie pracy, zaś Mistrz jest „świnką” odpowiedzialną za proces wytwórczy. Wszyscy pozostali to „kurczaki”. Kurczaki nie mogą mówić świnkom, jak mają wykonywać swoją pracę. Kurczaki i świnki pochodzą z tej historii:

*Kurczak i świnka siedzieli sobie razem, gdy nagle kurczak powiedział: „Założmy restaurację!”*

*Świnka zastanowiła się i zapytała: „A jak ją nazwiemy?”*

*Kurczak odparł: „Jaja na boczku!”*

*Na to świnka odrzekła: „O nie, dziękuję! Ty się w to tylko zaangażujesz, a ja – ja się poświęcę!”*

### **Mistrz**

Zadaniem Mistrza jest dopilnowanie przestrzegania przez Zespół scrumowy wartości, praktyk i reguł Scruma. Mistrz ma pomóc Zespołowi oraz całej organizacji zastosować metodykę Scrum. Uczy on Zespół, trenując i prowadząc jego członków do osiągnięcia większej wydajności i tworzenia produktów o wyższej jakości. Mistrz pomaga Zespołowi zrozumieć i stosować zasady samoorganizacji i wielofunkcyjności. Mistrz pomaga także Zespołowi scrumowemu działać sprawnie w środowisku, które nie jest jeszcze w pełni przystosowane do wytwarzania złożonych produktów. Kiedy Mistrz spowoduje, że taki stan rzeczy ulegnie zmianie, nazywamy to „usuwaniami przeszkód”. Mistrz stosuje przywództwo służebne do zapewnienia Zespołowi scrumowemu najlepszych możliwych warunków pracy.

#### **Wskazówka**

Mistrz współpracuje z klientami i kierownictwem, aby zidentyfikować osobę, która wcieli się w rolę Właściciela Produktu. Mistrz uczy Właściciela Produktu wykonywania jego pracy. Od Właściciela Produktu oczekuje się, że będzie wiedział, jak przy użyciu Scruma zoptymalizować wartość pracy. Jeżeli nie potrafi tego robić, odpowiedzialnością za ten fakt obarcza się Mistrza.

### **Właściciel Produktu**

Właściciel Produktu jest jedyną osobą odpowiedzialną za zarządzanie rejestrem produktowym i za czuwanie nad wartością pracy wykonywanej przez Zespół. Ta osoba prowadzi rejestr produktowy i ma za zadanie dopilnować, aby był on dla wszystkich widoczny i zrozumiały. Dzięki temu wszyscy wiedzą, które elementy rejestru mają najwyższy priorytet, a więc wiedzą nad czym będą pracować w kolejnych Sprintach.

Właściciel Produktu to jedna osoba, nie komitet. Mogą istnieć grupy, które doradzają tej osobie lub wpływają na nią, ale jeśli ktoś chce zmienić priorytet jednej z pozycji rejestru, musi przekonać do tego Właściciela. Firmy, które zaczynają stosować Scrum, mogą się przekonać, że z upływem czasu wpłynie to na ich metody ustanawiania priorytetów i wymagań.

#### **Wskazówka**

W komercyjnych projektach Właścicielem Produktu może być kierownik produktu (*Product Manager*). W projektach wewnętrznych Właścicielem może być kierownik funkcji biznesowej, która podlega informatyzacji.

Aby Właściciel Produktu odniósł sukces, wszyscy członkowie organizacji muszą szanować jego decyzje. Nikomu nie wolno nakazać Zespołowi, by pracował z innym zestawem priorytetów, a Zespołowi nie wolno słuchać kogoś, kto mówi coś innego niż Właściciel. Jego decyzje są widoczne w zawartości i priorytetach elementów rejestru produktowego. Potrzeba zapewnienia tej przejrzystości sprawia, że Właściciel Produktu musi zawsze robić wszystko, co w jego mocy, przez co jego rola staje się trudniejsza, ale też daje więcej satysfakcji.

## Zespół

Zespół programistów w czasie każdego Sprintu przetwarza rejestr produktowy w kolejne przyrosty potencjalnie zdanej do wydania funkcjonalności. Zespoły są interdyscyplinarne; członkowie Zespołu muszą posiadać wszystkie umiejętności potrzebne do wytworzenia kolejnego przyrostu produktu. Często członkowie Zespołu posiadają wyspecjalizowane umiejętności, na przykład: programowanie, kontrola jakości, analiza biznesowa, architektura, projektowanie interfejsu użytkownika, czy zarządzanie bazami danych. Jednak umiejętności wspólne dla wszystkich członków Zespołu – to znaczy umiejętność zespołowego podjęcia zobowiązania i przekształcenia wymagań w używalny produkt – są ważniejsze od umiejętności, których ze sobą nie dzielą. Ci, którzy nie chcą brać udziału w tworzeniu kodu, ponieważ są architektami czy projektantami, nie pasują do Zespołu. Każdy musi dołożyć swoją część pracy, nawet, jeśli wymaga to nabycia nowych umiejętności lub przypomnienia sobie starych. W Zespołach Scrum nie istnieje hierarchia służbowa, nie stosuje się również nazewnictwa stanowisk i od tej reguły nie ma wyjątków. Zespoły nie składają się też z podzespołów oddanych konkretnym obszarom pracy, jak na przykład testowanie czy analiza biznesowa.

### Wskazówka

Mistrz może być członkiem Zespołu, na przykład programistą wykonującym zadania w Sprincie. To jednak często prowadzi do konfliktów, gdy Mistrz musi wybierać pomiędzy usuwaniem przeszkód a wykonaniem swoich zadań. Mistrz nigdy nie powinien być Właścicielem Produktu.

Zespoły są samoorganizujące się. Nikt – nawet Mistrz – nie może mówić Zespołowi, jak przekształcać elementy rejestru produktowego w używalną funkcjonalność. Zespół dochodzi do tego sam. Każdy członek Zespołu wykorzystuje swoje kompetencje do rozwiązania problemu. Powstająca synergia podnosi całkowitą wydajność i skuteczność Zespołu.

### Wskazówka

Właściciel Produktu może być członkiem Zespołu, również programującym, jednak taka dodatkowa odpowiedzialność może przeszkadzać mu we współpracy z interesariuszami. Właściciel Produktu nigdy nie może być Mistrzem.

Optymalna liczebność Zespołu to siedem osób, plus-minus dwie. Gdy członków Zespołu jest mniej niż pięcioro, zachodzi mniej interakcji i uzyskuje się mniejszą produktywność. Co więcej, Zespół może doświadczyć niedoboru umiejętności w pewnych etapach Sprintu i nie będzie w stanie dostarczyć używalnego fragmentu produktu. Natomiast gdy w Zespole jest więcej niż dziewięć osób, wymagana jest intensywniejsza koordynacja. Duże Zespoły generują zbyt wielką złożoność zarządzania empirycznym procesem. Jednak znane są przypadki Zespołów, które przekroczyły dolną lub górną granicę liczebności, a jednak odniosły sukces. Właściciel Produktu i Mistrz nie są w tej liczbie uwzględniani,

chyba, że są jednocześnie „świnkami” odpowiedzialnymi jednocześnie za wykonywanie zadań z rejestru zadaniowego.

Skład Zespołu może się zmienić jedynie z końcem Sprintu. Za każdym razem, gdy zmienia się skład Zespołu, zmniejsza się jego produktywność uzyskana dzięki samoorganizacji. Dlatego zmianom składu osobowego Zespołu towarzyszyć musi duża ostrożność.

### **Ograniczenia czasowe**

Ograniczenia czasowe (*time-box*) w Scrumie dotyczą: spotkania planistycznego wydania, Sprintu, spotkania planistycznego Sprintu, przeglądu Sprintu, retrospektywy Sprintu i Codziennego Scruma.

### **Spotkanie planistyczne wydania**

Przedmiotem spotkania planistycznego wydania jest ustalenie celu i planu realizacji projektu, które Zespoły i reszta organizacji może zrozumieć i będzie wspierać. Planowanie wydania odpowiada na pytania: "W jaki sposób przekształcić wizję w produkt, który odniesie sukces na rynku?", "Jak możemy sprostać oczekiwaniom klienta i zapewnić wymagany zwrot inwestycji, a nawet je przewyższyć?". W planie realizacji ustala się: cel ostateczny projektu, wstępne założenia rejestru produktowego, opis cech i funkcjonalności, które produkt będzie zawierał w chwili wydania, oraz główne zagrożenia związane z realizacją projektu. Plan ten wyznacza również przypuszczalną datę zakończenia pracy i koszt wytworzenia produktu, które powinny być wiążące, o ile do planu nie zostaną wprowadzone zmiany. Dzięki temu organizacja może ze Sprintu na Sprint, w oparciu o kolejne przyrosty produktu, nadzorować postęp prac i dokonywać niezbędnych korekt w planie realizacji projektu.

Etap planowania wydania jest opcjonalny. Jeśli Zespoły Scrum rozpoczną pracę bez tego etapu, brak jego wyników (planu wydania) będzie stanowić przeszkodę, którą należy usunąć. Czynności prowadzące do usunięcia tej przeszkody staną się elementem rejestru produktowego.

W metodyce Scrum produkty tworzone są iteracyjnie: każdy Sprint wytwarza przyrost produktu, począwszy od elementów najbardziej pożądanym i najbardziej ryzykownych. Kolejne Sprints tworzą kolejne elementy produktu. Każdy z przyrostów jest potencjalnie zbywalnym wycinkiem całości. Gdy powstanie ilość elementów wystarczająca, aby produkt stał się wartościowy i użyteczny dla inwestora, następuje wydanie produktu klientowi.

Wiele organizacji już posiada ustalony proces planowania. W większości przypadków znaczna część planowania odbywa się na początku projektu, a w trakcie jego trwania nic się nie zmienia. W scrumowym planowaniu wydania definiuje się cel całościowy oraz prawdopodobne efekty pracy. Takie planowanie zwykle wymaga nie więcej niż 15-20% czasu, jaki organizacja zużywała w tradycyjnym planowaniu. Niemniej w projekcie scrumowym odbywa się również planowanie dokładnie na czas (*just-in-time*, JIT) podczas przeglądów Sprintu i spotkań planistycznych Sprintu, oraz codziennie podczas każdego Codziennego Scruma. W ogólnym rozrachunku, prace planistyczne w Scrumie pochłaniają nieco więcej wysiłku niż tradycyjne planowanie projektu.

Planowanie wydania wymaga estymacji i przypisania priorytetów kolejnym pozycjom rejestru produktowego. Istnieje wiele technik spoza metodyki Scrum, które można skutecznie wykorzystać w tym procesie.

## Sprint

Sprint jest iteracją. Sprints zamykają się w ograniczeniach czasowych. Podczas Sprintu Mistrz ma za zadanie dopilnować, by nie wprowadzono żadnych zmian, które wpłynęłyby na Cel Sprintu. Skład osobowy Zespołu i cele jakościowe muszą pozostać niezmiennie przez cały Sprint. Sprints zawierają i składają się z: spotkania planistycznego Sprintu, prac programistycznych, przeglądu Sprintu, i retrospektywy Sprintu. Sprints następują bezpośrednio po sobie, bez przerw pomiędzy nimi.

Aby coś osiągnąć, tworzymy projekt. W produkcji oprogramowania wykorzystuje się projekty do stworzenia produktu lub systemu. Definicja projektu obejmuje opis tego, co ma powstać, plan, jak to coś wytworzyć, opis prac wykonywanych według tego planu oraz powstały w rezultacie produkt końcowy. Każdy projekt posiada horyzont, czyli zaakceptowane ramy czasowe. Jeśli horyzont jest zbyt odległy, może zdarzyć się, że w między czasie zmieni się definicja produktu, pojawi się zbyt dużo zmiennych, ryzyko będzie zbyt duże, itd. Scrum jest metodyką odpowiednią dla projektów na tyle złożonych, że przygotowywanie planów o horyzoncie czasowym dłuższym niż jeden miesiąc byłoby dla nich zbyt ryzykowne. Przewidywalność projektu musi być kontrolowana przynajmniej co miesiąc, a ryzyko, że projekt może wymknąć się spod kontroli lub stanie się nieprzewidywalny, powinno być kontrolowane i minimalizowane nie rzadziej niż co miesiąc.

### Wskazówka

Gdy Zespół rozpoczyna przygodę ze Scrumem, dwutygodniowe Sprints pozwalają uczyć się bez błędzenia w niepewności. Sprints tej długości mogą zostać zsynchronizowane z pracą innych Zespołów scrumowych pracujących w dłuższych Sprintach, na przykład przez dodanie do siebie dwóch kolejnych przyrostów.

Sprint może zostać zakończony przed końcem swojego ograniczenia czasowego. Jedyne Właściciel Produktu jest uprawniony do zamknięcia Sprintu, choć może tak uczynić za namową interesariuszy, Zespołu lub Mistrza. Jakie warunki muszą zachodzić, by nastąpiła konieczność odwołania Sprintu? Kierownictwo może być zmuszone do odwołania Sprintu, jeśli Cel Sprintu jest nieaktualny. Tak może się stać, gdy firma zmienia kierunek, lub gdy zmieniają się warunki rynkowe czy technologiczne. Ogólnie rzecz biorąc, Sprint powinien zostać odwołany, gdy nie ma już sensu jego realizacja, biorąc pod uwagę zaistniałe okoliczności. Jednak, ponieważ Sprint nie trwa długo, odwoływanie go jest rzadko sensowne.

Gdy Sprint zostaje odwołany, dokonuje się przeglądu wszystkich wykonanych, zakończonych elementów rejestru produktowego. Akceptowane są te, które stanowią potencjalnie zbywalny przyrost. Wszystkie pozostałe trafiają z powrotem do rejestru produktowego z początkowymi estymacjami. Jakkolwiek wykonana nad nimi praca uznana zostaje za straconą. Zakończenie Sprintu w ten sposób pochłania zasoby, ponieważ wszyscy muszą się przegrupować podczas nowego zebrania planistycznego, by rozpocząć nowy Sprint. Przedwczesne zakończenia Sprintu są często traumatyczne dla Zespołu i w związku z tym są bardzo rzadkie.

## Spotkanie planistyczne Sprintu

Spotkanie planistyczne Sprintu odbywa się zawsze, gdy trzeba zaplanować nową iterację. Jego ograniczenie czasowe to osiem godzin przy planowaniu Sprintu miesięcznego. Dla krótszych Sprintów na ten etap należy przeznaczyć proporcjonalnie krótszy czas, na przykład cztery godziny dla dwutygodniowych Sprintów. Zebranie to składa się z dwóch części. W pierwszej części, decyduje się o tym, co ma być wykonane w czasie Sprintu. W drugiej części, Zespół ma ustalić, w jaki sposób zbudować funkcjonalność w postaci przyrostu produktu w czasie tego Sprintu. Istnieją więc dwie części spotkania planistycznego Sprintu, koncentrujące się kolejno na tym „co?” i „jak?” należy zrealizować. Niektóre Zespoły scrumowe łączą te dwie części ze sobą.

### Wskazówka

Zwykle tylko 60-70% całego rejestru zadaniowego opracowuje się w trakcie spotkania planistycznego Sprintu. Pozostałe zadania są zarysowane bardzo ogólnie i uszczegóławiane później, albo oszacowane zgrubnie, a następnie rozbijane na drobniejsze i estymowane bardziej precyzyjnie już w trakcie Sprintu.

W części pierwszej (czterogodzinnej dla miesięcznych Sprintów) Zespół zajmuje się pytaniem „co?”. Wtedy Właściciel Produktu przedstawia pozycje rejestru produktowego o najwyższym priorytecie. Zespół i Właściciel współpracują, by ustalić, jaka funkcjonalność ma być wypracowana podczas Sprintu. Danymi wejściowymi w tym spotkaniu są: rejestr produktowy, ostatnio stworzony przyrost produktu, możliwości produkcyjne Zespołu w planowanym Sprincie i dotychczasowa wydajność Zespołu. Ilość pracy wybranej przez Zespół zależy tylko i wyłącznie od niego: jedynie Zespół może ocenić, ile jest w stanie osiągnąć podczas nadchodzącego Sprintu.

Po wyborze zakresu pracy określa się Cel Sprintu (*Sprint Goal*). Jest to cel, który zostanie osiągnięty przez implementację wybranego fragmentu rejestru produktowego. Ustalenie go ma uzmysłowić Zespołowi, po co buduje się kolejny przyrost produktu. Cel Sprintu jest składową celu wydania. Cel Sprintu istnieje także, aby zapewnić Zespołowi nieco swobody, jeśli chodzi o wytwarzaną funkcjonalność. Na przykład, celem Sprintu może być: „automatyzacja funkcjonalności modyfikującej konto klienta poprzez możliwość zastosowania bezpiecznej, odtwarzalnej transakcji w warstwie pośredniej (*middleware*)”. Podczas pracy Zespół ma w pamięci ten cel. Aby go zrealizować, Zespół implementuje funkcjonalność i niezbędną infrastrukturę. Jeśli praca okazuje się trudniejsza niż oczekiwano, Zespół współpracuje z Właścicielem i implementuje funkcjonalność tylko częściowo.

W drugiej części spotkania planistycznego Sprintu Zespół zajmuje się pytaniem „jak?”. W czasie tego drugiego zebrania (czterogodzinnego dla miesięcznych Sprintów), Zespół zastanawia się, jak przekształcić elementy wybrane z rejestru produktowego podczas pierwszego bloku spotkania („co?”) w kompletny przyrost produktu. Zazwyczaj Zespół zaczyna od planowania pracy; wtedy właśnie zidentyfikowane zostają zadania, czyli szczegółowe opisy prac prowadzących do przekształcenia rejestru produkto-

### Wskazówka

Jeśli Zespół przeczuwa, że przeszacował ilość pracy, spotyka się z Właścicielem Produktu, aby przesunąć część pracy na kolejne Sprints, lub zredukować zakres pozycji rejestru produktowego, które wybrano do danego Sprintu. Jeśli Zespół uważa, że ma zapas czasu, może współpracować z Właścicielem przy wyborze dodatkowych prac.

wego w działający program (produkt). Prace te powinny zostać podzielone tak, by każde z nich mogło zostać wykonane w czasie maksymalnie jednego dnia. Lista tych zadań to rejestr zadaniowy (*Sprint Backlog*). Zespół organizuje się sam, aby podjąć się realizacji zadań z rejestru zadaniowego, czy to podczas spotkania planistycznego Sprintu, czy w miarę potrzeb, „dokładnie we właściwym czasie” (JIT) podczas trwania Sprintu.

Właściciel Produktu jest obecny podczas drugiej części spotkania planistycznego Sprintu, aby objaśniać rejestr produktowy i pomagać w osiągnięciu kompromisu pomiędzy swoimi oczekiwaniami a możliwościami produkcyjnymi Zespołu. Jeśli Zespół ustali, że ma zbyt dużo lub zbyt mało pracy, może renegecjować zakres pracy z Właścicielem. Nowy Zespół zwykle w czasie tego spotkania uświadamia sobie po raz pierwszy, że zwycięży lub pójdzie na dno razem, jako zespół, nie indywidualne osoby. Członkowie Zespołu zdają sobie sprawę, że muszą na sobie polegać. Gdy sobie to uświadamia, zaczynają się samoorganizować oraz nabierają cech i zachowań prawdziwego zespołu.

### **Przegląd Sprintu**

Pod koniec Sprintu organizuje się spotkanie przeglądu Sprintu (*Sprint Review*). Jest to spotkanie zawarte w czterogodzinnym ograniczeniu czasowym (dla miesięcznych Sprintów). Dla krótszych Sprintów na to spotkanie należy przeznaczyć odpowiednio mniej czasu (na przykład dwie godziny dla Sprintów dwutygodniowych). Podczas przeglądu Sprintu Zespół scrumowy i interesariusze podsumowują wykonaną pracę. Opierając się na tym oraz na zmianach wprowadzonych do rejestru produktowego podczas Sprintu, ustalają, jakie prace mogą zostać wykonane w kolejnych Sprintach. Jest to spotkanie nieformalne, podczas którego dokonuje się prezentacji funkcjonalności, aby ułatwić wspólną pracę wszystkich zainteresowanych nad ustalaniem kolejnych kroków.

Agenda tego spotkania zawiera przynajmniej następujące punkty: Właściciel Produktu stwierdza, które funkcjonalności zostały wykonane, a które nie; Zespół omawia prace wykonane w trakcie Sprintu zakończone sukcesem, jak również napotkane problemy i metody ich rozwiązania; następnie Zespół prezentuje wykonaną pracę i odpowiada na pytania; Właściciel Produktu omawia stan rejestru produktowego oraz wysuwa propozycje prawdopodobnych terminów zakończenia pracy przy różnych założeniach co do tempa pracy; następnie cała grupa omawia przedstawione fakty i ich istotność w ustaleniu zakresu kolejnych prac. Tym samym przegląd Sprintu dostarcza cennych informacji do wykorzystania w trakcie kolejnego spotkania planistycznego Sprintu.

### **Retrospektywa Sprintu**

Po przeglądzie Sprintu, a przed kolejnym spotkaniem planistycznym, Zespół przeprowadza retrospektywę Sprintu. W czasie tego spotkania, trwającego nie dłużej niż trzy godziny dla Sprintów miesięcznej długości (dla krótszych należy zarezerwować odpowiednio mniej czasu), Mistrz zachęca członków Zespołu do przejrzania, w ramach procesu i praktyki Scrum, ich pracy programistycznej, aby w kolejnym Sprincie uczynić ją bardziej efektywną i przyjemną. Istnieje literatura opisująca techniki przydatne w czasie retrospektywy.

Celem retrospektywy jest sprawdzenie przebiegu minionego Sprintu pod kątem osób biorących udział w przedsięwzięciu, relacji zachodzących między nimi, procesu i narzędzi. Inspekcja ta powin-

na zidentyfikować i zhierarchizować główne elementy – te, które były pozytywne oraz te, które, gdyby zostały zrealizowane inaczej, mogłyby wpłynąć pozytywnie na efekt pracy. Dotyczy to składu Zespołu, rozkładu spotkań, narzędzi, definicji tego, co „gotowe”, metod komunikacji i procesów stosowanych w przekształcaniu rejestru produktowego w „gotowe” fragmenty produktu. W trakcie trwania retrospektywy Zespół powinien ustalić kroki naprawcze, które zostaną podjęte w kolejnych Sprintach. Te zmiany stanowią *adaptację* wynikłą z empirycznej *inspekcji*.

### **Codzienny Scrum**

Każdy Zespół spotyka się codziennie na piętnastominutowym spotkaniu – Codziennym Scrumie. To spotkanie odbywa się o tym samym czasie i w tym samym miejscu w ciągu całego Sprintu. Podczas tego spotkania każdy członek Zespołu wyjaśnia:

*Co zrobił od ostatniego spotkania?  
Co będzie robił do następnego spotkania?  
Jakie napotyka przeszkody?*

Codzienny Scrum poprawia komunikację, eliminuje potrzebę innych spotkań, identyfikuje i usuwa przeszkody w pracy, podkreśla i promuje szybkie podejmowanie decyzji i podnosi poziom znajomości stanu prac projektowych w całym Zespole.

Mistrz ma obowiązek dopilnować, by spotkanie się odbyło. Zespół jest odpowiedzialny za przeprowadzenie Codziennego Scruma. Mistrz uczy Zespół, jak utrzymać krótki czas spotkań, egzekwując przestrzeganie reguł i pilnując, by wszyscy mówili zwięźle. Mistrz również ma za zadanie dopilnować, by „kurczaki” nie zabierały głosu i w żaden inny sposób nie przeszkadzały w Codziennym Scrumie.

Codzienny Scrum nie jest spotkaniem statusowym w tradycyjnym tego słowa rozumieniu. Nie jest otwarty dla wszystkich, a jedynie dla osób przekształcających rejestr produktowy w przyrost produktu (czyli dla Zespołu). Zespół podjął się osiągnięcia Celu Sprintu i zrealizowania wybranych elementów z rejestru produktowego. Codzienny Scrum jest okazją do kontroli postępu prac ku Celowi Sprintu (dzięki odpowiedziom na powyższe trzy pytania). Następnie rozmowy są kontynuowane w krótkich pod-spotkaniach, tak, aby wprowadzić niezbędne zmiany (adaptacje) do kolejnych prac realizowanych w Sprincie. Celem jest optymalizacja prawdopodobieństwa osiągnięcia przez Zespół Celu Sprintu. Codzienny Scrum jest najważniejszym spotkaniem inspekcyjno-adaptacyjnym w scrumowym procesie empirycznym.

### **Scrumowe narzędzia**

Narzędzia stosowane w Scrumie to: rejestr produktowy (*Product Backlog*), wykres wypalania dla wydania (*Release Burndown*), rejestr zadaniowy (*Sprint Backlog*) i wykres wypalania dla Sprintu (*Sprint Burndown*).

## Rejestr produktowy i wypalanie w projekcie

Wymagania dotyczące produktu wytwarzanego przez Zespół zebrane są w rejestrze produktowym. Odpowiedzialnym za jego zawartość, dostępność i ustalenie priorytetów jest Właściciel Produktu. Rejestr produktowy nigdy nie jest zamknięty. Jego wstępna wersja zawiera jedynie początkowo znane i najlepiej rozumiane wymagania. Rejestr produktowy ewoluuje wraz z produktem i środowiskiem, w którym będzie używany. Rejestr jest dynamiczny w tym znaczeniu, że stale się zmienia, aby uwzględnić to, czego produkt wymaga, aby być odpowiednim, konkurencyjnym i użytecznym. Rejestr istnieje tak długo, jak istnieje produkt.

Rejestr produktowy reprezentuje wszystko, co potrzebne, by stworzyć i wypuścić na rynek udany produkt. Rejestr jest listą wszystkich cech, funkcji, technologii, ulepszeń i napraw błędów, reprezentujących zmiany, które zostaną wprowadzone w produkcie przed kolejnym jego wypuszczeniem na rynek. Elementy rejestru posiadają następujące atrybuty: opis, priorytet i koszt realizacji. Priorytet ustala się na podstawie ryzyka, wartości i konieczności realizacji danego elementu rejestru. Istnieje wiele technik ustalania wartości tych atrybutów.

Rejestr produktowy jest uszeregowany według priorytetu. Pozycje o najwyższym priorytecie pozwalają natychmiast podjąć prace programistyczne nad nimi. Im wyższy priorytet, tym pilniejsza praca, tym bardziej jest przemyślana i tym większy stopień porozumienia co do jej rzeczywistej wartości. Elementy o wyższym priorytecie są przejrzystsze i zawierają więcej szczegółowych informacji niż elementy o niższym priorytecie. Na podstawie większej przejrzystości i zwiększonej szczegółowości można dokonać trafniejszych estymacji. Im niższy priorytet, tym mniej szczegółów, aż do poziomu, na którym ledwie można wyodrębnić poszczególne elementy rejestru.

W miarę, jak produkt jest używany, jak jego wartość rośnie, a rynek reaguje i dostarcza informacji zwrotnej, rejestr produktu zmienia się w coraz dłuższą i bardziej wyczerpującą listę. Wymagania stale się zmieniają. Rejestr jest żywym dokumentem. Zmiany w wymaganiach biznesowych, warunkach rynku, technologii i obsadzie pracowniczej powodują zmiany w rejestrze. By zminimalizować przeróbki, jedynie elementy o najwyższym priorytecie są szczegółowo opisane. Pozycje rejestru, którymi Zespół Scrumowy zajmie się w przeciągu kilku następnych Sprintów, są dokładne i szczegółowe oraz podzielone w

### Wskazówka

Elementy rejestru są zwykle ujęte w formie opowieści użytkownika (*User Stories*). Przypadki użycia (*Use Cases*) są również odpowiednie, ale jako bardziej formalne sprawdzają się lepiej w tworzeniu oprogramowania o znaczeniu krytycznym dla życia ludzkiego lub strategicznym dla danego przedsięwzięcia lub przedsiębiorstwa.

### Wskazówka

Zespoły Scrum często 10% każdego Sprintu poświęcają na cyzelowanie (*grooming*) rejestru produktowego, aby spełnić wymagania powyższej definicji. Elementy rejestru tak uszczegółowione znajdujące się na szczycie listy (najwyższy priorytet, największa wartość) są rozdzielane lub grupowane tak, aby mieściły się w pojedynczych Sprintach. W procesie cyzelowania rejestru zostały już dobrze przeanalizowane i przemyślane. Gdy nadchodzi spotkanie planistyczne Sprintu, prace o najwyższym priorytecie są więc dobrze rozumiane, a ich wybór do realizacji jest oczywisty.

taki sposób, aby każda z nich mogła zostać wykonana w trakcie pojedynczego Sprintu.

Często kilka Zespołów scrumowych pracuje nad jednym produktem, a do wyznaczenia kolejnych prac używa się wspólnego rejestru produktowego. Stosuje się wtedy dodatkowy atrybut, który grupuje elementy rejestru – według cech, technologii lub architektury – i który jest często używany celu zorganizowania pracy Zespołów.

Na wykresie wypalania dla produktu (lub wydania) rejestruje się ilość pozostałej do wykonania pracy zgodnie z szacunkami w rejestrze w stosunku do czasu realizacji projektu. Szacowana ilość pracy ukazywana jest w jednostkach pracy, na które zdecydował się Zespół i cała organizacja (osoby różnych). Jednostką czasu (osoby odciętych) jest zwykle Sprint.

Wstępne estymacje elementów rejestru produktowego są wyznaczane podczas planowania wydania (*Release Planning*). Dla elementów pojawiających się w trakcie trwania projektu estymacje wyznaczane są na bieżąco. W procesie cyzelowania rejestru estymacje poddawane są analizie i w każdej chwili mogą zostać zmienione. Za wszystkie estymacje odpowiedzialny jest Zespół. Właściciel Produktu może wpływać na Zespół, pomagając mu zrozumieć problem i dokonać wyboru, gdy zachodzi konieczność kompromisu (*trade-offs*), ale ostateczna estymacja jest dokonywana przez sam Zespół. Właściciel Produktu przez cały czas, na bieżąco, uaktualnia wykres wypalania produktu według rejestru produktowego. Linia trendu jest wyznaczana na podstawie zmian w ilości pozostałej pracy.

#### **Wskazówka**

Również testy akceptacyjne są często używane jako atrybut elementu rejestru. Nierzadko mogą one zastąpić szczegółowe opisy słowne zestawem kroków testowych, które jednoznacznie wskazują, jak implementacja danego elementu rejestru ma działać kiedy już zostanie ukończona.

### **Rejestr zadaniowy i wypalanie w Sprincie**

Rejestr zadaniowy zawiera opis prac, które Zespół ma wykonać, aby przekształcić elementy rejestru produktowego w pełni funkcjonalny przyrost produktu. Wiele z tych zadań opracowuje się podczas spotkania planistycznego Sprintu. Rejestr reprezentuje całość prac, które Zespół uznaje za niezbędne, aby osiągnąć Cel Sprintu. Zadania w rejestrze zadaniowym muszą być zdekomponowane. Taka dekompozycja jest konieczna, by zmiany w postępie pracy były zrozumiałe w trakcie komunikowania ich na Codziennych Scrumach. Zwykle zadania dekomponuje się do momentu, gdy ich wykonanie zajmuje jeden dzień lub mniej.

Zespół modyfikuje rejestr zadaniowy w czasie trwania Sprintu, w szczególności dodaje zadania wynikłe już w trakcie Sprintu. Kiedy przychodzi do realizacji indywidualnych zadań, może się okazać, że potrzeba ich będzie więcej lub mniej, lub że dane zadanie będzie wymagać więcej lub mniej czasu niż się spodziewano. Jeśli pojawia się potrzeba wykonania dodatkowych prac, Zespół dopisuje je do rejestru zadaniowego. W miarę, jak zadania są wykonywane i kończone, aktualizuje się ilość przewidywanej pracy pozostałej do końca każdego zadania. Gdy zadanie uznaje się za zbędne, zostaje ono usunięte. Jedynie Zespół może zmieniać rejestr zadaniowy w trakcie Sprintu (dodawać/usuwać zadania), jak również tylko Zespół może zmienić treść zadań lub estymacje. Rejestr zadaniowy jest

dobrze widocznym, rzeczywistym obrazem pracy, jaką Zespół planuje wykonać w czasie Sprintu i jest własnością tylko i wyłącznie Zespołu.

Wykres wypalania w Sprincie to wykres przedstawiający ilość zadań z rejestru pozostałych do wykonania w Sprincie w stosunku do czasu Sprintu. By stworzyć ten wykres, należy ustalić, ile pracy pozostało, sumując estymacje zadań z rejestru w każdym dniu Sprintu. Ilość pracy pozostałej do wykonania w Sprincie jest sumą pracy pozostałej do wykonania w całym rejestrze zadaniowym. Sumy te należy kontrolować codziennie, tworząc w ten sposób wykres pozostałej do wykonania pracy w stosunku do czasu. Rysując linię łączącą punkty wykresu, Zespół może kontrolować postęp wykonywania swych prac w Sprincie. W Scrumie nie jest istotny czas już spędzony na wykonaniu prac – jedynymi istotnymi wskaźnikami są ilość pozostałej pracy i data końcowa Sprintu.

#### Wskazówka

Linia trendu może być niewiarygodna w ciągu pierwszych dwóch czy trzech Sprintów projektu, chyba, że Zespoły pracowały już wcześniej razem, znają dobrze produkt i rozumieją jego technologiczne podstawy.

Jedną z reguł Scruma odnosi się do celu każdego Sprintu, to jest dostarczenia przyrostów potencjalnie zbywalnej funkcjonalności. To, co musi zostać przez Zespół scrumowy zapewnione, aby wykonana praca mogła zostać uznana za potencjalnie zbywalny produkt, określamy mianem kryterium gotowości produkcyjnej.

#### „Gotowe”, czyli kryterium gotowości produkcyjnej (*Definition of Done, DoD*)

Metodyka Scrum wymaga od Zespołu, aby w każdym Sprincie powstał przyrost funkcjonalności produktu. Ten przyrost musi być potencjalnie zbywalny, ponieważ Właściciel Produktu musi być w stanie w każdym momencie podjąć decyzje o wdrożeniu tej funkcjonalności. Aby tak się mogło stać, przyrost musi być skończoną porcją całości – musi być „gotowy”. Każdy przyrost musi być zintegrowany z poprzednimi i dokładnie przetestowany, aby dać nam pewność, że wszystkie fragmenty produktu działają po połączeniu.

#### Wskazówka

Gdy tylko jest to możliwe, należy ręcznie rysować wykres wypalania na wielkim arkuszu papieru i wywiesić go w pomieszczeniu, gdzie Zespół pracuje. Członkowie Zespołu prędzej dostrzegą duży, widoczny wykres, niż zajrzą do Excela czy innego narzędzia, by go zobaczyć.

W tworzeniu oprogramowania, stwierdzenie, że funkcjonalność jest „gotowa” może dla jednych oznaczać, że kod źródłowy jest w miarę czysty, zrefaktoryzowany, przetestowany jednostkowo, zbudowany i po testach akceptacyjnych. Dla innych może to oznaczać, że po prostu istnieje kod. Jeżeli nie wszyscy wiedzą, jaka jest obowiązująca definicja „gotowego”, to nie zadziałają dwa pozostałe filary empirycznej kontroli procesu. Jeśli określamy coś jako „gotowe”, wszyscy muszą wiedzieć, co „gotowe” oznacza.

„Gotowe” definiuje to, co ma na myśli Zespół, gdy podejmuje się „przygotowania” jednej z pozycji rejestru produktowego w Sprincie. Niektóre produkty nie wymagają dokumentacji, a więc definicja „gotowego” nie zawiera istnienia dokumentacji. Całkowicie „gotowy” przyrost produktu zawiera

wszystkie analizy, projekty, kodowanie, refaktoryzację, dokumentację i testy dla tego przyrostu i dla wszystkich elementów rejestru produktowego wykonanych w ramach tego przyrostu. Testowanie może obejmować testy jednostkowe, systemowe, przez użytkownika (funkcjonalne) i regresyjne, jak również testy niefunkcjonalne, na przykład: wydajności, stabilności, bezpieczeństwa i integracyjne. „Gotowość” obejmuje również umiędzynarodowienie produktu. Czasami Zespół nie jest w stanie na tym etapie włączyć do swej definicji „gotowego” wszystkich elementów wymaganych do implementacji. Właściciel Produktu musi to rozumieć. Ta pozostała część pracy musi zostać wykonana, zanim produkt będzie oddany do wdrożenia i użytkowania.

### Uwagi końcowe

Niektóre organizacje nie są w stanie zbudować kompletnego przyrostu w czasie jednego Sprintu. Czasami nie posiadają zautomatyzowanej infrastruktury testującej, by wykonać wszystkie potrzebne testy. W takim przypadku dla każdego przyrostu produktu tworzy się dwie kategorie: pracy „wykonanej” i „niewykonanej”. Praca „niewykonana” to porcja każdego przyrostu, która musi zostać ukończona w późniejszym czasie. Właściciel Produktu wie dokładnie, co kontroluje na końcu Sprintu, ponieważ przyrost spełnia wymogi definicji „gotowe”, a Właściciel zna tę definicję. „Niewykonana” praca zostaje dodana do rejestru produktowego jako element „praca niewykonana”, tam się nawarstwia i jest właściwie ukazywana przez wykres wypalania. Ta technika pozwala na utrzymanie klarowności w postępie prac ku celowi ostatecznemu. Inspekcja i adaptacja w przeglądzie Sprintu będzie na tyle dokładna, na ile dokładna jest ta klarowność.

Na przykład, jeśli Zespół nie jest w stanie wykonać testów wydajnościowych, regresyjnych, stabilności, bezpieczeństwa i integracji dla każdego elementu rejestru produktowego, oblicza się proporcje tej pracy do prac, które mogą zostać wykonane (analiza, projektowanie, kodowanie, refaktoryzacja, dokumentacja, testy jednostkowe i przez użytkownika). Powiedzmy na przykład, że te proporcje to sześć porcji „wykonanych” i cztery „niewykonane”. Jeśli Zespół wykona zadania odpowiadające tym sześciu jednostkom pracy (estymacja przez Zespół zależy od tego, co wiedzą, że potrafią wykonać), cztery jednostki dodaje się do elementu „praca niewykonana” w rejestrze produktowym.

Ze Sprintu na Sprint, „niewykonana” praca z każdego przyrostu nawarstwia się i trzeba się nią zająć, zanim produkt zostanie wydany. Zwykle ilość pracy przyrasta liniowo, choć istnieje również pewien rodzaj kumulacji pracy wykładniczo, zależnie od charakterystyki danej organizacji. Pod koniec projektu dodaje się Sprints wydaniowe (*Release Sprints*), by zamknąć prace „niewykonane”. Ilość Sprintów jest nieprzewidywalna w takim stopniu, w jakim kumulacja pracy „niewykonanej” nie jest liniowa.

#### Wskazówka

W niektórych organizacjach do rejestru dodaje się więcej pracy niż się jej wykonuje. Może to spowodować, że linia trendu będzie na stałym poziomie, a nawet lekko wznosząca. By to zrekompensować i zachować przejrzystość, można wyznaczyć nowy poziom zerowy wykresu, gdy praca zostaje dodana lub odjęta. Należy tu dodawać oraz odejmować jedynie najistotniejsze zmiany, a wyznaczenie nowego „zera” musi być dobrze uzasadnione.