Wyrozębski P., Metodyka SCRUM [w] Metodyki zarządzania projektami, wyd. Bizarre, Warszawa 2011

Paweł Wyrozębski

13. Metodyka SCRUM

Geneza metodyki

Korzenie metodyki SCRUM sięgają drugiej połowy lat 80-tych XX w., kiedy to H. Takeuchi i I. Nonaka, dwaj inżynierowie-naukowcy japońscy opublikowali w Harvard Business Review artykuł zatytułowany *The New New Product Development Game*[136]. W artykule tym poddali krytyce szeroko stosowane w procesie opracowania nowych produktów podejście sekwencyjne, polegające na niezależnej pracy wyspecjalizowanych zespołów funkcjonalnych, które jak w sztafecie przekazywały sobie pałeczkę od fazy rozwoju koncepcji, poprzez testowanie wykonalności, projektowanie, rozwój produktu, produkcję pilotażową, aż do produkcji finalnej. Taki sposób realizacji projektu nosił nazwę PPP – „phased program planning" i stosowany był m.in. w projektach NASA.

Nonaka i Takeuchi, analizując firmy takie jak Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M, Xerox i Hewlett-Packard, zauważyli sposób postępowania zgoła odmienny od kosmicznego potentata. Podejściu tradycyjnemu przeciwstawili oni własną holistyczną koncepcję silnie zachodzących na siebie faz realizacji projektu, toczących się nie w sposób liniowy, lecz wyłaniających się w postaci iteracyjnych eksperymentów z ciągłych, wzajemnych interakcji międzyfunkcjonalnego zespołu odpowiedzialnego za projekt od początku do jego zakończenia. Dla lepszego zobrazowania koncepcji Nonaka i Takeuchi odnieśli ją do sposobu gry charakterystycznego dla rugby, gdzie „zespół jako całość stara się zyskać pole, przekazując piłkę raz do przodu, raz do tyłu"[137]. Jako główne filary nowego podejścia wskazali oni sześć elementów: inherentną niestabilność, samoorganizujące się zespoły, zachodzące na siebie fazy rozwoju produktu, grupowe uczenie się, subtelną kontrolę oraz organizacyjny transfer wiedzy.

Pomysł Japończyków został następnie rozwinięty przez Petera DeGrace'a i Leasliego Huleta Stahla w książce pod tytułem *Wicked Problems, Righteous Solutions*[138].

W publikacji tej autorzy wskazali na przyczyny niepowodzenia podejścia kaskadowego w rozwoju oprogramowania. Były to m.in.: brak możliwości pełnego zrozu-

[136] Takeuchi H., Nonaka I., *The New New Product Development Game*, Harvard Business Review, Jan/Feb 1986.

[137] Tamże.

[138] DeGrace P., Stahl L.H., *Wicked Problems, Righteous Solutions: A Catalogue of Modern Software Engineering Paradigms*, Yourdon Press, 1990.

mienia wymagań przed rozpoczęciem projektu, możliwość rzeczywistego określenia oczekiwań klientów dopiero w momencie otrzymania przez nich wstępnej wersji oprogramowania, częste zmiany wymagań w trakcie procesu tworzenia oprogramowania, zastosowanie nowych narzędzi i technologii, utrudniających przewidzenie skutecznej strategii implementacji[139].

Na podstawie analiz różnych modeli tworzenia oprogramowania DeGrace i Stahl doszli do wniosku, iż modelem, który w najlepszy sposób rozwiąże powyższe trudności jest model „wszystko naraz" (ang. all-at-once). Model ten zakładał, że praca nad wszystkimi dotychczas sekwencyjnymi etapami produkcji oprogramowania odbywać się będzie jednocześnie, zaś jej najprostszą realizacją będzie jeden superprogramista odpowiedzialny za produkt od początku do końca[140]. Modele takie rzeczywiście wykorzystywano w praktyce i chociaż miały również wady, były dobrze oceniane m.in. przez IBM[141]. Kontynuując odniesienie Takeuchiego i Nonaki do gry w rugby, opisując swoją koncepcję, posłużyli się terminem młyn (ang. scrum).

Bezpośrednimi twórcami metodyki młyna – SCRUM byli Jeff Sutherland z firmy Easel Corporation oraz Ken Schwaber, Dyrektor Zarządzający firmy Advanced Development Methods. Jeff Sutherland był Głównym Inżynierem w zespole Object Studio, który, zainspirowany wcześniejszymi badaniami nad modelami tworzenia oprogramowania, zdefiniował role, zatrudnił pierwszego Właściciela Produktu i Szefa Scrum'a, opracował pierwszy wykaz prac produktu (ang. product backlog), wykaz prac sprintu (ang. sprint backlog) i zbudował pierwszy portfel produktów stworzonych z wykorzystaniem metodyki SCRUM[142].

Pierwszy szczegółowy opis metody, który datuje się na 1993 rok Sutherland przedstawił w artykule *Agile Development: Lessons Learned from the First Scrum*[143], zaś w 1995 r. wspólnie z K. Schwaberem na konferencji *Object-Oriented Programming, Systems, Languages & Applications* (OOPSLA) w Stanach Zjednoczonych po raz pierwszy zaprezentowali metodykę szerokiej publiczności. W 2002 r. i 2004 r. ciągle

[139] Brooks F.P., *The Mythical Man Month: Essays on Software Engineering*, Adison-Wesley 1995.

[140] Sutherland J., *Agile Development: Lessons Learned from the First SCRUM*, Cutter Agile Project Management Advisory Service, Executive Update, Vol. 5, No. 20, www.scrumalliance.org.

[141] Brooks F.P., *The Mythical Man Month: Essays on Software Engineering*, Adison-Wesley 1995.

[142] Sutherland J., Schwaber K., *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*, www.scrumtraininginstitute.com.

[143] Sutherland J., *Agile Development: Lessons Learned from the First SCRUM*, Cutter Agile Project Management Advisory Service, Executive Update, Vol. 5, No. 20, www.scrumalliance.org.

doskonalona przez autorów metodyka SCRUM została opublikowana w książce *Agile Software Development with Scrum*[144].

Główne założenia metodyki

Autorzy metodyki podkreślają, że SCRUM jest przede wszystkim szkieletem, zbiorem zasad, które pozwalają szybko i efektywnie zorganizować pracę zespołową oraz zrealizować powierzone zadania bardziej wydajnie i przy wyższej niż zazwyczaj jakości produktu finalnego. Podejście to umożliwia zespołom samoorganizowanie się, deleguje na nie decyzje o przyjęciu ilości pracy do wykonania, jak również decyzję, jak to zrobić.

Metodyka, zgodnie z pryncypiami Agile, zaprojektowana jest iteracyjne i elastycznie, czyli tak, aby w jak najlepszy sposób adaptować projekt do zmieniających się oczekiwań interesariuszy i umożliwić w ten sposób dostarczenie produktu na czas i przy jak najmniejszej ilości pracy wykonanej niepotrzebnie. W tym celu SCRUM posługuje się krótkimi, regularnymi etapami realizacji projektu tzw. sprintami (ang. sprints), kładzie bardzo silny nacisk na jasną i czytelną priorytetyzację wymagań klienta i szybkie dostosowywanie rezultatów pracy do jego rzeczywistych potrzeb[145].

W przeciwieństwie do niektórych metodyk zwinnych jak np. Extreme Programing (XP), Scrum w większym stopniu koncentruje się na sferze zarządzania projektem (ról, procesów, podejmowania decyzji itp.), podczas gdy kwestie techniczne pozostają w gestii samego zespołu. XP podchodzi raczej do projektu od strony technicznej i inżynierskiej, dostarczając technik i narzędzi wykorzystywanych bezpośrednio przez wykonawców produktów[146]. Jednocześnie, warto podkreślić, iż SCRUM może być w łatwy sposób połączony i zintegrowany z innymi iteracyjnymi metodykami zarządzania np. z praktykami Unified Process lub Extreme Programming[147].

Schematyczny model realizacji projektu zgodnie z metodyką SCRUM przedstawia rysunek 48:

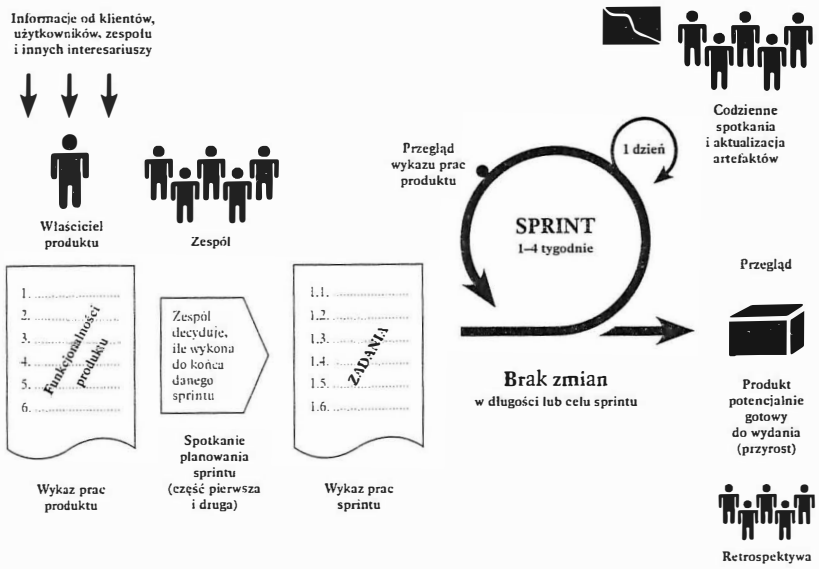
[144] Schwaber K., Beedle M., *Agile Software Development with SCRUM*, Prentice Hall 2002; Schwaber K., *Agile Software Development with SCRUM*, Microsoft Press 2004.

[145] Sutherland J., Schwaber K., *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*, www.scrumtraininginstitute.com.

[146] Asproni G., *Wstęp do SCRUM*, Software Developer's Journal 6/2006, www.sdjournal.org.

[147] Sutherland J., *A Brief Introduction to SCRUM*, Scrum Alliance 2006.

Rys. 48. Model metodyki SCRUM



Źródło: Deemer P., Benefield G., Larman C., Vodde B., *SCRUM Primer: An Introduction to Agile Project Management with Scrum*, ver. 1.2, 2010, http://scrumtraininginstitute.com.

Struktura metodyki SCRUM oparta jest na trzech elementach: trzech rolach, trzech ceremoniach i trzech artefaktach:[148]

- Role:
  - właściciel produktu (ang. product owner),
  - szef scruma (ang. scrum master, scrummaster),
  - zespół (ang. scrum team).
- Ceremonie:
  - spotkanie planowania sprintu (ang. sprint planning meeting),
  - spotkanie przeglądu sprintu (ang. sprint review meeting),
  - codzienne zebrania (ang. daily scrum meeting).
- Artefakty:
  - wykaz prac produktu (ang. product backlog),
  - wykaz prac sprintu (ang. sprint backlog),
  - wykres malejący (ang. burndown chart).

[148] Asproni G., *Wstęp do SCRUM*, Software Developer's Journal 6/2006, www.sdjournal.org.

Role

Pierwszą rolą, zgodnie z metodyką SCRUM, jest **właściciel produktu (ang. product owner).** Właściciel produktu jest rolą, w której odpowiedzialności znajduje się produkt finalny oraz wszystkie funkcjonalności, które będą dostarczały wartość klientom i użytkownikom. Rola ta zarządza także priorytetami ich implementacji. Głównym zadaniem właściciela produktu jest zapewnienie, że zespół robi „właściwe, z biznesowego punktu widzenia rzeczy".

W przypadku produktu opracowywanego do sprzedaży na rynku właściciel produktu ponosi odpowiedzialność również za opłacalność jego komercjalizacji (rozumianą w kategoriach ROI). W projektach realizowanych na potrzeby wewnętrzne właściciel produktu będzie utożsamiany z osobą reprezentującą interesy i potrzeby użytkowników.

Właściciel produktu odpowiada za:[149]

- dostarczenie przez produkt wartości klientom;
- identyfikację i definiowanie funkcjonalności produktu;
- nadanie priorytetów funkcjonalnościom zgodnie z ich wartością dla klientów;
- podejmowanie decyzji o datach i zawartości kolejnych wydań produktu;
- aktualizowanie zestawu funkcjonalności i ich priorytetów w kolejnych sprintach;
- akceptację lub odrzucenie wykonanej pracy.

Rola właściciela produktu podobna jest do roli Menedżera Produktu lub Menedżera Marketingu Produktu. Warto jednocześnie podkreślić, iż osoba zajmująca to stanowisko musi pozostać aktywna i ściśle współpracować z pozostałymi rolami SCRUM'a. Rola właściciela produktu jest rolą jednoosobową i nie może być łączona z rolą szefa scruma.

Drugą rolą w metodyce SCRUM jest rola zespołu. Zadaniem zespołu jest wykonanie pracy potrzebnej do dostarczenia działających funkcjonalności produktu, wskazanych przez właściciela produktu.

W skład zespołu wchodzi zazwyczaj 5 do 9 osób – posiadających umiejętności specjalistyczne konieczne do pracy w projekcie i osiągnięcia celów sprintu. W skład zespołu wchodzą zazwyczaj: analitycy, projektanci, programiści, testerzy, administratorzy i inni.

Zespół w SCRUM'ie charakteryzuje się zupełnie płaską strukturą. Każdy członek występuje na równych zasadach i może zaangażować się w dowolne zadanie. Zespół

[149] Sutherland J., Schwaber K., *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*, www.scrumtraininginstitute.com.

ten jest jednocześnie zespołem samoorganizującym się, czyli członkowie we własnym zakresie decydują o zobowiązaniach zespołu i sposobie ich wypełnienia, odpowiadają za organizację swojej pracy i wykonują ją bez ingerencji z zewnątrz. Mogą wybrać dowolną ścieżkę działania pod warunkiem, że prowadzi ona do osiągnięcia celu sprintu i na jego koniec będą mogli zaprezentować działającą funkcjonalność oprogramowania. Zarówno szef, jak i właściciel produktu muszą pozostawić zespołowi swobodę poruszania się w ramach pracy przydzielonej do wykonania w ramach danej iteracji, nie powinni ingerować w zatwierdzony plan.

Ważnym zaleceniem metody jest, aby członkowie zespołu byli w całości zaangażowani w jego pracę. Podobnie jak w przypadku metody łańcucha krytycznego – SCRUM postuluje rezygnację z wielozadaniowości (dodatkowa praca poza projektem, wiele projektów lub produktów jednocześnie).

Ostatnią rolą wyróżnioną w metodyce jest rola **szefa SCRUM'a.** Szef scruma, wbrew swojej nazwie, nie posiada uprawnień rozkazodawczych, nie pełni roli kierownika, a tym bardziej roli kierownika projektu. Szef scruma jest rolą wspierającą zespół i właściciela produktu w realizacji projektu z sukcesem, odpowiada za zrozumienie i stosowanie zasad metodyki w zespole, wspiera procesy uczenia się. Szef scruma działa na rzecz zespołu i jednym z jego kluczowych zadań jest bycie „buforem" między zespołem a jego bliższym i dalszym otoczeniem, osobą odsuwającą przeciwności, blokującą interwencje z zewnątrz, rozwiązującą problemy, mogące utrudnić zespołowi realizację zadeklarowanych celów.

Szef scruma:[150]

- zapewnia, że zespół jest gotowy do pracy i produktywny,
- umożliwia bliską współpracę pomiędzy wszystkimi rolami i funkcjami,
- usuwa bariery,
- ochrania zespół przed ingerencjami z zewnątrz,
- zapewnia, że projekt prowadzony jest zgodnie z metodyką SCRUM.

Trzeba w tym miejscu silnie podkreślić, że metodyka SCRUM nie przewiduje roli kierownika projektu takiej, jaka jest znana z innych metodyk zarządzania projektami. Jego zadania zostały rozdzielone na trzy istniejące role. Zgodnie z ideą SCRUM'a właściciel produktu odpowiada za wyznaczenie i priorytetyzację celów, szef scruma pełni rolę „guru", „opiekuna" zespołu, zaś sam zespół – złożony przecież z najlepszych

[150] Sutherland J., Schwaber K., *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*, www.scrumtraininginstitute.com.

specjalistów – ma samodzielnie decydować o sposobach realizacji zadań i samoorganizować swoją pracę. Taki podział odpowiedzialności w zespole stanowi znaczną zmianę i wyzwanie dla dotychczasowych stylów zarządzania w wielu organizacjach.

Proces SCRUMa

Proces realizacji projektu zgodnie z metodyką SCRUM ma charakter iteracyjny, czyli oparty na cyklach krótkich przejść, mających na celu dostarczenie określonych dla nich produktów.

Pojedynczy cykl pracy, czyli iteracja, w metodyce SCRUM nosi nazwę sprintu. Określenie to odnosi się, podobnie jak nazwa metodyki, do rugby, gdzie w kolejnych partiach gry zespół przebiega z piłką określony odcinek boiska. Długość trwania pojedynczego sprintu zależy od decyzji zespołu. W zależności od specyfiki projektu przyjąć można okresy od jednego do czterech tygodni, ale dłużej niż miesiąc. Raz ustalona długość sprintu pozostaje niezmienna. W momencie, gdy sprint zbliża się ku końcowi, należy go zakończyć i rozpocząć nowy – niedopuszczalne jest wydłużanie jego czasu, nawet jeśli przewidziana praca nie została wykonana w całości.

Kolejne iteracje realizowane są jedna po drugiej, bez okresów przerwy. Ma to za zadanie wprowadzić określoną dynamikę pracy oraz równe, jednostajne i jednocześnie w miarę szybkie tempo pracy całego zespołu.

Wykaz prac produktu

Pierwszy krok realizacji projektu zgodnie z metodyką SCRUM należy do właściciela produktu. Jego zadaniem jest stworzenie wizji produktu, która następnie przybiera formę listy, składającej się z uporządkowanych zgodnie z priorytetami elementów funkcjonalności tworzonego oprogramowania. Lista ta nosi nazwę wykazu prac produktu (ang. product backlog). Wykaz prac produktu jest jednym dokumentem przedstawiającym jeden, definitywny opis „wszystkiego, co może być kiedykolwiek zrobione przez zespół, w kolejności ważności"[151].

Elementami umieszczanymi w wykazie będą przede wszystkim nowe funkcjonalności oprogramowania, cele związane z poprawą wydajności systemu, prace rozwojowe i pomysły na innowacje, jak również prace nad rozwiązaniem znanych usterek. Metodyka nie zakłada jednego sposobu opisu funkcjonalności systemu. Mogą być one opisywane w dowolny sposób przyjęty w projekcie: w postaci równoważników

[151] Deemer P., Benefield G., Larman C., Vodde B., *SCRUM Primer: An Introduction to Agile Project Management with Scrum*, ver. 1.2, 2010, http://scrumtraininginstitute.com.

zmian, tzw. historyjek, życzeń użytkownika itp. Autorzy metodyki sugerują, aby pracochłonność jednej pozycji w wykazie nie przekraczała 10 dni pracy programistów[152]. Przykład wykazu prac produktu przedstawiony jest w poniższej tabeli:

Tab. 27. Wykaz prac produktu

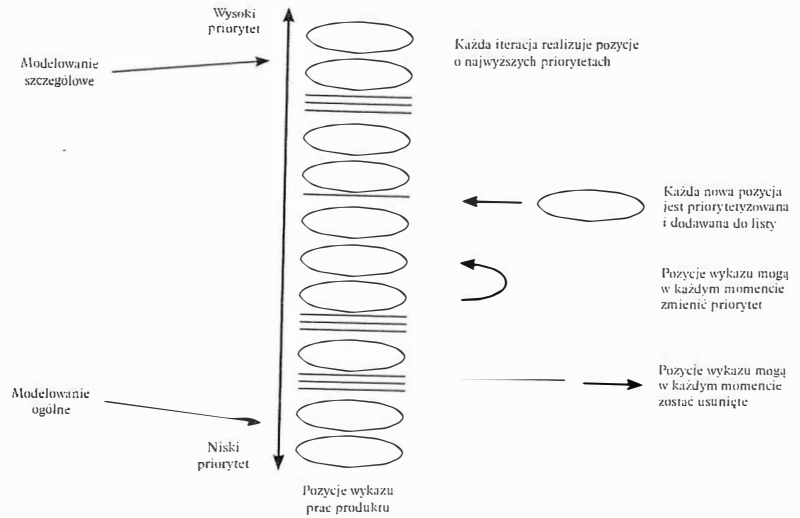
Pozycja wykazu	Specyfikacja (link wiki)	Priorytet	Szacowana wartość	Wstępny szacunek pracochłonności	Aktualne szacunki pracy pozostałej do wykonania					
					1	2	3	4	5	6
Jako kupujący chcę móc umieścić książkę w koszyku zakupów	...	1	7	5						
Jako kupujący chcę móc usunąć książkę z koszyka zakupów	...	2	6	2						
Podnieść wskaźniki wydajności przetwarzania transakcji	...	3	6	13						
Rozpoznać możliwości przyspieszenia walidacji kart płatniczych	...	4	6	20						
Zaktualizować wszystkie serwery do Apache 2.2.3	...	5	5	13						
Zdiagnozować i naprawić błędy skryptu realizacji zamówienia	...	6	2	3						
Jako kupujący chcę stworzyć i zapisać listę zakupów	...	7	7	40						
Jako kupujący chcę móc dodawać i odejmować pozycje z listy zakupów	...	8	4	20						

Źródło: Deemer P., Benefield G., Larman C., Vodde B., *SCRUM Primer: An Introduction to Agile Project Management with Scrum*, ver. 1.2, 2010, http://scrumtraininginstitute.com.

Osobą odpowiedzialną za określenie priorytetów pozycji wykazu prac produktu jest właściciel produktu, który decyzję tę podejmuje, uwzględniając oczekiwania interesariuszy i zdanie zespołu. Metodyka nie precyzuje sposobu, w jaki właściciel produktu powinien to zrobić, jednak zazwyczaj uwzględnia się atrybuty takie jak m.in. złożoność funkcjonalności i jej pracochłonność (ang. effort estimate), wartość biznesowa (ang. business value estimate) oraz poziom ryzyka (ang. risk estimate).

[152] Sutherland J., *A Brief Introduction to SCRUM*, Scrum Alliance 2006.

Rys. 49. Dynamika wykazu prac produktu



Źródło: Ambler Scott, Rethinking the Role of Business Analysts: Towards Agile Business Analysts? http://www.agilemodeling.com.

Jak podkreślają sami autorzy, nieprawdą jest, jakoby SCRUM odrzucał szczegółową pisemną specyfikację funkcjonalności oprogramowania. W rzeczywistości decyzja o stopniu sformalizowania i szczegółowości opisu należy do właściciela produktu i zespołu. Najczęściej elementy o wysokim priorytecie, bardzo dobrze znane i wdrażane w niedługiej przyszłości będą opisane w sposób bardziej dokładny niż funkcjonalności o marginalnym znaczeniu.

W trakcie trwania projektu wykaz prac produktu podlega ciągłej ewolucji, zmieniając się w czasie pod wpływem nowych wymagań, modyfikacji już zapisanych oraz usunięcia tych wykonanych. Zmianie mogą ulegać wszystkie elementy wykazu.

Spotkanie planowania sprintu

Definicja produktu zapisana w postaci wykazu prac produktu stanowi punkt wyjścia do realizacji pierwszej iteracji projektu, czyli pierwszego sprintu.

Na początku pierwszego oraz każdego kolejnego sprintu odbywa się spotkanie planowania sprintu (ang. sprint planning meeting). Jego celem jest opracowanie szczegółowego planu dla bieżącej iteracji.

Spotkanie planowania sprintu trwa zazwyczaj jeden pełny dzień i składa się z dwóch części, po 4 godziny każda. W trakcie pierwszej części właściciel produktu

wraz z zespołem i przy wsparciu ze strony szefa scruma dokonuje przeglądu wizji, mapy drogowej, planu wydań oraz wykazu prac produktu. Zespół powinien zapoznać się z opisami pozycji wykazu, którymi w pierwszej kolejności zainteresowany jest właściciel produktu oraz wyrazić swoją opinię na temat dokładności szacunków ich pracochłonności.

Następnie zespół dokonuje wyboru pozycji wykazu, które będą podlegały opracowaniu w danym sprincie. Zespół „zdejmuje" elementy z góry listy priorytetów i podejmuje się ich realizacji w trakcie trwania jednej iteracji (czyli zazwyczaj czterech tygodni roboczych). Decyzje te wiążą się zawsze ze zobowiązaniem do wykonania w pełni działającej i dającej się zademonstrować funkcjonalności systemu. Ilość pracy podjętej przez zespół będzie zależała od wielkości zespołu, dostępnych roboczogodzin oraz poziomu produktywności zespołu.

Zjawisko odpowiedzialnego „zasysania" pracy, a nie „wpychania" jej do zespołu przez kierownictwo, co podkreślają autorzy metodyki, działa zdecydowanie sprzyjająco produktywności i jest jednym z kluczowych komponentów szczupłego rozwoju produktów (ang. lean product development)[153].

Po podjęciu decyzji o elementach wchodzących w skład sprintu rozpoczyna się część druga spotkania, polegająca na szczegółowym zaplanowaniu zadań i czasochłonności pracy do wykonania. Dzieje się to za pomocą dekompozycji funkcjonalności wyszczególnionych w wykazie prac produktu na zadania cząstkowe (ang. sprint tasks), konieczne do ich wdrożenia. Podział ten powinien doprowadzić do sytuacji, w której zidentyfikowane zadania nie będą trwały więcej niż 2 dni / 16 godzin roboczych. Opracowana lista zadań na najbliższą iterację nosi nazwę wykazu prac sprintu (ang. sprint backlog). Na moment spisania lista wykaz prac sprintu nie koniecznie musi być kompletna, gdyż plan konstruowany jest przy danym (niepełnym) stanie wiedzy zespołu. Lista ta z pewnością będzie aktualizowana w miarę upływu czasu trwania danej iteracji.

Jeśli szacunki pracochłonności poczynione przez zespół różnią się w zasadniczy sposób od szacunków w dyspozycji właściciela produktu, zespół może negocjować realną ilość pracy do wykonania.

Bardzo często do wizualizacji wykazu prac sprintu wykorzystuje się tablice do pisania lub tablice korkowe, na które nanosi się żółte karteczki, symbolizujące poszczególne zadania i funkcjonalności realizowane w trakcie iteracji. Zadania te dzieli się na „jeszcze nie rozpoczęte", „w trakcie realizacji" oraz „zakończone".

[153] Sutherland J., *A Brief Introduction to SCRUM*, Scrum Alliance 2006.

Tab. 28. Wykaz prac sprintu

Pozycja wykazu prac produktu	Zadanie sprintu	Ochotnik	Wstępny szacunek pracochłonności	Aktualne szacunki pracy pozostałej do wykonania					
				1	2	3	4	5	6
Jako kupujący chcę móc usunąć książkę z koszyka zakupów	Zmodyfikować bazę danych		5						
	Opracować stronę (UI)		8						
	Opracować stronę (Javascript)		13						
	Przygotować automatyczne testy akceptacji		13						
	Zaktualizować stronę pomocy dla kupującego		3						
	...								
Podnieść wskaźniki wydajności przetwarzania transakcji	Zintegrować kod DCP i przeprowadzić testy		5						
	Zmienić kod DCP i czytnik do wykorzystania pRank http API		8						
	...								

Źródło: Deemer P., Benefield G., Larman C., Vodde B., *SCRUM Primer: An Introduction to Agile Project Management with Scrum*, ver. 1.2, 2010, http://scrumtraininginstitute.com.

Bardzo ważną zasadą metodyki SCRUM jest warunek niezmienności listy funkcjonalności przyjętej do realizacji w danym sprincie. Oznacza to, że gdy zespół i właściciel produktu raz osiągną porozumienie co do zakresu prac do wykonania, żadna ze stron nie może go zmienić. Zapobiega to sytuacjom, w których kierownictwo po tygodniu od spotkania próbuje wymóc na zespole dodatkową pracę lub z dużą częstotliwością zmieniać priorytety zadań, zaś zespół motywuje do dotrzymania obietnic. Z drugiej jednak strony, wymaga zarówno od jednej, jak i drugiej strony odpowiedzialności za podjęte zobowiązania. Jeżeli jednak sytuacja zmieni się drastycznie, także interwencja będzie rzeczywiście konieczna; możliwe jest zaproponowanie przerwania aktualnie toczącego się sprintu i ponownej organizacji spotkania planowania sprintu.

Scrum powszedni

Od momentu rozpoczęcia sprintu zespół pracuje na bieżąco nad realizacją przyjętych zobowiązań. Zgodnie z metodyką SCRUM w trakcie sprintu odbywają się dodatkowo codzienne zebrania (ang. daily scrum meetings), inaczej zwane również „spotkaniami na dzień dobry" lub „scrumem powszednim". Zebrania te odbywają się codziennie o tej samej porze, najlepiej zanim rozpocznie się dzień pracy, prowadzone są przez sze-

fa scruma i mają charakter „synchronizacji" i bieżącej wymiany informacji na temat codziennej pracy, wykonanych postępów oraz zauważonych trudności. Spotkania te są z założenia krótkie i nie powinny trwać więcej niż kwadrans, dlatego najczęściej przeprowadza się je na stojąco, stąd również ich inna nazwa ang. stand-up meetings. Wśród członków zespołu, którzy jako jedyni mogą zabierać głos, obecność jest obowiązkowa. Inni zainteresowani mogą być obecni, ale bez możliwości wypowiadania się.

Podczas samego zebrania każdy członek zespołu kolejno zabiera głos i przedstawia pozostałym członkom zespołu odpowiedzi na trzy stałe pytania:

- Co udało mi się wykonać od ostatniego spotkania (czyli od wczoraj)?
- Co zamierzam zrobić do kolejnego spotkania (czyli do jutra)?
- Czy pojawiły się jakieś trudności, przeszkody w mojej pracy?

Udzielenie odpowiedzi na te pytania ma poinformować innych o stanie prac i pomóc im koordynować ich przebieg. Wszelkie dalsze dyskusje i wątpliwości mogą być rozstrzygnięte na spotkaniach odbywających się po (!) zebraniu codziennym, lecz nie w jego trakcie. Uczestnictwo w takim zebraniu nie jest już jednak obowiązkowe.

Wykres spalania

Metodyka SCRUM do monitorowania pracy zespołu podczas sprintu posługuje się tzw. wykresem spalania lub inaczej wykresem malejącym (ang. sprint burndown chart).

Zadaniem wykresu jest skupienie uwagi zespołu na celu – zrealizowaniu wszystkich zadań sprintu w zamierzonym czasie – oraz pokazanie postępu w realizacji tego celu. Postęp ten nie jest jednak mierzony w odniesieniu do czasu, jaki upłynął, ale w stosunku do czasu, jaki pozostał.

Podczas każdego spotkania, codziennie, każdy z członków zespołu dokonuje aktualizacji szacunku czasu pracy potrzebnego do zakończenia zadań z wykazu prac sprintu. Suma szacunków pokazuje skumulowaną ilość pracy (w roboczogodzinach), pozostałą do wykonania podczas danej iteracji. W trakcie upływu czasu wartość ta powinna maleć, aż osiągnie zero na zakończenie sprintu (lub jeśli to możliwe wcześniej). Informacje o postępach prac zapisywane są w wykazie prac sprintu, jednakże bardzo często stosuje się ich graficzną prezentację w postaci wykresu.

W idealnej sytuacji linia wykresu skierowana jest do dołu. Jednakże w praktyce, mogą zdarzyć się przypadki, gdy pomimo wykonanej pracy ilość pracy pozostałej jeszcze wzrośnie. Może wystąpić to m.in. w przypadku gdy:[154]

[154] Sutherland J., *A Brief Introduction to SCRUM*, Scrum Alliance 2006.

Tab. 29. Codzienna aktualizacja wykazu prac sprintu

Pozycja wykazu prac produktu	Zadanie sprintu	Ochotnik	Wstępny szacunek pracochłonności	Aktualne szacunki pracy pozostałej do wykonania					
				1	2	3	4	5	6
Jako kupujący chcę móc usunąć książkę z koszyka zakupów	Zmodyfikować bazę danych	Adam	5						
	Opracować stronę (UI)	Tomasz	8						
	Opracować stronę (Javascript)	Beata	13						
	Przygotować automatyczne testy akceptacji	Piotr i Ewa	13						
	Zaktualizować stronę pomocy dla kupującego	Beata	3						
	...								
Podnieść wskaźniki wydajności przetwarzania transakcji	Zintegrować kod DCP i przeprowadzić testy		5						
	Zmienić kod DCP i czytnik do wykorzystania pRank http API		8						
	...	...							
w sumie roboczogodzin:			50	49	48	44	43	34	

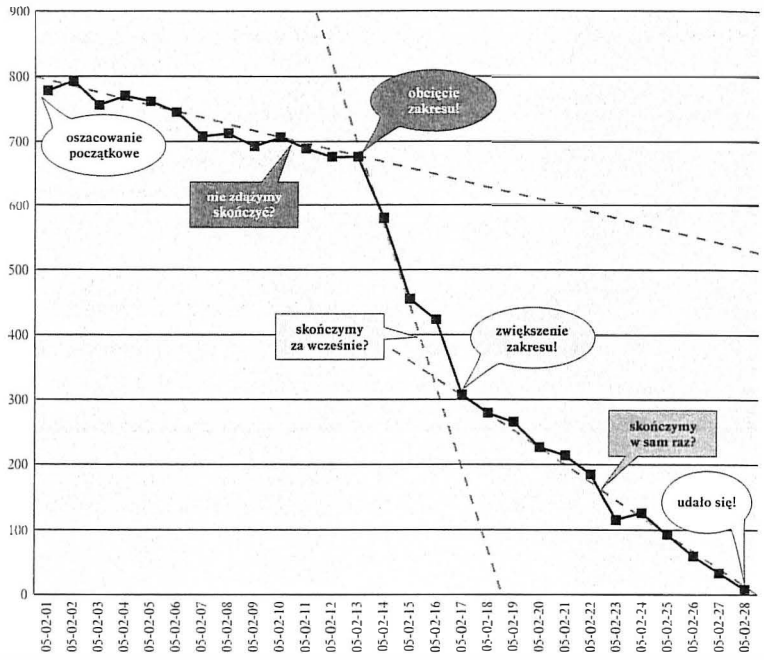
Źródło: Deemer P., Benefield G., Larman C., Vodde B., *SCRUM Primer: An Introduction to Agile Project Management with Scrum*, ver. 1.2, 2010, http://scrumtraininginstitute.com.

- w trakcie realizacji funkcjonalności zespół zdobywa większą wiedzę o faktycznej złożoności zadania i może podjąć decyzję o dodaniu niezbędnych czynności do wykazu prac sprintu;
- rozpoznane wady i defekty mogą wymagać zajęcia się nimi w postaci osobnych, dodatkowych zadań;
- zaistnieje możliwość doprecyzowania celów i wprowadzenia we współpracy z właścicielem produktu drobnych zmian, zwiększających wartość dla klienta, ale nie powodujących wydłużenia czasu sprintu.

Spotkanie przeglądu sprintu i spotkanie retrospektywne sprintu

Zgodnie z metodyką SCRUM czas przeznaczony na jeden sprint jest nieprzekraczalny. Bez względu na stan prac sprint kończy się zgodnie z przyjętą datą. W początkowych iteracjach zespoły, które dopiero rozpoczynają swoją przygodę ze SCRUMem, mogą mieć trudności w oszacowaniu ilości pracy, którą mogą podjąć podczas jednej iteracji. W takim przypadku może zdarzyć się, że pierwsze iteracje będą niedoszacowane lub przeszacowane pod kątem pracochłonności zadań. Jednakże po pewnym czasie dopasowania i rozpoznania swoich możliwości zespół powinien móc kończyć zgodnie z rytmem SCRUM'a.

Rys. 50. Monitorowanie projektu z wykorzystaniem wykresu spalania



Oś X: oszacowanie dokonane w dniu x
Oś Y: liczba pozostałych roboczogodzin wg stanu na dzień x

Źródło: Milewski J., *Projektowy młyn*, Computerworld, nr 29/2005.

Zakończenie sprintu związane jest z organizacją dwóch czterogodzinnych spotkań, które odbywają się najczęściej jednego dnia, są to: spotkanie przeglądu sprintu (ang. sprint review meeting) oraz spotkanie retrospektywne sprintu (ang. sprint retrospective meeting).

W trakcie spotkania przeglądu sprintu zespół wraz z właścicielem produktu oraz przy wsparciu szefa scruma dokonują przeglądu wszystkiego tego, co wydarzyło się w trakcie ostatniej iteracji i dotyczyło opracowywanego produktu. Zespół przedstawia wyniki swojej pracy oraz demonstruje działające funkcjonalności produktu, zaś właściciel produktu ma za zadanie zapoznać się z rozwiniętym produktem i przekazać zespołowi aktualne informacje na temat wizji produktu i stanu rynku. Istotnym przesłaniem spotkania jest zapewnienie wymiany informacji i konwersacji pomiędzy stronami tak, aby uczestnicy spotkania nauczyli się od siebie jak najwięcej i mogli lepiej zrozumieć aktualny stan i perspektywy realizowanego projektu.

Pozycje z wykazu prac produktu, które zostały wykonane i odebrane, w całości zostają wykreślone. Pozycje, które wymagają dalszej pracy, nie mogą być prezentowane na spotkaniu i wracają do zestawienia, gdzie będą oczekiwać na zaktualizowane priorytety od właściciela produktu.

W spotkaniu oprócz wspomnianych powyżej uczestników mogą brać udział także inne osoby zainteresowane projektem i swobodnie dzielić się swoimi wątpliwościami i propozycjami.

Spotkanie retrospektywne sprintu odbywa się zaraz po zakończeniu przeglądu sprintu. Jego celem jest zapewnienie ciągłego doskonalenia procesu i uczenia się wszystkich uczestników scruma tak, aby kolejne iteracje toczyły się lepiej i sprawniej od poprzednich. Jest to bardzo ważny element metodyki i nie powinien być pomijany lub marginalizowany. W trakcie spotkania retrospektywnego każdy z uczestników wypowiada się na temat przebiegu ostatniego sprintu, odnosząc się do tego, co robione było dobrze, do tego, co robione było źle, oraz jakie zmiany należałoby wprowadzić od kolejnego sprintu. Spotkanie takie może poprowadzić szef scruma, ale dla zachowania neutralności sugeruje się zaproszenie zewnętrznego prowadzącego. Właściciel produktu może w nim uczestniczyć, lecz jego obecność nie jest obowiązkowa.

Kolejna iteracja – nowy sprint

W momencie zakończenia sprintu zadaniem właściciela produktu staje się zaktualizowanie wykazu prac produktu tak, aby odzwierciedlał on nie tylko postęp jego realizacji, ale również informacje i zmiany w otoczeniu bliższym i dalszym projektu (np. oczekiwania rynku/użytkowników wobec funkcjonalności produktu). Mogą pojawić się nowe pozycje, z pewnością zmienią się ich priorytety, być może znane będą dokładniejsze szacunki ich pracochłonności, część zupełnie straci swoje uzasadnienie i zostanie usunięta z wykazu itd.

Podobnie jak w przypadku monitorowania stanu sprintu przez zespół z wykorzystaniem wykresu spalania sprintu, właściciel produktu może posłużyć się analogicznym wykresem do monitorowania całego projektu i stworzyć wykres spalania produktu (lub jego pojedynczego wydania).

Zgodnie z metodyką SCRUM nowy sprint powinien rozpocząć się bezzwłocznie po zakończeniu ostatniego. Najlepiej spotkanie planujące sprint zaplanować już na kolejny dzień roboczy po spotkaniu przeglądu sprintu. Silnie podkreślaną cechą metodyki SCRUM jest jej szybkie tempo i realizacja kolejnych cykli iteracji zgodnie ze stałą częstotliwością bez zbędnych opóźnień.

Podsumowanie

Źródłem powstania metodyki SCRUM były bez wątpienia projekty opracowania nowych produktów w branży informatycznej. Jednakże jego zastosowanie nie ogranicza się wyłącznie do oprogramowania i systemów informatycznych. Idea koncentracji uwagi na produkcie i bliskiej, dynamicznej współpracy w ramach zespołu, bieżącej wymiany informacji oraz rozwoju produktu opartego na podejściu przyrostowym, znajduje szerokie zastosowanie w wielu innych branżach i projektach, szczególnie tych o wysokim stopniu złożoności i innowacji. SCRUM może znaleźć zastosowanie w zarządzaniu oprogramowaniem i rozwojem sprzętu, w dostarczaniu wsparcia, działalności reklamowej i marketingu i wielu innych[155]. Na stronie Scrum Alliance przeczytać można opisy przypadków zastosowania metodyki m.in. w operacjach logistycznych marynarki brytyjskiej oraz zarządzania portfelem działalności dużego banku[156].

Od wielu lat metodyka SCRUM przebojem zdobywa obszar zarządzania projektami informatycznymi i obecnie jest jedną z dwóch (obok XP) najpopularniejszych metodyk zwinnych. Wśród firm stosującej SCRUM w tworzeniu i rozwoju swoich produktów można wymienić m.in.: Microsoft, Google, Yahoo!, British Telecom, Siemens, Adobe, Lockheed Martin, Motorola, SAP, Cisco, GE Medical i wiele innych.

Doświadczenia Yahoo! były przedmiotem badania przeprowadzonego od 2005 do 2007 roku przez Gabrielle Benefield, Senior Director of Agile Development[157]. Wśród zaobserwowanych zmian dostrzegła ona m.in. wzrost produktywności zespołów średnio o 37%, przy chęci dobrowolnego kontynuowania stosowania metody przez 86% ankietowanych. W innym artykule przedstawione są wyniki firmy informatycznej znajdującej się na piątym poziomie CMMI, w której po zastosowaniu metodyki SCRUM koszt projektów informatycznych spadł o połowę, a poziom błędów o 40%, przy zachowaniu poziomu CMMI[158].

[155] Hundermark P., *Do Better Scrum 2nd ed.*, Scrum Sense, Nov. 2009.

[156] Scrum Alliance, http://scrumalliance.com.

[157] Benefield G., *Rolling out Agile at Large Enterprise* [w] Sutherland J., Schwaber K., *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*, www.scrumtraininginstitute.com.

[158] Sutherland J., Jacobson C., Johnson K., *Scrum and CMMI Level 5: A Magic Potion for Code Warriors!*, Agile 2007, Washington 2007.

Wśród innych zalet metodyki SCRUM wskazać można m.in.:[159]

- wysoki poziom kontroli nad przebiegiem projektu za pomocą krótkich iteracji;
- wczesne uzyskanie istotnych elementów projektu dających wymierne korzyści biznesowe;
- przejrzystość planowania i wdrożenia oprogramowania;
- wzrost szybkości opracowywania nowych produktów;
- powiązanie celów indywidualnych z celami organizacji;
- tworzenie kultury opartej na rezultatach;
- wspieranie tworzenia wartości dla interesariuszy;
- zapewnienie stabilnych i trwałych kanałów komunikacyjnych na wszystkich poziomach;
- umożliwienie rozwoju osobistego i poniesienie jakości życia.

SCRUM posiada niewątpliwie również pewne wady i ograniczenia. Przede wszystkim, jak podkreślają to sami autorzy SCRUM, jest metodyką ramową i nie dostarcza zespołom szczegółowego zestawu praktyk do zarządzania projektem. SCRUM może ujawnić problemy i nieprawidłowości w sposobie prowadzenia projektu, ale nie dostarczy ich jednoznacznych rozwiązań – wymaga w tym zakresie dojrzałości i wiedzy od uczestników projektów[160]. Jak stwierdza G. Asproni „należy uwzględnić fakt, iż nie każdy się do Scrum nadaje [...] u najbardziej introwertycznych programistów konieczność tak ścisłej współpracy może wywołać poczucie dyskomfortu. Niektórzy programiści czują się też jedynymi właścicielami kodu i nie lubią, gdy ktoś im ten kod modyfikuje lub choćby ogląda"[161].

Często poruszany jest również temat skalowalności metody oraz stosowania jej w dużych rozproszonych geograficznie zespołach, jak również w zespołach, które są w słabym stopniu zintegrowane. Zdecydowanie utrudnia to częsty bezpośredni kontakt, który jest jednym z kluczowych czynników sukcesu tej metody[162].

Wątpliwości budzi również zastosowanie metodyki SCRUM w projektach o znaczeniu krytycznym, jednakże dotychczasowe doświadczenia wskazują, iż w takim przypadku nic nie stoi na przeszkodzie, aby proces SCRUM uzupełnić o bardziej

[159] Sutherland J., Schwaber K., *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*, www.scrumtraininginstitute.com.

[160] Deemer P., Benefield G., Larman C., Vodde B., *SCRUM Primer: An Introduction to Agile Project Management with Scrum*, ver. 1.2, 2010, http://scrumtraininginstitute.com.

[161] Asproni G., *Wstęp do SCRUM*, Software Developer's Journal 6/2006, www.sdjournal.org.

[162] Milewski J., *Projektowy młyn*, „Computerworld", nr 29/2005.

szczegółowe i rozbudowane procedury kontroli jakości i zawansowane mechanizmy testowania[163].

Spis literatury

1. Asproni G., *Wstęp do SCRUM*, Software Developer's Journal 6/2006, www.sdjournal.org.
2. Benefield G., *Rolling out Agile at large Enterprise* [w] Sutherland J., Schwaber K., *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*, www.scrumtraininginstitute.com.
3. Brooks F.P., *The Mythical Man Month: Essays on Software Engineering*, Adison-Wesley 1995.
4. Deemer P., Benefield G., Larman C., Vodde B., *SCRUM Primer: An Introduction to Agile Project Management with Scrum*, ver. 1.2, 2010, http://scrumtraininginstitute.com.
5. DeGrace P., Stahl L.H., *Wicked Problems, Righteous Solutions: A Catalogue of Modern Software Engineering Paradigms*, Yourdon Press, 1990.
6. Hundermark P., *Do Better Scrum 2nd ed.*, Scrum Sense, Nov. 2009.
7. Milewski J., *Projektowy młyn*, „Computerworld", nr 29/2005.
8. Schwaber K., *Agile Software Development with SCRUM*, Microsoft Press 2004.
9. Schwaber K., Beedle M., *Agile Software Development with SCRUM*, Prentice Hall 2002.
10. Scrum Alliance, http://scrumalliance.com.
11. Sutherland J., *A Brief Introduction to SCRUM*, Scrum Alliance 2006.
12. Sutherland J., *Agile Development: Lessons Learned from the First SCRUM, Cutter Agile Project Management Advisory Service*, Executive Update, Vol. 5, No. 20, www.scrumalliance.org.
13. Sutherland J., Jacobson C., Johnson K., *Scrum and CMMI Level 5: A Magic Potion for Code Warriors!*, Agile 2007, Washington, 2007.
14. Sutherland J., Schwaber K., *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*, www.scrumtraininginstitute.com.
15. Takeuchi H., Nonaka I., *The New New Product Development Game*, Harvard Business Review, Jan/Feb 1986.

[163] Tamże.

Liczne studia przypadków i przykłady zastosowania metodyki SCRUM można znaleźć na stronach internetowych:

- Scrum Alliance, http://scrumalliance.com
- Scrum Training Institute http://scrumtraininginstitute.com

Filmy i prezentacje nt. metodyki dostępne są również w portalu Youtube, zaś na portalach społecznościowych znaleźć można wiele wątków dyskusyjnych oraz grupy praktyków i sympatyków Scruma.