

Jan WEREWKA, Michał TUREK, Tomasz WŁODAREK
AGH, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Automatyki

SYSTEMATYCZNY OPIS METODYKI SCRUM DLA ZESPOŁÓW PROJEKTOWYCH

Streszczenie. Artykuł przedstawia systematyczny opis metodyki *Scrum* – najbardziej popularnej zwinnej metodyki łączącej zarządzanie projektami z wytwarzaniem produktów. Do opisu różnych metodyk tego typu zaproponowano użycie metamodelu. Do opisu metodyk przyjęto zestaw podstawowych konceptów, przy czym w opracowaniu ograniczono się do pięciu podstawowych. Rozwiązanie problemu wiarygodnego wyznaczenia własności metodyk będzie polegać na opracowaniu modeli ontologicznych dla wybranych metodyk, bazując na wspólnym metamodelu.

Dodatkowo opracowanie porządkuje niektóre rozwiązania nie w pełni wyspecyfikowane w Scrum, dokonując jednocześnie ewaluacji ich przydatności w praktyce. Przedstawione rozwiązania są wynikiem prowadzenia prac badawczych, w tym licznych symulacji i eksperymentów z udziałem rzeczywistych zespołów i w konkretnych realizacjach komercyjnych.

Słowa kluczowe: inżynieria oprogramowania, zwinne metodyki wytwarzania oprogramowania, Scrum

SYSTEMATIC DESCRIPTION OF SCRUM FOR PROJECT TEAMS

Summary. The article presents a systematic description of Scrum – the most popular agile methodology connecting project management and product development. In order to describe different methodologies of such type, in this paper there is proposed a use of a meta model, and a set of basic concepts suggested, yet only five of them illustrated. For credible methodologies description, ontology models basing on a common meta model will be developed.

In addition, the article orders a number of solutions unspecified thoroughly in Scrum, evaluating their usefulness in practice at the same time. The presented solutions are the result of research, including numerous simulations and experiments with real teams and in specific commercial ventures.

Keywords: software engineering, agile software development methodology, Scrum

1. Zagadnienie konstrukcji systematycznego opisu metodyk wytwarzania i zarządzania

Metodyki wytwarzania oprogramowania oraz zarządzania projektami są opisywane przeważnie nieformalnie i na różnorodne sposoby. Firmy informatyczne, próbujące wykorzystać metodyki w praktyce, często borykają się z ustaleniem istotnych informacji opisujących metodyki i różnice pomiędzy nimi. Nieprzerwanie są rozwijane metody i narzędzia do wiarygodnego opisu różnego typu systemów. Coraz większe znaczenie zdobywają modele ontologiczne uzupełnione o modele procesów, np. w notacji BPMN (ang. *Business Process Modeling Notation*). Autorzy pracy obrali taki właśnie kierunek [14, 15]. Opracowane w ten sposób modele wymagają weryfikacji pod względem zgodności z pierwowzorem poprzez uzyskanie opinii ekspertów. W niniejszym opracowaniu w wyniku rozważań nad modelem ontologicznym [16] przedstawiono opis metodyki Scrum po wstępnej weryfikacji z ekspertami dziedzinowymi.

W celu opisu różnych metodyk w obszarze zarządzania projektami i wytwarzania oprogramowania proponuje się użycie metamodelu. Metamodel będzie polegał na utworzeniu języka zdolnego do opisu różnych metodyk. Z jednej strony metamodel będzie służył do ograniczenia sposobu definiowania pojęć reprezentujących różne metodyki, zaś z drugiej strony model musi być na tyle ogólny, by nie ograniczał i nie zmieniał opisywanych metodyk. Zastosowany metamodel będzie polegał na konceptach oraz zbiorze konceptów o pewnej strukturze. Koncept jest jednostką myśli, która może być zdefiniowana lub opisana.

Do opisu metodyk przyjęto zestaw podstawowych konceptów, przy czym w opracowaniu ograniczono się do pięciu podstawowych, są to:

- Uniwersalne zasady (pryncypia). Zasady te są wytycznymi do tych działań w metodykach, które nie zostały dokładnie opisane. Działania takie powinny być zgodne z uniwersalnymi zasadami. Uniwersalne zasady postępowania mogą zależeć między innymi od tego, czym się kierujemy (np. wizji, norm, które uznajemy oraz wyznawanych wartości) oraz od tego, co i jak robimy (np.: misji, postulatów, polityki).
- Artefakty. Wszystko to co powstaje w trakcie wykonania prac projektu (np. wyroby, przedmioty, dokumenty, wzory zachowań).
- Role. Rola oznacza wirtualną osobę, która jest w stanie wykonywać pewien zbiór czynności. Do tworzenia artefaktów są potrzebne osoby wykonujące pewne role.
- Procesy. Proces jest to zespół czynności, w wyniku którego powstają artefakty.
- Zdarzenia. Zjawiska, które występują w trakcie wykonania projektu.

Konstruując opisy metodyk zarządzania projektami, wyszczególniono zdarzenia i procesy. Ważne jest zrozumienie różnicy obu pojęć. Zdarzenia są zdefiniowane poprzez relacje czasowe (czas wystąpienia, trwania itd.) oraz własności (uczestnicy, miejsce, przedmiot

uwagi) i nie wytwarzają żadnych artefaktów. Procesy transformują wejścia w wyjścia i są uruchamiane podczas zdarzeń lub w wyniku pojawienia się zdarzenia.

Zaproponowany metamodel zastosowano do opisu części metodyki Scrum, dotyczącej spojrzenia na metodykę głównie z punktu widzenia zespołu i roli zwanej Scrum Masterem. Zaproponowany opis nie zmienia oraz nie poprawia metodyki. Opis ten ma za zadanie w sposób możliwie wiernie opisać metodykę, nie powodując jej zmian. Przedstawiony opis zawiera procesy obejmujące około 90% prac zespołu.

Zakłada się, że pozostała część metodyki zostanie przedstawiona w osobnym opracowaniu. Opracowanie takie będzie dotyczyło głównie spojrzenia na metodykę z punktu widzenia właściciela produktu i kierownictwa firmy. W opisie tym uwzględną się m.in. wizję produktu, wydania produktu, usuwanie blokad oraz pozostałe 10% prac zespołu przeznaczonych na wspomaganie prac właściciela produktu, związanych z estymacją, definicją kryteriów gotowości i wykonania itp.

Metodyki zarządzania projektami dzielimy na dwie podstawowe kategorie: klasyczne zorientowane na plan i zwinne zorientowane na zmiany produktu. Metodyki zwinne, określane w literaturze anglojęzycznej terminem *agile* (zwinny, sprawny, rzutki, zwrotny), w ostatnich kilku latach zdobywają coraz większą popularność przy produkcji oprogramowania różnej skali, w większości obszarów i domen. Znajdują one zastosowanie szczególnie w tworzeniu systemów i realizacji projektów, które są obłożone ryzykiem lub niewiadomymi. Podstawowymi przesłaniami stojącymi za wykorzystaniem metodyk zwinnych jest zwiększona podatność na dokonywanie zmian (wymagań, technologii) w projekcie, lepsza kontrola ryzyka, a także osiągnięcie w warunkach zmian wysokiej zbieżności wytworzonego produktu z oczekiwaniami odbiorcy oprogramowania, przede wszystkim poprzez wczesne (w cyklu życia projektu) i nieprzerwane dostarczanie wysokiej jakości produktu.

W literaturze polskiej czasami utożsamia się metodyki lekkie z metodykami zwinnymi. W tym opracowaniu metodykę lekką rozumie się jako metodykę, z której usunięto wszystko co nadmiarowe. Naszym zdaniem lekkość metodyki nie implikuje jej zwinności. Przykładowo, można opracować lekką metodykę wykonywania pracy w pewnego typu urzędach. Metodyka taka może okazać się jednak mało elastyczna wobec konieczności wprowadzenia zmian w opisywanych przez nią procesach lub w przypadku zastosowania jej w innych instytucjach. Zakładamy jednak, że metodyka zwinna powinna być lekka, wymaga bowiem sprawnego reagowania na zachodzącą w projekcie zmianę. Zbyt duże obciążenie wymogami formalnymi (procesami, procedurami) ogranicza zwinność nie tylko ze względu na nakład pracy potrzebny do osiągnięcia zgodności z regułami metodyki, ale również nakłady dostosowywania metodyki do nowej sytuacji.

2. SCRUM – geneza i rozwój

Podstawowe założenia metodyki wywodzą się z prac prof. Hirotaka Takeuchi i Ikujiro Nonaka będących wynikiem studiów procesów wytwórczych przeprowadzonych w ponadnarodowych koncernach. W swojej publikacji pt. „The New New Product Development”[4], uważanej powszechnie za źródło inspiracji metodyki Scrum, wśród kilku opisywanych typów procesów, autorzy opisali także tzw. podejście holistyczne (całościowe), najbardziej ich zdaniem efektywne w wytwarzaniu innowacyjnych produktów. Zdaniem autorów dopiero ujęcie holistyczne umożliwia uzyskanie oczekiwanej produktywności i elastyczności procesu. Planowanie w takim procesie z założenia jest planowaniem adaptacyjnym, a sam proces jest procesem empirycznym.

Zainspirowani sportem – w opisie procesu holistycznego przytaczają strategię gry rugby, w szczególności formację młyna (ang. *Scrum*), jako dobrą metaforę odzwierciedlającą zestaw reguł obowiązujących zespoły realizujące podejście holistyczne –Takeuchi i Nonaka wyszczególnili sześć cech, którymi ich zdaniem powinien charakteryzować się proces holistyczny. Są to kolejno:

- Wbudowana niestabilność, rozumiana jako duża autonomia działania zespołu, którego zadaniem jest opracowanie innowacyjnego produktu o strategicznym znaczeniu, co za tym idzie brak jest szczegółowych wytycznych co do postaci końcowej produktu. Nowy, innowacyjny produkt oznacza nowy proces, zatem wiedza z poprzednich projektów nie może zostać bezpośrednio zastosowana. Zespołowi, osobom najbardziej kompetentnym w kwestii doboru narzędzi i procedur, jest pozostawiana swoboda w wyborze szczegółowych rozwiązań oraz sposobu, w jaki cel strategiczny zostanie osiągnięty. Naczelne kierownictwo występuje w roli katalizatora, pełniąc rolę sponsora prac.
- Samoorganizacja zespołów, rozumiana jako rekonstrukcja struktury zespołu w kierunku zapewnienia jego autonomii, wypracowania kultury pracy zachęcającej do przekraczania ograniczeń i poszukiwania nowych wyzwań, oraz wystąpienia elementu wzajemnej inspiracji pomiędzy członkami zespołu reprezentującymi odrębne doświadczenie, kwalifikacje i sposoby myślenia. Członkowie zespołu są zachęceni do podejmowania inicjatywy, ryzyka i odpowiedzialności za swoje działania, współpracy, koncentracji na rozwiązywaniu problemów i rozwijania zróżnicowanych umiejętności. Członkowie zespołu udzielają się we wszystkich niemal obszarach działalności, a procesy są formułowane przez nich tam, gdzie rzeczywiście są potrzebne. Proces wytwórczy ulega zatem ciągłej rekonfiguracji – dopasowaniu do rzeczywistych potrzeb.
- Zachodzące na siebie fazy (etapy) rozwoju produktu, pozwalające na ograniczenie liczby przekazów produktu pomiędzy grupami czy stanowiskami. Podejście takie skutkuje skróceniem całkowitego czasu realizacji projektu o około 30%. Możliwe jest to poprzez

samoistne dopasowanie współpracujących ze sobą członków zespołu, a także podwykonawców i odbiorców do wzajemnych potrzeb. Takie zsynchronizowanie wzajemnych interesów zostało nazwane rytmem (pulem) i określone zostało jako cecha charakteryzująca dany zespół projektowy, i tym samym nie może to stanowić stałego elementu procesu wytwórczego, a jedynie jego ogólną wytyczną. Dodatkowo, aby taki puls mógł się pojawić, istotne jest zapewnienie stabilności składu osobowego zespołu.

- Nabywanie wszechstronnej wiedzy (multilearning), rozumiane jako kultywowanie procesu ciągłego nabywania nowych umiejętności i wiedzy przez członków zespołu, a także wykraczanie w tym procesie poza dotychczasową domenę ekspertyzy. Proces ten powinien zaistnieć przede wszystkim dzięki podejmowaniu kolejnych prób i wyciąganiu wniosków z popełnionych błędów, ale także dzięki wymianie wiedzy pomiędzy członkami multifunkcyjnego zespołu. Bezpośredni kontakt ze specjalistami z różnych dziedzin powoduje, iż wszyscy członkowie zespołu nabywają szerokiej wiedzy wykraczającej poza ich początkowe kwalifikacje, dzięki czemu zespół jest w stanie pokonywać bariery i przełamywać status quo (eksperci, często kierując się rutyną, podążają utartymi ścieżkami, podczas gdy nowicjusze, nie wiedząc, że „tak się nie da”, osiągają cel). W procesie pozostawiono zatem znaczący margines na błędy; zarezerwowano czas na ich naprawę oraz samodoskonalenie.
- Subtelna kontrola, pojęcie to opisuje proces kontroli, który z jednej strony nie ogranicza kreatywności i spontaniczności jednostki, a z drugiej zabezpiecza zespół, i co za tym idzie całą organizację przed popadnięciem w chaos. Element kontroli jest obecny w procesie holistycznym zarówno pod postacią punktów kontrolnych wyznaczanych przez naczelne kierownictwo, sprawdzające zgodność postępów prac z wyznaczonym celem strategicznym, jak również pod postacią presji utrzymywania wysokich standardów pracy płynącej ze strony współpracowników. Takeuchi i Nonaka zwracają uwagę na fakt, iż celem subtelnej kontroli jest wytworzenie kultury ciągłego doskonalenia, a nie wytykanie pomyłek. Pomocny w tym procesie jest wspólny dla wszystkich członków zespołu zestaw wartości.
- Niezakłócony transfer wiedzy, rozumiany jako swobodny przepływ informacji pomiędzy zainteresowanymi stronami, ale także transfer doświadczenia z jednego zespołu do innych w obrębie danej organizacji. Członkowie pierwotnego zespołu, zakładając nowe komórki (zespoły), przekazują nie tyle wiedzę merytoryczną, ile kulturę pracy. Instytucjonalizacja procesów (przenoszenie rozwiązań z przeszłości do teraźniejszości) może być zagrożeniem dla uczenia się organizacji i przełamywania status quo [6].

Podejście Takeuchiego i Nonaki jednoznacznie wskazuje na korzyści zastosowania podejścia holistycznego w branżach wymagających innowacyjności i charakteryzujących się dużą zmiennością oczekiwań odbiorców – a takimi przecież cechami niewątpliwie odznacza

się branża IT. Wśród potencjalnych ograniczeń Takeuchi i Nonaka wskazują natomiast na trudności zastosowania tego podejścia w wytwarzaniu rozległych systemów, a także w silnie scentralizowanych organizacjach, których rozwój produktu jest realizowany pod dyktando wąskiego grona ekspertów. Zwracają oni także uwagę na trudności związane z restrukturyzacją przedsiębiorstw podczas przechodzenia na nowy sposób działania.

Rozważania Takeuchiego i Nonaki, nawiązujące do formacji młyna jako metafory zespołu projektowego realizującego podejście holistyczne, były bezpośrednią inspiracją dla Kena Schwabera, Jeffa Sutherlanda i Mike'a Beedle'a do opracowania metodyki postępowania przy realizacji projektów informatycznych [8]. W metodyce Scrum zawarty jest wiele nawiązań do wyszczególnionych powyżej postulatów Manifestu Agile oraz cech procesu holistycznego. Z formalną definicją metodyki Scrum w zastosowaniu do projektów informatycznych wystąpił Ken Schwaber podczas konferencji OOPSLA w 1995 [1].

W chwili obecnej Scrum wydaje się być najbardziej rozpowszechnioną wśród metodyk zwinnych zarządzania. Scrum skupia się na kwestiach zarządzania projektem oraz organizacji pracy zespołu. Badanie stanu wykorzystania metodyk zwinnych przeprowadzone przez VersionOne Inc. w 2010, w którym udział wzięło blisko pięć tysięcy respondentów, wskazało 58-procentowy udział Scruma i 17-procentowy udział hybrydy Scruma z programowaniem ekstremalnym (XP) [13]. Pozostałe metodyki, do których należą AgileUP, FDD, Lean Development, DSDM, OpenUP, Agile Modeling, Crystal, łącznie nie przekraczają progu 20%. Pozostałą część udziałów stanowią zwinne metodyki autorskie dostosowane i opracowane na potrzeby danego przedsiębiorstwa.

3. Postulaty metodyk zwinnych

Metodyka Scrum nie jest opisana w dokumencie w postaci standardu. Zwykle jak ktoś opracuje pewne rozwiązanie związane z metodyką Scrum, wtedy odwołuje się do wiodących prac w tym obszarze, głównie uznanych autorytetów. Dokonując analiz opracowań zgromadzonych w dużej mierze wokół organizacji popularyzujących metodykę Scrum – Scrum Alliance [11] i Scrum.org [12], autorzy pracy wydzieliли sześć uniwersalnych zasad (rys. 1), które są uznawane w różnych opracowaniach związanych z metodyką Scrum, są to:

- Podejście holistyczne.
- Zasady Manifestu Agile.
- Samoorganizacja zespołu.
- Wartości zespołu.
- Empiryzm procesów metodyki.
- Normatywny charakter metodyki.



Rys. 1. Uniwersalne zasady Scrum

Fig. 1. Universal Scrum rules

Podejście holistyczne. Podejście całościowe zostało przedstawione przez Takeuchi i Nonaka [4, 6], którzy wyszczególnili sześć charakterystycznych cech poprawnego procesu holistycznego. W podejściu holistycznym na proces należy patrzeć jak na układ całościowy, a całości nie da się sprowadzić do sumy składników. Cechy procesu holistycznego zostały opisane dokładnie w poprzednim podrozdziale.

Zasady Manifestu Agile. Manifest Zwinnego Wytwarzania Oprogramowania [3] jest wykładnią postępowania, a zasady metodyki opierają się na jego pryncypiach. Wszyscy twórcy Scruma Ken Schwaber, Jeff Sutherland i Mike Beedle są współautorami tego dokumentu.

W roku 2001 reprezentanci nowego nurtu w sposobie wytwarzania oprogramowania wypracowali kilka uniwersalnych twierdzeń i wytycznych opisujących nowe podejście. Ogłoszono je pod postacią tzw. Manifestu Zwinnego Wytwarzania Oprogramowania (ang. *Manifesto for Agile Software Development*) [3]. Manifest ten, nazywany w skrócie Manifestem Agile, deklaruje, wspólne dla wszystkich metodyk zwinnych, postulaty wartościujące składowe procesu wytwarzania oprogramowania. I tak, w podejściu zwinnym przedkłada się:

- Ludzi i współpracę ponad procesy i narzędzia.
- Działające oprogramowanie ponad obszerną dokumentację.
- Współpracę z klientem ponad negocjowanie kontraktu.
- Reagowanie na zmiany ponad podążanie za planem.

Dla wyjaśnienia w manifeście dodane jest zdanie: „Oznacza to, że wprawdzie doceniamy to, co wymieniono po prawej stronie, jednak bardziej cenimy to, co jest po stronie lewej”.

Manifest Agile uzupełnia zbiór 12 zasad, nazywanych pryncypiami, do których stosowania nawołują autorzy Manifestu:

- Najwyższym priorytetem jest zaspokajanie potrzeb klienta, dokonywane poprzez wczesne i częste dostarczanie wartościowego oprogramowania.
- Zmiany wymagań są przyjmowane z aprobatą, nawet jeśli występują późno w cyklu rozwoju. Procesy zwinne wykorzystują (zaprzegają) zmiany do uzyskania przez klienta przewagi konkurencyjnej.

- Działające oprogramowanie dostarczane jest często, w odstępach od kilku tygodni do kilku miesięcy, z preferencją krótszych okresów.
- Osoby reprezentujące biznes muszą pracować razem z zespołem produkcyjnym – dzień w dzień – przez cały okres realizacji projektu.
- Projekty powstają na bazie motywacji osób biorących w nich udział. Udostępniane są im wszystkie niezbędne zasoby, ponadto osoby te są obdarzane zaufaniem, że ich praca zostanie wykonana.
- Najbardziej skutecznym i wydajnym sposobem przekazywania informacji do i w obrębie w zespole projektowego jest bezpośrednia rozmowa.
- Działające oprogramowanie jest głównym wskaźnikiem postępu.
- Procesy zwinne promują zrównoważone tempo rozwoju oprogramowania. Inwestorzy, zespół projektowy i użytkownicy powinni być w stanie utrzymywać stałe tempo bez końca.
- Ciągłe przywiązywanie wagi do technicznej doskonałości i architektury oprogramowania poprawia zwinność.
- Niezbędna jest prostota rozumiana jako sztuka maksymalizowania pracy, której nie trzeba wykonywać.
- Najlepsze możliwe: architektura, wymagania i projekty powstają w rezultacie pracy samoorganizujących się zespołów.
- W regularnych odstępach czasu zespół zastanawia się, jak zwiększyć swoją efektywność, w efekcie czego dostraja i poprawia odpowiednio swoje zachowanie.

Drugie zdanie punktu 2 może być również przetłumaczone na „Procesy zwinne ujarzmiają zmiany w celu uzyskania przez klienta przewagi konkurencyjnej”. Według autorów zdanie to może być interpretowane jako założenie, że zmiany powinny być ujarzmione. Oznacza to, że nie mogą one być w dowolny sposób przyjmowane. Sugeruje się z jednej strony zwinność podejścia, z drugiej strony opracowanie zasad wprowadzania zmian w sposób kontrolowany.

Przy tak dużym przyzwoleniu na wprowadzanie zmian w projekcie sprawne prowadzenie prac projektowych nie byłoby możliwe w sytuacji, gdyby proces kontroli i adaptacji do zmian obejmowałby jedynie specyfikację wytwarzanego systemu. Dlatego metodyki zwinne na równi z doskonaleniem produktu końcowego stawiają doskonalenie procesu wytwórczego. Takie podejście wymaga ustanowienia częstych punktów kontrolnych, które pozwolą nie tylko oceniać dotychczasowy przebieg przedsięwzięcia, ale także ustalać, jakie elementy procesu wytwórczego należy poddać modyfikacji w kolejnych etapach, tak aby osiągnąć najlepsze możliwe dopasowanie procesu do uwarunkowań budżetowych i terminowych realizowanego przedsięwzięcia.

Samoorganizacja zespołu – zespół Scrum jest z założenia zespołem samoorganizującym się, a więc powinien spełniać postulaty Takeuchiego i Nonaki pod względem autonomii działania, międzyfunkcjonalności, wzajemnej kontroli służącej utrzymaniu wysokiej jakości pracy oraz kultury pracy promującej poszukiwanie nowych rozwiązań, a także przejmowanie współodpowiedzialności za sukces przedsięwzięcia. Co równie istotne, organizacja wykorzystująca tę metodykę powinna zapewnić odpowiednie otoczenie i warunki sprzyjające istnieniu samoorganizacji (por. [6]).

Wartości zespołu Scrum, jako element również nawiązujący do postulatów Takeuchiego i Nonaki (subtelna kontrola), zostały zdefiniowane przez Ken Schwabera, Mike Beedle w następujący sposób [8]:

- Umiejętność podejmowania zobowiązania rozumiana jako silne poczucie związku z wyrażanymi opiniami i podejmowanymi działaniami oraz konsekwencja w ich realizacji. Przykładem może być podjęcie generalnego zobowiązania postępowania, zgodnie z regułami metody Scrum zestawione z faktyczną determinacją w realizacji tego postanowienia wobec pojawiających się problemów.
- Uważność i koncentracja uwagi, bez nich nie jest możliwe dogłębne rozumienie problemów i potrzeb, których rozwiązaniem ma być wytworzone oprogramowanie. Wobec dużej liczby dystraktorów obecnych w środowiskach dużych organizacji (zaangażowanie w poboczne projekty czy komunikacja email) łatwo o rozproszenie uwagi, co nie przyczynia się do uzyskania wysokiej efektywności i doskonałości technicznej wytwarzanego oprogramowania. Scrum wspomaga utrzymanie koncentracji poprzez wprowadzenie ograniczeń czasowych z jednoczesnym klarownym oznaczeniem celu, z których najbardziej jaskrawym przykładem jest fakt wyodrębnienia atomowych (separowanych od czynników zewnętrznych) cykli produkcyjnych.
- Otwartość wskazywana jako gotowość do dzielenia się informacją ze wszystkimi członkami zespołu, bez względu na charakter – zgodny z oczekiwaniami czy nie – tej informacji. Scrum wspomaga otwartość poprzez wbudowaną zasadę przejrzystości i klarowny rozdział odpowiedzialności. Wszystkie elementy Scruma – w szczególności rejestry (backlogi), przyrost produktu, postęp prac – są nie tylko jawne (dostępne dla wszystkich zainteresowanych), ale wręcz poddawane zespołowemu procesowi kontroli. Dodatkowo wyznacza się osobę stojącą na straży tej reguły.
- Poszanowanie, oznaczające pozytywne odczucia jednej osoby względem drugiej i umiejętność budowania synergii przy uwzględnieniu silnych i słabych stron innych członków zespołu, różnic w doświadczeniu, wykształceniu i roli w zespole. Umiejętność budowania pozytywnych relacji zorientowanych na stawiany przed zespołem cel jest pożądaną cechą wspomagającą proces budowania zespołu. Cecha ta nie powinna ulegać wpływowi wzajemnych animozji, politykierstwa i innych negatywnych elementów codziennego

zycia każdej organizacji. Scrum, poprzez wykorzystanie samoorganizacji, rozumianej jako powierzenie odpowiedzialności za realizację zadań w ręce zespołu, a nie pojedynczych osób. Zespół Scrum stanowi całość i jedność, więc tak jest postrzegany przez resztę organizacji, o czym wszyscy pracujący z zespołem Scrum powinni pamiętać.

- Odwaga jest elementem niezbędnym do istnienia i rozwoju Scruma. Jako nowa metoda pracy wymaga ona zmiany dotychczasowego porządku, przełamywania zwyczajów i sposobów wykonywania pracy, co często wiąże się z dużym oporem. Odwaga jest wymagana w celu wyjawiania i wyjaśniania problemów, identyfikacji przeszkód, zwracania się o pomoc, udzielania lub przyjmowania tej pomocy. Scrum jako proces empiryczny wymaga ciągłego poszukiwania nowych, lepszych, bardziej efektywnych sposobów pracy i rozwiązywania problemów. Bez odwagi do eksplorowania niezbadanych dotąd obszarów nie jest to możliwe.

3.1. Scrum jako proces empiryczny – filary procesu

Podejście empiryczne (doświadczalne) jest właściwe dla zjawisk, których przebiegu nie można z góry przewidzieć i dokładnie opisać. Oprócz podejścia empirycznego, wyróżnia się również podejście zdefiniowane (preskryptywne, teoretyczne).

Scrum jest procesem empirycznym, którego podstawowymi założeniami są [1]:

- Przejrzystość (ang. *transparency*) – każdy element procesu musi być jawny i zrozumiały dla wszystkich.
- Inspekcja (ang. *inspection*) – każdy element procesu musi być cyklicznie poddawany przeglądowi pod kątem jego dopasowania do bieżących warunków. Przegląd ten jest wykonywany w celu zlokalizowania przeszkód, które powodują, iż proces ten nie jest optymalny.
- Adaptacja (ang. *adaptation*) – po zlokalizowaniu przeszkody w procesie należy tak szybko, jak to możliwe, wyeliminować ją w celu usprawnienia procesu.

3.2. Scrum jako metodyka normatywna (szkielet pracy)

W niektórych opracowaniach dotyczących Scruma występują stwierdzenia [1], że Scrum nie jest metodyką, lecz jest szkieletem pracy (ang. *framework*). By rozstrzygnąć tę kwestię, należy wyjaśnić, jaki obszar obejmuje pojęcie metodyka, a jaki szkielet pracy. Istnieje podział na dwa podstawowe typy metodyk [2]:

- Metodyka opisowa (*descriptive*) – koncentruje się na opisie reguł, czynności i ich wytworów.
- Metodyka normatywna (*prescriptive*) – ustala reguły i normy poprawnego postępowania.

Większość metod klasycznej teorii decyzji ma charakter normatywny, tzn. polega na wyznaczeniu optymalnego rozwiązania przez idealnego decydenta, który całkowicie wykorzystuje dostępne mu informacje, wyznacza korzyści z perfekcyjną dokładnością i działa w pełni racjonalnie.

Podejście deskryptywne, tzn. opisowe, opisujące typowe zachowania człowieka w danej sytuacji decyzyjnej. W takim podejściu jest dokonywany wybór działań spośród możliwych działań przy rozważeniu korzyści i strat związanych z wyborem działania.

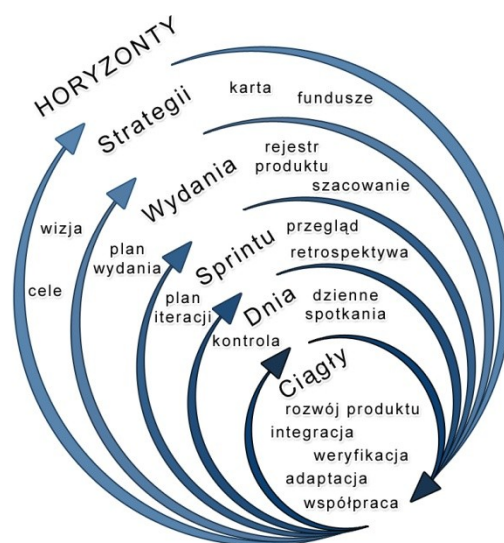
Metodyka normatywna jest rozumiana jako szkielet pracy, gdyż nie tylko opisuje, co powinno być zrobione, lecz dostarcza struktury implementacyjnej metodyki.

W związku z powyższymi rozważaniami Scrum będzie traktowany jako metodyka normatywna utożsamiana z ramą pracy (framework). Ściślej rzecz ujmując, Scrum stanowi ramę pracy, w obrębie której, w sposób empiryczny, powinien zostać wypracowany proces wytwórczy właściwy dla danej organizacji i realizowanego przez nią przedsięwzięcia [1].

4. Zdarzenia metodyki SCRUM

W metodyce Scrum jest definiowanych wiele powtarzających się przedziałów czasowych, które mogą być interpretowane jako zdarzenia. Zastosowano w niej postulaty Manifestu Agile, dotyczące dostaw produktu w krótkich (postulat 3), regularnych (postulat 12) odstępach oraz utrzymania zrównoważonego i stałego tempa rozwoju (postulat 8). Postulaty zrealizowano poprzez wyznaczenie wielu powtarzalnych cykli. Podstawowy cykl metodyki zwany Sprintem posiada stałą długość. Sprint jest zawarty w cyklu wydania, a ten w cyklu strategii. Z drugiej strony, cykl sprintu zawiera w sobie cykl dnia, a ten cykle bieżące (ciągłe). Zaletą takiego rozwiązania jest regularność (rytm) i wynikająca z niego przewidywalność, ograniczenia czasowe stanowią także czynnik mobilizujący – konieczność zdążenia z pracami przed zakończeniem się cyklu.

W tym opracowaniu jako punkt wyjścia dla cykli czasowych przyjęto specyfikację Mike'a Cohna [9], który opisał etapy zwinnego rozwoju projektu w postaci tzw. cebulki planowania. Warstwami cebulki są zawarte kolejno w sobie horyzonty czasowe, od ogólnych, długich cykli aż po cykle bieżące (ciągłe). Ta propozycja została następnie, przy jego pełnym wsparciu, zmodyfikowana przez firmę VersionOne [10] tak, aby lepiej przystawała do ogólnego opisu metodyk zwinnych. Modelując horyzonty planowania, jako zdarzenia wykorzystano pojęcia zaproponowane przez VersionOne (rys. 2). W opracowaniu dokonano niezna-
cznej modyfikacji uwzględniającej adaptacyjność metodyki oraz fakt, że Scrum nie musi być wykorzystywany tylko i wyłącznie przy wytwarzaniu oprogramowania.



Rys. 2. Horyzonty planowania Scrum
Fig. 2. Scrum planning boundaries

Cykle metodyki (horyzonty czasowe), zwane także przedziałami czasowymi (ang. *time-box*), są zdarzeniami określonymi przez okresy czasu, w których mogą być wykonywane różne procesy. W Scrum wyróżniamy następujące horyzonty czasowe:

- Strategii – horyzont ten dotyczy całego cyklu życia produktu. Uruchomienie cyklu strategicznego wymaga określenia celów, wizji, karty produktu i decyzji uruchamiających rozwój produktu.
- Wydania – horyzont ten obejmuje rozwój produktu związany z jednym wydaniem. W trakcie rozwoju produktu może pojawić się wiele wydań. Cykl wydania rozpoczyna określenie potrzeb wobec produktu, planu wydania i estymacji wymaganych prac.
- Sprint – jest to horyzont o stałej długości i dotyczy rozwoju produktu. Do wykonania sprintu jest potrzebny plan sprintu, a w jego trakcie jest uzyskiwany przyrost produktu. Cykl zamyka przegląd uzyskanego w trakcie sprintu przyrostu oraz retrospektywę.
- Horyzont dzienny – każdego dnia jest kontrolowany i monitorowany stan postępu prac oraz jest dokonywana adaptacja – zmiana w obrębie sposobu wykonania pracy – prowadząca do zakończenia pracy w sprincie z sukcesem.
- Horyzont ciągłego wykonania obejmuje wiele prac wykonywanych na co dzień, do których można zaliczyć: rozwój produktu (projektowanie, programowanie), integrację, weryfikację, współpracę, komunikację.

Każdy sprint jest samodzielnym, kompletnym cyklem wytwórczym, którego celem jest realizacja wydzielonego fragmentu funkcjonalności systemu (przyrost produktu). Pojedynczy sprint ma fundamentalne znaczenie, gdyż reguluje i zawiera wszelkie czynności związane z planowaniem i realizacją prac prowadzących do rozbudowy funkcjonalności systemu.

Autorzy metodyki sugerują zastosowanie sprintów o długości zawartej pomiędzy 2 a 4 tygodniami. W celu ograniczenia ryzyka związanego z liczbą zmian w trakcie sprintu czas

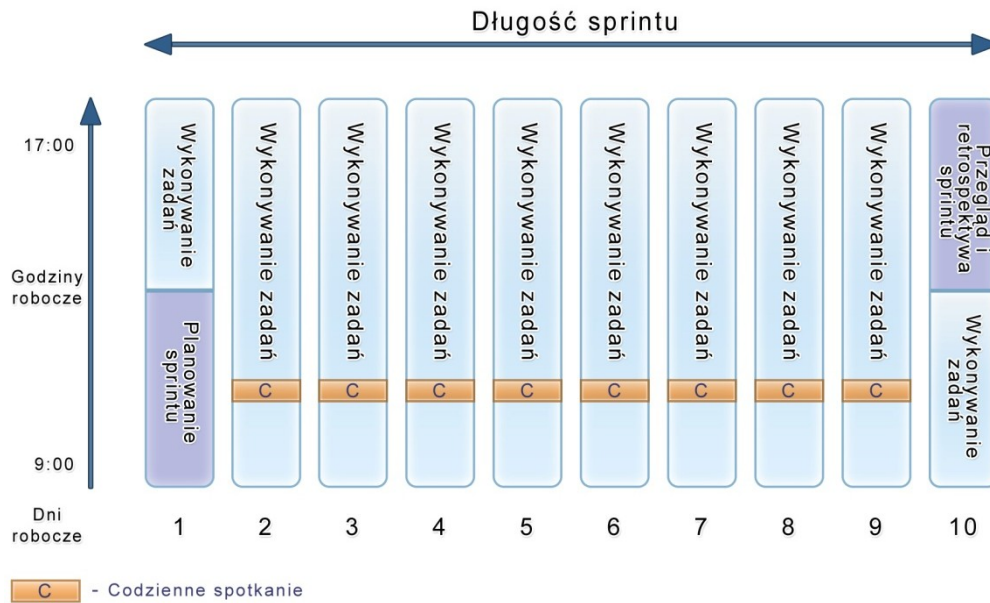
jego trwania nie powinien przekraczać 1 miesiąca. W praktyce obserwuje się jednak również sprinty krótsze (1–tygodniowe) oraz dłuższe 5– i 6–tygodniowe. Należy pamiętać o utrzymaniu stałego czasu trwania sprintów, a także o tym, iż kolejne sprinty następują bezpośrednio po sobie, bez przerw pomiędzy nimi. Nie jest możliwe wydłużenie czasu trwania sprintu w jego trakcie, jest natomiast dopuszczalne, w sytuacjach wyjątkowych, jego skrócenie, polegające na jego przedwczesnym zamknięciu i natychmiastowym uruchomieniu kolejnego sprintu bazującego na zmienionych założeniach. Utrzymanie dyscypliny czasowej sprintów jest niezbędne do zainicjowania i utrzymania procesu kontroli i adaptacji.

W trakcie każdego sprintu zachodzą stałe zdarzenia o charakterze spotkań zespołu Scrum (rys. 3):

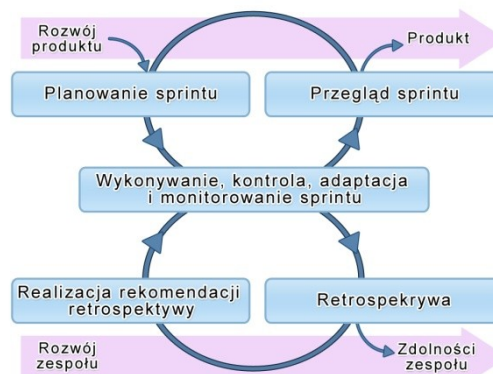
- Spotkanie planistyczne dla sprintu przebiega dwuetapowo:
 - Część I dotyczy wyznaczania celu i zakresu sprintu.
 - Część II dotyczy planowania zadań sprintu.
- Spotkanie planistyczne dla sprintu 1–miesięcznego trwa 8 godzin, w tym 4 godziny na planowanie celu sprintu oraz 4 godziny na planowanie zadań sprintu. Zgodnie z wytycznymi metodyki planowanie sprintu powinno obejmować około 5–10% czasu jego trwania. „W ogólnym rozrachunku prace planistyczne w Scrumie pochłaniają nieco więcej wysiłku niż tradycyjne planowanie projektu” [1]. Jest to tym bardziej prawdziwe twierdzenie, jeśli uwzględni się fakt, iż w planowaniu bierze udział cały zespół projektowy, w kontraście do planowania wykonywanego przez dedykowaną osobę, ewentualnie wspomaganą przez niewielkie grono ekspertów.
- Codzienne spotkanie (tzw. codzienny Scrum) jest 15–minutowym spotkaniem kontrolnym zespołu przeznaczonym na omówienie bieżącego stanu prac.
- Spotkanie przeglądu sprintu polega na przeglądzie i ocenie (walidacji) rezultatów (przyrostu produktu) osiągniętych w trakcie sprintu. Przegląd sprintu podlega ograniczeniu czasowemu (4 godziny dla sprintu miesięcznego, nie więcej niż 5% trwania sprintu).
- Spotkanie retrospektywy sprintu jest wykorzystywane do sformułowania wniosków z przebiegu iteracji i zaleceń na przyszłość, dotyczących poprawy pracy zespołu. Czas trwania spotkania retrospektywy sprintu jest ograniczony i wynosi 3 godziny.
- Część wykonawcza sprintu jest związana z rozwojem produktu. Jest to przedział czasowy sprintu bez spotkań planistycznych, przeglądu i retrospektywy sprintu. Na rysunku 3 jest zaznaczona jako „Wykonanie zadań” sprintu.

Strukturę sprintu wraz z występującymi stałymi fragmentami czasowymi przedstawiono na rysunku 4.

Należy zwrócić uwagę, że metodyka Scrum kładzie nacisk na proces ciągłego dostarczania kolejnych przyrostów funkcjonalności, w postaci stabilnego produktu, w niewielkich okresach czasu.



Rys. 3. Struktura czasowa sprintu dwutygodniowego (10-dniowego)
 Fig. 3. Time structure of a two-week sprint (10 days)



Rys. 4. Schemat przebiegu sprintu
 Fig. 4. Sprint flowchart

Planowanie sprintu jest realizowane na podstawie wymagań wobec produktu zawartych w rejestrze produktowym. Początkowo może zawierać on jedynie tyle informacji, aby wypełnić zespołowi czas pracy w pojedynczym *Sprincie*. W projektach o lepiej określonym zakresie prac horyzont ten może być bardziej odległy i może obejmować okres od kilku do kilkunastu *Sprintów*. Odradza się jednak planowanie prac na odleglejsze okresy czasu jako obciążone zbyt dużą niepewnością.

W trakcie przebiegu sprintu jest wytwarzany przyrost produktu. Pod koniec sprintu jest dokonywana ocena produktu i produkt może być potencjalnie wydany na zewnątrz.

Po wykonaniu przebiegu i przeglądu sprintu jest dokonywana ocena działań podejmowanych przez zespół projektowy. Ocena taka jest dokonywana w trakcie spotkania, zwanego retrospektywą sprintu.

Na rysunku 4 przedstawiono dwa strumienie rezultatów sprintu związane z:

- Przyrostem wartości produktu postrzeganej przez jego odbiorcę (ang. *potentially shippable product increment*).
- Przyrostem potencjalnych możliwości zespołu (ang. *increased team capability*).

5. Artefakty

Kategoria artefakt (ang. *artifact*) grupuje klasy obiektów, które powstają w trakcie wykonywania procesów. W wyniku analizy opracowań dotyczącej metodyki Scrum autorzy opracowania postanowili wybrać i opisać zestaw artefaktów istotnych z punktu widzenia zespołu Scrum.

Rejestr produktowy (ang. *product backlog*) zawiera wykaz wymagań przeznaczonych do wykonania w trakcie rozwoju produktu. Rejestr ten jest nazywany także zaległościami produktowymi w celu podkreślenia, jakie jeszcze prace powinny być wykonane związane z rozwojem produktu.

Rejestr produktowy przyjmuje zazwyczaj postać listy uporządkowanej pod względem ważności jej elementów, determinującej kolejność ich realizacji. Metodyka nie specyfikuje natury poszczególnych wpisów, określając je ogólnym mianem *elementów rejestru produktowego*, wymagając jedynie, aby niosły one ze sobą wyczerpującą informację o cechach realizowanego systemu (mogą to być zarówno wymagania funkcjonalne, jak i niefunkcjonalne), koszcie ich realizacji (określenie złożoności lub pracochłonności) oraz aby były uporządkowane (wartościowane) pod względem oczekiwanej kolejności realizacji.

Zawartość rejestru produktowego ewoluuje wraz z postępem prac i może być modyfikowana nie tylko na podstawie informacji zwrotnej od końcowego użytkownika oraz nowych informacji biznesowych czy technicznych, lecz także w wyniku dokonywania dekompozycji poszczególnych pozycji rejestru na bardziej szczegółowe.

Ponadto, kolejność realizowania poszczególnych wymagań (elementów rejestru produktowego) może ulegać zmianie ze sprintu na sprint. Elementy przeznaczone do realizacji w najbliższych sprintach są opisane w sposób precyzyjny i szczegółowy, natomiast te o odleglejszym czasie realizacji są zazwyczaj zarysowane jedynie ogólnie.

Zdolności zespołu (ang. *team capability*) określają ogólne zdolności zespołu do wykonania prac w sprintach. Zdolność zespołu jest scharakteryzowana między innymi przez następujące własności:

- Prędkość zespołu (ang. *velocity*) jest to prędkość produkcyjna zespołu wyrażana jako wartość liczbowa reprezentująca ilość wytwarzanej przez zespół funkcjonalności produktu w pojedynczej iteracji (sprincie). Wartość ta jest wyliczana jako suma początkowych

estymat zespołu wszystkich oddawanych na koniec iteracji funkcjonalności. Na potrzeby planowania jest wykorzystywana średnia (lub średnia krocząca) tej wartości obejmująca poprzednio zrealizowane sprinty.

- Dostępność zespołu w sprincie (ang. *team capacity*) rozumiana zazwyczaj jako sumaryczna liczba godzin roboczych w sprincie pomniejszona o liczbę godzin przeznaczonych na urlopy i inne prace członków zespołu.
- Kompetencje zespołu reprezentują wiedzę i umiejętności dotyczące zarówno wytwarzanego produktu, jak i używanej w tym celu technologii, a także inne własności zespołu, jak na przykład jego interdyscyplinarność i samoorganizację.

Kryterium gotowości (ang. *definition of ready*) do rozpoczęcia sprintu określa warunki, dotyczące opisów elementów rejestru produktowego, których spełnienie umożliwi rozpoczęcie sprintu. Kryterium gotowości określa zatem, czy stan rejestru produktowego zezwala na właściwe rozpoczęcie sprintu. Zwyczajowo mówi się, że rejestr produktowy jest dobrze przygotowany, jeżeli liczba precyzyjnie opisanych wpisów w rejestrze wystarcza na wykonanie od 2 do 3 sprintów.

Kryterium ukończenia (ang. *definition of done*) może być rozumiane jako lista definicji opisujących, w jakim przypadku poszczególne lub pewna grupa elementów rejestru produktowego mogą zostać uznane przez właściciela produktu za poprawnie wykonane. Kryterium to, szczególnie w procesie rozwoju oprogramowania, wyznacza oczekiwany poziom jakościowy. Przykładowo:

- Wykonane (done) – zaimplementowane i wykonywalne w środowisku programisty.
- Wykonane – wykonane (done – done) – ponadto pozytywnie zweryfikowane poprzez testy, przeglądy kodu źródłowego itp.
- Wykonane – wykonane – wykonane (done – done – done) – dodatkowo pomyślna walidacja dokonana poprzez testy integracyjne, akceptacyjne, testy użyteczności itp.

Kryterium ukończenia powinno zostać uzgodnione pomiędzy zespołem a właścicielem produktu przed rozpoczęciem sprintu, aby zespół projektowy zdawał sobie sprawę z oczekiwań właściciela produktu, a z drugiej strony, by właściciel produktu rozumiał, co będzie podlegało walidacji na koniec sprintu i jakiego rodzaju prace zostaną w sprincie wykonane. Dobrze opisane kryterium ukończenia pomaga określić ilość pracy do wykonania w sprincie oraz właściwie przygotować rejestr zadaniowy. Kryterium ukończenia może dotyczyć pojedynczych elementów rejestru produktowego, ich grup, całego sprintu lub wydania.

Cel sprintu (ang. *sprint goal*) przedstawia opisowy cel pojedynczego Sprintu, wyznaczający kierunek prac przeznaczonych do wykonania w Sprincie. Nadrzędny cel Sprintu pomaga zespołowi utrzymać koncentrację podczas wykonywania prac w Sprincie. Cel sprintu zazwyczaj jest przedstawiony lapidarnie jako odpowiednio dobrana nazwa sprintu lub metaforyczne zdanie podsumowujące zakres prac do wykonania w sprincie.

Zakres sprintu (ang. *sprint scope*) stanowi grupę elementów rejestru produktowego wybraną do wykonania w sprincie. Grupa tych elementów jest także nazywana w środowisku Scrum rejestrem uzgodnionym (ang. *comitted backlog*).

Rejestr zadaniowy (ang. *sprint backlog*) jest szczegółowym wykazem prac zaplanowanych do zrealizowania w obrębie pojedynczego sprintu. Rejestr zadaniowy jest uzyskiwany w wyniku dekompozycji elementów rejestru produktowego na bardziej szczegółowe zadania. Rejestr jest wykorzystywany przez zespół do skoordynowania jego własnych działań w sprincie. Zazwyczaj rejestr zawiera listę opisów zadań i informacje o ich przydziale do konkretnych wykonawców, statusie i szacowanym czasie wykonania. Rejestr zadaniowy odpowiadający przyjętemu do realizacji zakresowi sprintu jest nazywany bazowym zakresem sprintu (ang. *sprint scope baseline*). W trakcie wykonania sprintu rejestr zadaniowy może ulec zmianie, jeśli zespół zidentyfikuje nowe, niezbędne zadania lub uzna część zadań za niepotrzebną.

Wykres wypalania dla sprintu (ang. *sprint burndown chart*). W trakcie wykonania sprintu zespół przedstawia postęp prac w postaci diagramu zwanego wykresem wypalania. Postęp prac jest podawany jako liczba godzin pracy (oś Y) zespołu pozostających do zrealizowania celu sprintu, dla każdego dnia roboczego (oś X) sprintu. Liczba godzin pozostałych do wykonania celu sprintu powinna zmniejszać się z każdym kolejnym dniem sprintu, może jednak ulec zwiększeniu w wyniku stwierdzenia, że czas potrzebny na wykonanie zadań jest dłuższy, niż to początkowo szacowano, lub w wyniku konieczności przeprowadzenia innych, nieprzewidzianych prac. Warto pamiętać, że wykres wypalania (podobnie jak tablica) jest jedynie narzędziem wspomagającym pracę zespołu. Jego przygotowanie i prowadzenie nie powinno zatem stać się celem samym w sobie.

Przyrost produktu (ang. *product increment*). Przyrost produktu jest identyfikowany zazwyczaj jako zbiór elementów z uzgodnionego rejestru produktowego, który pod koniec sprintu spełniają kryterium ukończenia. Warto jednak zauważyć, iż w trakcie sprintu zespół może zaproponować inne niż początkowo dyskutowane rozwiązanie, co powoduje, że przyrost jest identyfikowany i oceniany także pod kątem spełnienia (bądź nie) celu sprintu. Na podstawie zaimplementowanych własności produktu jest planowany jego dalszy rozwój.

Produkt (ang. *product*). Zazwyczaj w trakcie kolejnych sprintów rozwijany jest produkt, który na zakończenie powinien charakteryzować się potencjalną możliwością przekazania go odbiorcy. Wykonany w trakcie pojedynczego sprintu przyrost produktu powinien zostać zintegrowany w spójną całość z pracami wykonanymi w poprzednich sprintach bądź pracami wykonanymi przez inne zespoły współtworzące produkt i w takiej postaci przekazany odbiorcy.

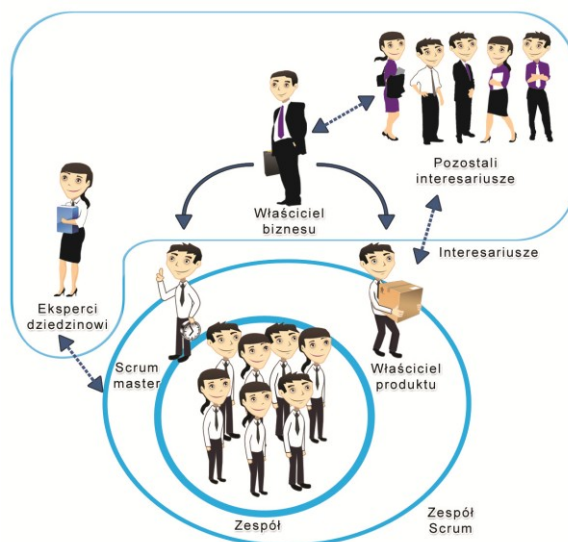
Rekomendacje rozwoju produktu (ang. *recommendations of product development*) opisują, jakie kroki związane z rozwojem produktu zostaną podjęte w kolejnym i następnych sprintach.

Rekomendacje retrospektywy (ang. *retrospective recommendations*) przedstawiają zalecenie związane z ciągłym doskonaleniem działania zespołu. Rekomendacje te mogą dotyczyć działań bieżących (ang. *short-term*) i długoterminowych (ang. *long-term*), przekraczających długość pojedynczego sprintu.

6. Role

W celu sprawnego rozwoju produktu wyróżniono tylko kilka podstawowych i ważnych ról. Role można podzielić na dwie kategorie – zespół Scrum i interesariuszy. W metodyce Scrum wyróżnia się jedynie kilka ról, są to (rys. 5):

- Właściciel produktu (ang. *product owner*).
- Scrum Master (ang. *scrum master*).
- Zespół projektowy (ang. *team*).
- Interesariusze (ang. *stakeholders*).



Rys. 5. Role w metodyce Scrum
Fig. 5. Roles in a Scrum methodology

Zespół Scrum (ang. *scrum team*). Aby umożliwić identyfikację grupy odpowiedzialnej za wytworzenie danego produktu na tle całej organizacji, a także, by ułatwić wprowadzenie i egzekwowanie reguł metodyki wobec i w obrębie tej grupy, wyróżnia się tak zwany zespół Scrum, w skład którego wchodzi: Scrum Master, właściciel produktu i zespół wykonawczy [1]. W praktyce często ta sama osoba pełni rolę Scrum Mastera i jest aktywnym członkiem zespołu rozwijającego produkt (np. programistą). Nie jest wskazane natomiast, by osoba, peł-

niąca rolę właściciela produktu, była jednocześnie członkiem zespołu rozwijającego produkt. Nie można również łączyć ról Scrum Mastera i właściciela produktu.

Właściciel produktu (ang. *product owner*) jest osobą odpowiedzialną za utrzymanie zestawu cech produktu przewidzianych do implementacji, zarządzanie kolejnością realizacji wymagań, zgodnie z reprezentowaną przez nie wartością biznesową oraz za przejrzyste komunikowanie tych informacji. Właściciel produktu ma uprawnienia związane z wyznaczaniem celów zespołu Scrum i jest odpowiedzialny za jego sukces. Zespół Scrum może mieć tylko jednego właściciela produktu, jednak właściciel może kierować pracą jednego lub więcej zespołów.

Lista obowiązków właściciela produktu może wyglądać następująco:

- Udział w przygotowaniu i rozwoju wizji produktu oraz przekazywanie jej zespołowi.
- Tworzenie i aktualizacja planów wydań w celu kierowania projektem w sposób proaktywny, angażując w ten proces zespół Scrum i interesariuszy.
- Przygotowywanie zestawu cech, którymi ma się charakteryzować wytwarzany produkt, oraz podejmowanie decyzji o kolejności ich wykonania przez zespół. Zapisywanie pożądanых cech produktu w rejestrze produktowym i współpraca z zespołem i interesariuszami w tym zakresie.
- Szacowanie zyskowności przedsięwzięcia związanego z tworzonym produktem.
- Koncentrowanie się na tworzeniu produktu o minimalnym zakresie funkcjonalności oraz wydawaniu przyrostów produktu wcześniej i często. Usuwanie elementów rejestru produktowego, które nie są kluczowe.
- Dobór celu sprintu.
- Ocena rezultatów pracy zespołu, w wyniku której następuje akceptacja lub odrzucenie zadań sprintu.
- Monitorowanie przebiegu prac i raportowanie stanu projektu przed interesariuszami,
- Codzienna współpraca z zespołem oraz udział w spotkaniach przewidzianych w metodyce Scrum.

Postuluje się [1], aby rola właściciela produktu była pełniona wyłącznie przez jedną osobę, posiadającą odpowiednią moc sprawczą, a nie przez grupę osób. Nie oznacza to oczywiście, iż właściciel produktu nie może zasięgać opinii szerszego gremium i w praktyce bardzo często tak właśnie jest. Właściciela produktu wspiera zarówno zespół wykonawczy, jak i osoby spoza tego zespołu, np. dział marketingu. Właściciel produktu jest osobą oficjalnie odpowiedzialną za powodzenie projektu, każdy w organizacji powinien respektować podejmowane przez niego decyzje. Właściciel produktu reprezentuje przed zespołem stanowisko wszystkich interesariuszy realizowanego przedsięwzięcia. Nie jest dopuszczalne, by ktoś inny przekazywał zespołowi inne priorytety prac, natomiast zespół ze swojej strony musi podporządkować się wytycznym stawianym przez właściciela produktu [14].

Poważnym problemem dla rozwoju produktu są sytuacje, w których właściciel produktu:

- nie został formalnie zdefiniowany i przypisany do zespołu,
- nie posiada odpowiednich pełnomocnictw (opóźnienia w podejmowaniu decyzji),
- nie posiada odpowiednich kwalifikacji lub wiedzy merytorycznej w zakresie realizowanego przedsięwzięcia,
- odznacza się brakiem zaangażowania w sukces przedsięwzięcia,
- odznacza się brakiem cech przywódczych (umiejętności kierowania, motywowania i podejmowania decyzji),
- nie jest w stanie wygospodarować odpowiedniej ilości czasu dla zespołu (np. brak przygotowania do sesji planistycznych, opóźnienia w reagowaniu na zapytania zespołu)
- nie został przeszkolony w zakresie metodyki Scrum i technik pomocniczych (np. strategia zarządzania rejestrem produktowym, historie użytkownika).

Scrum Master (ang. *scrum master*) – w polskiej nomenklaturze rola ta jest nazywana bardzo różnie, między innymi: młynarz, mistrz, strażnik młyna, scrum majster lub mistrz ceremonii. Ponieważ rola ta jest ściśle powiązana z metodyką Scrum, postanowiono pozostać przy nazwie angielskiej. Scrum Master jest osobą odpowiedzialną za egzekwowanie praktyk i zasad metodyki Scrum. Jego podstawową rolą jest udzielanie wsparcia zespołowi i całej organizacji związanego z efektywnym jej wykorzystaniem. W pewnym zakresie jest także odpowiedzialny za szkolenie zespołu i właściciela produktu, jeżeli nie posiadają wystarczającej wiedzy na temat swoich ról i obowiązków wynikających z metodyki. Scrum Master pomaga także zespołowi wykonywać pracę sprawnie, z zachowaniem zasad samoorganizacji, w środowisku, które nie jest w pełni przystosowane do wytwarzania złożonych produktów. Pomoc ta, w nomenklaturze metodyki Scrum, jest nazywana usuwaniem blokad (ang. *impediments*) [1]. Scrum Master przebywa z zespołem, dba o zespół i jego morale, stabilne tempo i postęp prac, nadzoruje zgodność praktyk projektowych z założeniami metodyki. Scrum Master nie zarządza zespołem – nie definiuje i nie przypisuje zadań – zespół jest samoorganizujący, a Scrum Master pełni funkcję katalizatora, który tę samoorganizację wsporaga. Lista obowiązków osoby pełniącej rolę Scrum Mastera może wyglądać następująco:

- Egzekwowanie od zespołu i reszty organizacji poszanowania reguł metodyki Scrum oraz edukowanie w tym zakresie i zachęcanie do wykorzystania jej mechanizmów w rozwiązywaniu codziennych problemów.
- Aktywne zachęcanie zespołu do poszukiwania alternatywnych rozwiązań dla napotkanych problemów.
- Sprawne i terminowe usuwanie wewnątrz i poza zespołem czynników, mogących negatywnie wpłynąć na rezultaty sprintu. Scrum Master nie musi rozwiązywać problemów samodzielnie, szczególnie że mogą one wykraczać poza jego możliwości lub kompetencje, oczekuje się natomiast, iż wypracuje sposoby zaangażowania odpowiednich

osób (w szczególności zespołu Scrum lub kierownictwa) do wypracowania i wdrożenia rozwiązania.

- Zapewnianie ścisłej współpracy pomiędzy reprezentantami wszystkich funkcji i ról oraz kontrola przebiegu tej współpracy.
- Zarządzanie zmianą zachodzącą w organizacji (każda zmiana powoduje stres ze względu na pojawiające się problemy, braki organizacji, kompetencji itp.).

Zespół wykonawczy (ang. *team*), nazywany również zespołem projektowym lub krócej – zespołem, to grupa osób zaangażowanych w realizację przedsięwzięcia. Grupa ta jest mało liczna, by ograniczyć narzut komunikacyjny, jednak na tyle duża, by mógł zachodzić efekt wzajemnego motywowania się przez jej członków. Zaleca się, by zespół wykonawczy liczył 7 ± 2 osoby (nie licząc właściciela produktu i Scrum Mastera) [1].

Kompetencje osób wchodzących w skład zespołu są tak dobrane, aby możliwe było wykonanie stawianych przed nimi zadań, prowadzących zazwyczaj do wytworzenia (kompletnego) produktu. W przypadku wytwarzania oprogramowania będą to więc zarówno programiści, projektanci, jak i analitycy i testerzy, a także, jeżeli projekt tego wymaga, administratorzy baz danych, administratorzy systemu, graficy etc. Pracują oni ze sobą na równych prawach, przy zachowaniu koleżeńskich relacji i wykonując wspólnie postawione przed nimi zadania, nawet jeśli wymagałoby to nabycia nowych kompetencji i było związane z przekraczaniem granic swojej podstawowej roli (analityka, projektanta, programisty, testera).

Jak wspomniano wcześniej wielokrotnie, zespół jest samoorganizujący się. Członkowie zespołu powinni zatem dążyć do wykonania powierzonych im zadań we własnym gronie, organizując przebieg pracy we własnym zakresie. Każdy członek zespołu aktywnie uczestniczy w procesie podziału zadań i poszukiwania rozwiązań napotkanych problemów. Członkowie zespołu powinni w pełni poświęcić się (ang. *commit*) wykonaniu projektu i nie brać udziału w innych pracach. Nikt nie definiuje przebiegu pracy zespołu. Zespół zostanie oceniony na podstawie stopnia zrealizowania założonego celu sprintu.

Interesariusze to osoby spoza zespołu Scrum zainteresowane lub zależne od przebiegu prac projektowych. Najbardziej typowymi interesariuszami są: inwestor, kierownictwo firmy, użytkownicy. Zakłada się, że interesariusze powinni być aktywnie zaangażowani w realizację projektu (ang. *involved*), w szczególności biorą udział w niektórych spotkaniach zespołu (pierwsza część planowania i przegląd sprintu), jednak wobec zespołu wykonawczego ich reprezentantem jest właściciel produktu.

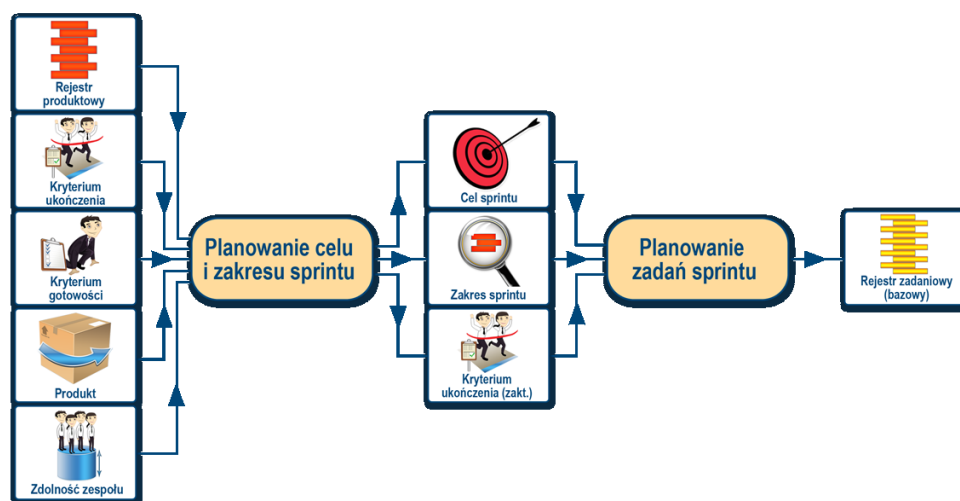
7. Procesy

W wyniku analizy opracowań związanych z metodyką Scrum w systematycznym opisie metodyki wyróżniono zestaw podstawowych procesów, każdy z nich został powiązany z czterema rodzajami elementów procesu: artefaktami wejściowymi, artefaktami wyjściowymi, rolami odpowiedzialnymi związanymi z realizacją procesu zdarzeń wyzwalających proces lub w trakcie których proces jest wykonywany. Warto zwrócić uwagę na uwidocznione w modelu (rys. 4 i 3) ogólne powiązania procesów Scrum.

7.1. Planowanie sprintu

Czynności związane z rozpoczęciem Sprintu, określane wspólnym mianem planowania sprintu (ang. *sprint planning*), przebiegają dwuetapowo podczas dwóch następujących po sobie spotkań (rys. 6):

- Spotkanie I – Planowanie celu i zakresu sprintu.
- Spotkanie II – Planowanie zadań sprintu.



Rys. 6. Schemat procesów planowania sprintu

Fig. 6. Sprint planning processes scheme

W trakcie I spotkania jest dokonywane planowanie celu i zakresu sprintu. Celem tej części spotkania planistycznego jest określenie tego, co ma zostać zrealizowane w rozpoczynanym właśnie sprincie. Właściciel produktu przedstawia elementy rejestru produktowego o najwyższych priorytetach, podając maksymalnie dużo informacji kontekstowej – z czego wynika taka a nie inna kolejność, co oznacza wykonanie danej funkcjonalności z perspektywy strategii produktu etc. W pierwszej kolejności z rejestru produktowego jest wybierany podzbiór elementów, których wykonanie w danym sprincie byłoby najbardziej zasadne. Podzbiór jest wyznaczany zgodnie z celami wydania i ogólną wizją produktu. Przesłankami do dokonania tego wyboru może być wysokie znaczenie danej funkcjonalności dla końco-

wego odbiorcy, względnie wysokie ryzyko realizacji, które właściciel produktu w porozumieniu z zespołem chce obniżyć poprzez wczesne rozpoczęcie prac.

Tak wybrany zbiór elementów jest następnie diskutowany z zespołem projektowym w celu zapewnienia wspólnego rozumienia zakresu prac i kryteriów, którymi kierować się będzie właściciel produktu, poddając ocenie dostarczony przez zespół produkt. Kryteria te odzwierciedlają oczekiwany przez właściciela produktu poziom jakościowy określany jako kryterium ukończenia (ang. *definition of done*). Warto zaznaczyć, że „kryterium”, a raczej „kryteria ukończenia” mogą być podczas sesji planistycznej „odświeżone”, czyli znajdują się zarówno na wejściu, jak i na wyjściu procesu.

Na etapie planowania jest diskutowane również prawdopodobieństwo (stopień) wykonalności. Brane są pod uwagę zależności, ograniczenia, proponowane przez zespół rozwiązania, a także zdolności produkcyjne zespołu (historyczne oraz przewidywane). Ostatecznej decyzji co do zakresu, którego realizacji zespół podejmie się w kolejnym sprincie dokonuje wspólnie cały zespół wykonawczy.

Wybrany w ten sposób podzbiór rejestru produktowego jest nazywany także rejestrem uzgodnionym (ang. *committed backlog*). Osiągnięty konsensus jest podsumowywany i zapisywany w formie celu sprintu, po czym zespół przechodzi do kolejnej, bardziej szczegółowej fazy planowania, obejmującej przygotowanie rejestru zadaniowego (ang. *sprint backlog*), czyli szczegółowego zestawienia prac, których wykonanie jest niezbędne, by osiągnąć cel sprintu.

W trakcie II spotkania jest dokonywane planowanie zadań sprintu. Celem tej części jest określenie w sposób szczegółowy, jak praca ma być wykonana w sprincie. Wybrane w I części sesji planistycznej elementy rejestru produktowego zespół przekształca w zadania do wykonania w sprincie i zapisuje w postaci tak zwanego rejestru zadaniowego sprintu. Dzięki wspólnej pracy i wysiłkowi włożonemu w przygotowanie rejestru zadaniowego jego końcowa postać odzwierciedla realistyczny plan prac, a wszyscy członkowie zespołu poczuwając się do jego autorstwa mają szansę być jednakowo zaangażowanymi w jego realizację. Rejestr zadaniowy jest narzędziem i przedmiotem pracy zespołu, i jest w pełni widoczny dla wszystkich jego członków.

W spotkaniu może uczestniczyć właściciel produktu w celu ewentualnego doprecyzowania zawartości rejestru produktowego, oczekiwań i oceny zależności (ang. *trade-offs*). Warto zwrócić uwagę, iż na tym etapie planowania, w razie zgłoszenia przez zespół wątpliwości, zakres prac w sprincie wciąż może ulegać zmianie, tak by było możliwe efektywne wykorzystanie czasu pracy całego zespołu.

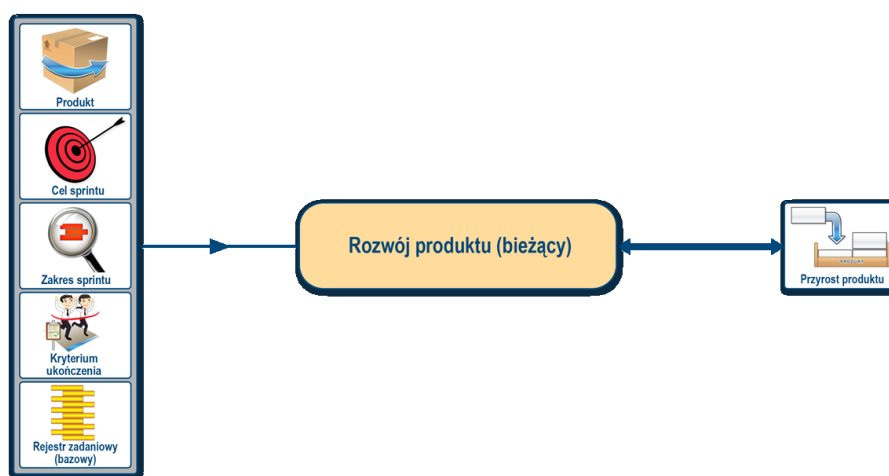
O poziomie szczegółowości tego etapu planowania może świadczyć fakt, iż zaleca się, by oszacowanie czasowe pojedynczego zadania mieściło się w granicach 10 godzin roboczych. Wytyczna ta jest związana z minimalizacją ryzyka, ale stanowi także element stymulujący

pracę zespołową już na etapie planowania. Wysiłek włożony w dekompozycję zadań pozwala zespołowi lepiej zrozumieć pracę, a fakt przygotowania planu pracy wspólnie spaja zespół.

Wybrany do realizacji zbiór elementów rejestru produktowego łącznie z celem sprintu oraz rejestrem sprintu reprezentują tzw. zobowiązanie zespołu na sprint. Od tego momentu zakres pracy zaplanowanej do realizacji w danym sprincie ulega zamrożeniu, ochrona poczynionych postanowień leży w gestii Scrum Mastera. Zadaniem zespołu staje się dążenie do dostarczenia tego, do czego się zobowiązał. Aby zobowiązanie to miało szanse być dotrzymane, skład zespołu pozostaje niezmienny podczas trwania sprintu.

7.2. Rozwój produktu (część wykonawcza sprintu)

W ogólnym ujęciu celem zespołu Scrum jest dostarczenie, na podstawie wybranego przez właściciela produktu wycinka wymagań, w pełni działającego przyrostu produktu (rys. 7) w terminie przewidzianym jedną iteracją (sprintem). Tylko na jego podstawie może być dokonana korekta planu realizacji przedsięwzięcia oraz zmiana w obrębie procesu wytwórczego. Sprint nie może zostać skrócony lub wydłużony. W przypadku gdy prace sprintu mogą nie mieć sensu, sprint może zostać unieważniony przed jego zakończeniem. Tylko właściciel produktu może dokonać takiego unieważnienia [1].



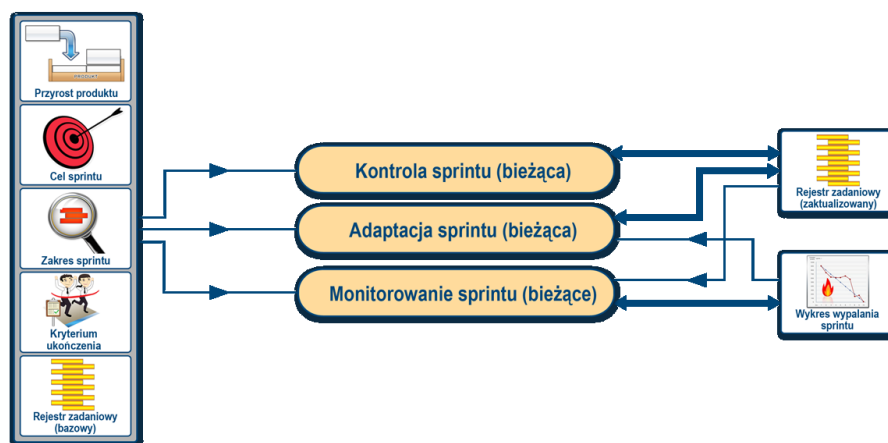
Rys. 7. Schemat procesu rozwoju produktu
Fig. 7. Product development process scheme

W trakcie drugiego etapu planowania oraz wykonania sprintu zasada samoorganizacji zespołu jest najbardziej wyrazista. W praktycznym wymiarze objawia się ona tym, iż zadania nie są przygotowywane i przydzielane odgórnie członkom zespołu, nikt nie ingeruje w to, kto spośród jego członków i kiedy rozpocznie czy też zakończy realizowanie zadań. W szerszym ujęciu jest to przejaw zgodności metodyki z postulatami Takeuchiego i Nonaki. Zespół jest zachęcany do samodzielnego opracowania rozwiązania postawionego przed nim problemu (celu sprintu), wykorzystując wszystkie kompetencje, które członkowie zespołu posiadają oraz empirycznie dobierają proces produkcji i narzędzia. Do dyspozycji zespołu pozostaje

właściciel produktu, będący głównym źródłem informacji odnośnie do nadrzędnego celu projektu, oraz Scrum Master dbający o to, by zespół, postępując zgodnie z regułami metodyki, dysponował wszystkim, co jest niezbędne do wytworzenia produktu. Ponadto, odpowiedzialnością Scrum Mastera jest dbałość o to, by w trakcie sprintu nie były dokonywane żadne zmiany wpływające na zakres prac i cele sprintu.

7.3. Kontrola i monitorowanie sprintu

Ulotna natura samoorganizacji wymaga dyscypliny i zaangażowania wszystkich osób uczestniczących w przedsięwzięciu. Wymaga także wprowadzenia mechanizmu subtelnej kontroli, aby ustrzec zespół przed popadnięciem w chaos. Autorzy metodyki podpowiadają dwie stosunkowo proste techniki związane z kontrolowaniem i monitorowaniem (rys. 8), pozwalające uzyskać efekt pozytywnego wpływu presji grupy na zaangażowanie oraz utrzymanie wysokich standardów pracy przez wszystkich jej członków. W wyniku kontroli i monitorowania może być podjęta decyzja, dotycząca zmian, nazwanych tutaj adaptacją sprintu (rys.8)



Rys. 8. Schemat procesów kontroli i adaptacji w części wykonawczej sprintu
Fig. 8. Control and adaptation processes scheme in the sprint executive part

7.3.1. Kontrola sprintu

Pierwszą z technik jest tzw. codzienny Scrum (ang. *daily scrum*) – krótkie, maksymalnie 15-minutowe, spotkanie całego zespołu. Spotkanie to jest przeprowadzane w tym samym miejscu, o tym samym czasie, a obecność wszystkich członków zespołu jest obowiązkowa. Zespół jest odpowiedzialny za przeprowadzenie i efektywność spotkania. Scrum Master upewnia się, że zespół takie spotkania organizuje oraz dba o to, by utrzymać je w założonym czasie poprzez wprowadzenie reguł, polegających na wypowiedaniu się członków zespołu w sposób treściwy i zwarty. Upewnia się także, by osoby spoza zespołu Scrum nie przerywały jego przebiegu.

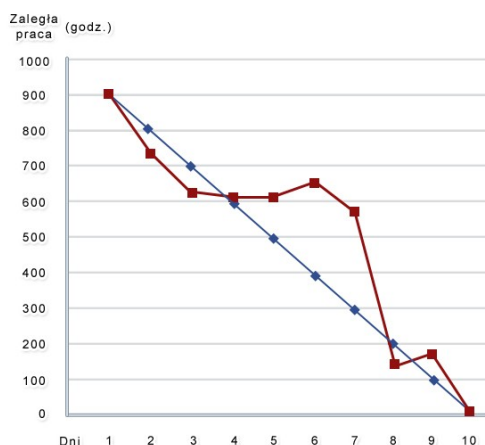
W trakcie codziennych spotkań członkowie zespołu dyskutują postęp prac w stosunku do dnia poprzedniego oraz przedstawiają napotkane przeszkody. Informacja ta pozwala na synchronizację zadań oraz dobór odpowiednich zachowań zespołu zapewniających uzyskanie celów sprintu. Codzienny Scrum nie jest spotkaniem omawiającym szczegółowo stan prac, a jedynie wglądem (inspekcją) w przebieg prac. Inspekcji ten dokonuje zespół i tylko zespół ocenia, czy wobec zaistniałych zdarzeń cele sprintu są zagrożone czy też nie. Sugeruje się 3 pytania, które kierowane do każdego członka zespołu pomagają skupić dyskusję na istotnych dla zespołu (pod kątem osiągnięcia celów sprintu) kwestiach [1]:

- Co zostało wykonane od ostatniego spotkania?
- Czym dana osoba będzie się zajmować do kolejnego spotkania?
- Jakie przeszkody stanęły lub mogą stać na drodze?

Tak zrealizowany proces kontroli usprawnia przepływ informacji w zespole, umożliwia wczesne wykrycie problemów czy sytuacji wymagających przeorganizowania przydziału prac w zespole. Jest także przejawem wzajemnej i subtelnej kontroli, gdyż omówienie przebiegu prac odbywa się wobec i przy aktywnym udziale wszystkich członków zespołu. Uzyskane podczas spotkania informacje mogą skutkować zmianą rejestru zadaniowego i przeplanowaniem przebiegu prac.

7.3.2. Monitorowanie sprintu

Drugą użyteczną techniką zaproponowaną przez twórców metodyki, pomagającą zaktywizować zespół, jest wizualizacja ilości pracy pozostałej do wykonania w sprintcie. W tym celu niezbędne jest regularne określanie liczby godzin koniecznych do zakończenia rozpoczętych zadań i wizualizacja tej informacji w postaci tzw. wykresu wypalania (ang. *burn-down chart*).



Rys. 9. Wykres wypalania sprintu
Fig. 9. Sprint burnout chart

Zespół samodzielnie wyznacza te dane, bazując na aktualnym stanie prac i zawartości rejestru zadaniowego oraz przedstawia w postaci wykresu wypalania. Wykresy wypalania

sprintu są wykorzystywane przez zespół do monitorowania wykonania sprintu, gdyż na ich podstawie można ocenić bieżący stan prac, a także prognozować osiągnięcie celu sprintu. Przykładowy wykres wypalania przedstawiony na rysunku 9 podaje dla poszczególnych dni sprintu liczbę godzin potrzebnych do wykonania zobowiązania podjętego w drugiej części spotkania planistycznego.

7.3.3. *Adaptacja sprintu*

Wyniki kontroli i monitorowania sprintu mogą powodować konieczność zmiany rejestru zadaniowego, co będzie wymagało przeplanowania przebiegu prac w sprincie. Jak wspomniano wcześniej, adaptacja sprintu ma również swój przejaw w bieżącej aktualizacji liczby godzin pozostałych do wykonania w sprincie. Rejestr zadaniowy i wykres wypalania są zatem na bieżąco aktualizowane w trakcie części wykonawczej sprintu.

7.4. **Przegląd sprintu**

Zespół w trakcie sprintu uzyskuje ogromną swobodę co do sposobu wywiązania się z podjętego zobowiązania – doboru zadań prowadzących do jego wypełnienia, wyboru technologii, kolejności wykonania prac, również kontrola przebiegu tego procesu w ogromnej mierze leży po stronie zespołu. Zarządzanie w ten sposób nie byłoby efektywne bez wprowadzenia do procesu elementu inspekcji i adaptacji następującego po wykonaniu sprintu. W metodyce Scrum te elementy cyklu produkcyjnego znajdują reprezentację w postaci przeglądu oraz retrospektywy sprintu.

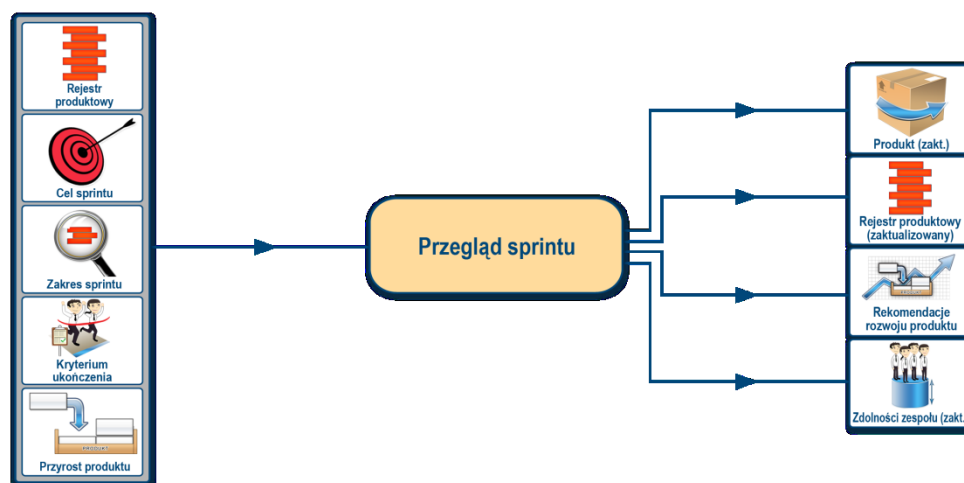
Przegląd sprintu (ang. *sprint review*) jest dokonywany pod koniec sprintu, zwykle w ostatnim jego dniu. Przegląd sprintu polega na sprawdzeniu w jakim stopniu został zrealizowany cel sprintu, które z zadań postawionych w sesji planistycznej zostały wykonane i dyskusji dotyczącej tego, co proponowane przez zespół rozwiązania (lub ich brak) oznaczają dla produktu i dalszego przebiegu prac.

Podczas przeglądu sprintu są prezentowane te cechy produktu, które udało się zrealizować w sposób kompletny, umożliwiając (potencjalne) przekazanie ich odbiorcy końcowemu (ang. *potentially shippable product increment*). Właściciel produktu, często w obecności interesariuszy, wspólnie z zespołem projektowym identyfikuje, co zostało wykonane, a co nie. W ocenie stopnia wykonalności poszczególnych cech produktu kierują się uzgodnionym wcześniej kryterium ukończenia. Zwykle substytuty (prototypy) oraz częściowo wykonana praca (na przykład dokumentacja projektowa bez towarzyszącego jej kodu źródłowego, niekompletny kod źródłowy, kod źródłowy pozbawiony dokumentacji, nieprzetestowany odpowiednio produkt) jako elementy z definicji niespełniające kryterium ukończenia nie podlegają prezentacji. Można zatem stwierdzić, że podstawową miarą oceny postępu

prac oraz przesłanką do dokonywania jakiegokolwiek korekty planów projektu, jest działające oprogramowanie (lub jego brak).

Podczas spotkania następuje określenie stopnia, w jakim osiągnięto zaplanowany cel sprintu, cel wydania oraz cel nadrzędny przedsięwzięcia. Zespół otrzymuje od właściciela produktu informację zwrotną odnośnie do oceny wykonanej pracy oraz skuteczności pracy zespołu.

Przegląd sprintu (rys. 10) dostarcza istotnych wiadomości potrzebnych do planowania kolejnych sprintów. Dyskutowane są kierunki rozwoju produktu w kolejnych sprintach i związane z tym zmiany w rejestrze produktowym. Właściciel produktu dokonuje oceny daty zakończenia projektu, wykorzystując w tym celu empirycznie wyznaczoną zdolność produkcyjną zespołu tzw. prędkość (ang. *velocity*) i dokonuje zmian w rejestrze produktowym. Obrazowo rzecz ujmując, rejestr produktowy po każdym zakończeniu sprintu jest redukowany poprzez usuwanie z niego elementów wykonanych w trakcie sprintu. Trzeba jednak pamiętać, że może on jednocześnie przyrastać, jeśli pojawi się potrzeba wprowadzenia nowych bądź zmodyfikowania już zaimplementowanych cech produktu. W szczególności może również zapaść decyzja o przerwaniu projektu czy to na skutek wcześniejszej niż zakładana realizacji głównych założeń, czy przeciwnie z powodu braku oczekiwanych postępów. W przypadku metodyki Scrum mamy więc do czynienia nie tylko z przykładem iteracyjnego i przyrostowego procesu rozwoju oprogramowania, ale również z przykładem planowania adaptacyjnego, realizowanego iteracyjnie.



Rys. 10. Schemat procesu przeglądu sprintu

Fig. 10. Sprint review process schema

Prezentacja wyników sprintu w postaci działającego oprogramowania ma na celu ocenę i ewentualne poszerzenie spektrum możliwości wykorzystania produktu przez odbiorcę (reprezentowanego przez właściciela produktu lub faktycznych odbiorców uczestniczących w przeglądzie sprintu), umożliwiając mu empiryczne zapoznanie się dokładnie z tym, co znajdzie się w przyszłym, eksploatowanym przez niego systemie. Przekazane przez niego

uwagi mogą być w takim przypadku szczególnie trafne i zgodne z jego wyobrażeniem ewentualnych zmian. Przeglądu sprintu nie należy zatem traktować zbyt formalnie, raczej chodzi tutaj o rzetelną sesję roboczą, której celem jest uzyskanie przez obie strony użytecznej informacji zwrotnej i wypracowanie zasad postępowania w kolejnych etapach prac.

7.5. Retrospektywa sprintu

Retrospektywa sprintu (ang. *sprint retrospective*) służy ocenie skuteczności działania zespołu i ustaleniu, jakie modyfikacje w kolejnych sprintach zwiększą jego produktywność (rys. 11). Scrum Master zachęca zespół do dokonania przeglądu funkcjonowania procesów i praktyk Scruma z szerszej perspektywy, w celu dokonania identyfikacji problemów i korekty w następnym lub kolejnych sprintach.

Warto zwrócić uwagę na fakt, iż właściciel produktu zwykle nie uczestniczy w retrospektywie, podobnie jak wyższe kierownictwo, co daje okazję do przeprowadzenia swobodnej i efektywnej dyskusji. Retrospektywa jest postrzegana jako „faza adaptacyjna” w pracy zespołu, znacznie poprawiając komunikację w zespole i stymulując jego samoświadomość i rozwój.

Podczas retrospektywy zespół dokonuje podsumowania zadań, których realizacja powiodła się lub które sprawiały problemy, omawia skuteczność wykorzystanych narzędzi, technik i procedur. Dyskutuje ich skuteczność w kontekście osiągniętych rezultatów, kierując się informacją zwrotną przekazaną przez właściciela produktu podczas przeglądu sprintu. W wyniku spotkania powstają konkretne propozycje poprawy.

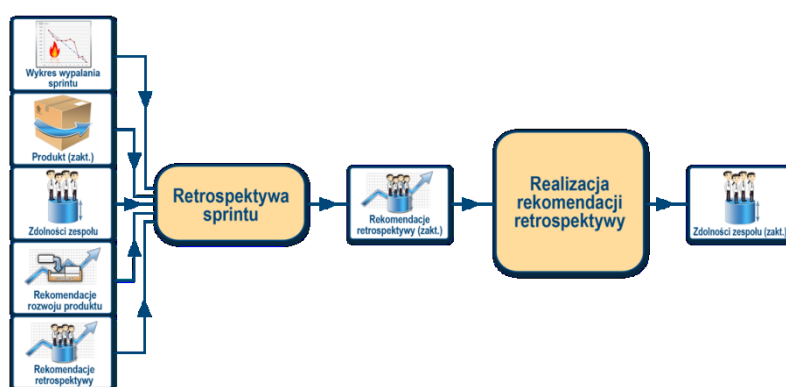
W retrospektywie bierze udział cały zespół wykonawczy, więc ocena jest dokonywana z różnych perspektyw (analizy, projektowania, programowania, testowania). Uczestnicy zespołu dokonują analizy pod kątem:

- Dotrzymania deklarowanych czasów realizacji.
- Sprawności działania zespołu, w szczególności: relacji międzyludzkich, składu zespołu, procesu wytwórczego, organizacji i efektywności spotkań, metod komunikacji.
- Adekwatności i efektywności wykorzystywanych narzędzi i technik.
- Jakości produktu (na podstawie stanu faktycznego i pożądanej postaci wyrażonej poprzez kryterium ukończenia).

Oprócz identyfikacji kwestii problematycznych, zespół dyskutuje propozycje rozwiązań, których zastosowanie w kolejnych sprintach może zapobiec powtórzeniu się niepowodzeń. Rekomendacje te są następnie oceniane i porządkowane pod kątem możliwości i zasadności ich wdrożenia oraz kolejno i stopniowo wdrażane.

7.6. Realizacja rekomendacji retrospektywy

Jak zostało to zasygnalizowane w poprzednim rozdziale w trakcie retrospektywy są definiowane działania bieżące i długoterminowe, dotyczące poprawy procesów, oraz strategia ich realizacji w kolejnych sprintach. Częściowo mogą to być działania, których realizacji zespół może podjąć się już w kolejnym sprincie, częściowo osiągnięcie celów działań naprawczych może wymagać środków, do dysponowania którymi zespół nie jest upoważniony (na przykład zakup narzędzi). Wówczas do bieżących działań zespołu może należeć identyfikacja tych narzędzi, przeprowadzenie ich ewaluacji, dokonanie analizy kosztów wdrożenia czy przygotowanie zamówienia dla przełożonych.



Rys. 11. Schemat procesów retrospektywy sprintu i realizacji rekomendacji retrospektywy
Fig. 11. Sprint retrospective and execution of retrospective recommendation schema

W wyniku realizacji działań wynikających z rekomendacji retrospektywy następuje zmiana – potencjalnie na lepsze – zdolności produkcyjnych zespołu (rys. 11).

8. Zestawienie składowych i pojęć metodyki SCRUM

W literaturze polskiej są stosowane różne nazwy pojęć związanych z metodyką Scrum, autorzy przeglądali słowniki, różne inne pozycje literaturowe i strony internetowe, dotyczące rozważanych zagadnień. W wyniku takiego przeglądu wystąpiono z próbą uporządkowania nazewnictwa polskiego, którą zawarto w tabeli 1.

Tabela 1

Słownik ważniejszych pojęć związanych z metodyką Scrum

<i>Nazwa angielska</i>	<i>Proponowana (zastępcza) nazwa polska</i>
Universal principles	Uniwersalne zasady
Scrum	Scrum (młyn)
agile	zwinny
agile manifesto	Manifest Zwinnego Wytwarzania Oprogramowania, Manifest Agile
normative methodology	metodyka normatywna

framework	szkielet pracy
Events (time boxes)	Zdarzenia
strategy timebox	horyzont czasowy strategii
release timebox	horyzont czasowy wydania
sprint timebox	horyzont czasowy sprintu
sprint planning meeting – I part	spotkanie planistycznie sprintu – część I
sprint planning meeting – II part	spotkanie planistycznie sprintu – część II
daily Scrum, daily Scrum meeting	dzienny Scrum, codzienne spotkanie zespołu Scrum
sprint review meeting	spotkanie przeglądu sprintu
sprint retrospective meeting	spotkanie retrospektywy sprintu, spotkanie retrospektywne sprintu
executing sprint part	wykonawcza część sprintu
Artifacts	Artefakty
product	produkt
product backlog	rejestr produktowy (zaległości produktowe)
definition of ready	kryterium gotowości
team capabilities	zdolności zespołu
definition of done	kryterium ukończenia
sprint goal	cel sprintu
sprint scope (committed backlog)	zakres sprintu (uzgodniony rejestr)
sprint backlog	rejestr zadaniowy sprintu
sprint burndown chart	wykres wypalania sprintu
product increment	przyrost produktu
product development recommendations	rekomendacje rozwoju produktu
retrospective recommendations	rekomendacje retrospektywy
Roles	Role
Scrum master	Scrum master (opiekun zespołu, trener zespołu, mistrz młyna, mistrz ceremonii, młynarz, mistrz, strażnik młyna)
product owner	właściciel produktu
stakeholders	interesariusze (osoby zainteresowane)
team	zespół
team member	członek zespołu
Processes	Procesy
sprint goal and scope planning	planowanie celu i zakresu sprintu
sprint task planning	planowanie zadań sprintu
product development	rozwój produktu
sprint control	kontrola sprintu
sprint monitoring	monitorowanie sprintu
sprint adaptation	adaptacja sprintu
sprint review	przeгляд sprintu
sprint retrospection	retrospektywa sprintu
retrospective recommendations realization	realizacja rekomendacji retrospektywy
Other vocabulary	Pozostałe słownictwo
user story	historyjka (historia, opowiadanie) użytkownika
story points	punkty historyjki
task board	tablica zadań
team velocity	prędkość zespołu
team capacity	zasobność zespołu

9. Podsumowanie

W przedsiębiorstwach bazujących na zarządzaniu wiedzą dąży się do uzyskania wiedzy wartościowej na bazie posiadanej rozbudowanej i nieuporządkowanej informacji. W przypadku zarządzania projektami informatycznymi istotne jest, by można było oceniać różne metodyki i ich elementy w celu dostosowania ich do potrzeb firm.

Rozwiązanie problemu wiarygodnego wyznaczenia własności metodyk polega na opracowaniu ontologii dla wybranych metodyk zarządzania – bazując na wspólnym metamodelu. W celu dokonania oceny przydatności ontologii metodyk przedstawiano uzyskane rozwiązania ekspertom dziedzinowym w formie opisowej przy zastosowaniu konceptów metamodelu. Głównym założeniem było opracowanie opisu niepowodującego zmiany samej metodyki.

Autorzy dokonali wielu wdrożeń metodyki Scrum w przedsiębiorstwach, przeprowadzili wiele dyskusji z praktykami z przedsiębiorstw informatycznych w trakcie prezentowania wyników na różnych seminariach. Weryfikacji opisu dokonywano w trakcie nauczania na studiach podyplomowych z zakresu zarządzania projektami oraz podczas szkoleń prowadzonych dla pracowników firm informatycznych.

W wyniku tych dyskusji uzyskano w dużej mierze akceptację zaproponowanych rozwiązań. Autorzy stwierdzili, że podstawowym problemem jest brak wcześniejszego opisu metodyki Scrum o charakterze standardu. W związku z tym występują podzielone zdania co do kwestii, czy dane rozwiązanie jest elementem metodyki czy nie.

Podjęte badania i przeprowadzone kroki praktyczne wskazują na potrzebę systematyzacji opisu rozwiązań. Rozwiązania takie umożliwią szybszy dostęp do wartościowej wiedzy, umożliwią wiarygodne i łatwe porównanie metodyk. Niniejsze opracowanie, zdaniem autorów, stanowi systematyczny opis metodyki Scrum z uporządkowaniem jej niewyspecyfikowanych elementów.

BIBLIOGRAFIA

1. Schwaber K., Sutherland J.: Scrum. Przewodnik po metodyce. 2010, <http://www.scrum.org/scrumguides>.
2. Decision Theory and Decision Behaviour: Normative and Descriptive Approaches. Springer, 1989.
3. Agile Manifesto, <http://www.agilemanifesto.org>
4. Takeuchi H., Nonaka I.: The New New Product Development Game. Harvard Business Review, January-February, 1986.

5. Ćwiklicki M., Włodarek T.: Metodyka Scrum w Polsce w świetle badań. *Nauka i Gospodarka*, nr 3 (6), 2010, s. 6÷11.
6. Ćwiklicki M., Jabłoński M., Włodarek T.: Samoorganizacja w zarządzaniu projektami metodą Scrum. Wydawnictwo Mfiles.pl, Kraków 2010.
7. Schwaber K.: *Agile Project Management with Scrum*. Microsoft Professional, 2004,
8. Schwaber K., Beedle M.: *Agile Software Development with Scrum 202*. 2001.
9. Cohn M.: *Agile estimating and planning*. Pearson Education, 2006.
10. Agile Development Poster, <http://pm.versionone.com/AgilePoster.html>.
11. Scrum Alliance, <http://www.scrumalliance.org>.
12. Scrum.org, <http://www.scrum.org>.
13. State of Agile Survey, 5th annual State of Agile Software Development Survey, 2010, VersionOne Inc. http://www.versionone.com/pdf/2010_State_of_Agile_Development_-_Survey_Results.pdf.
14. Werewka J, Szwed P., Rogus G.: Integration of classical and agile project management methodologies based on ontological models, [w:] Łebkowski P (red.): *Production engineering in making*. AGH University of Science and Technology Press, Kraków 2010. s. 7÷28.
15. Szwed P., Werewka J., Rogus G.: Ontology based alignment of classic and agile project management for an IT enterprise. *Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej*, t. 19, seria *Technologie Informacyjne*, nr 8, 2010, s. 345÷350.
16. Szwed P., Rogus G., Werewka J.: Wykorzystanie ontologii do modelowania zwinnej metodyki zarządzania projektami Scrum. *Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej*, t. 19, seria: *Technologie Informacyjne*, nr 8, 2010, s. 339÷344.

Wpłynęło do Redakcji 30 września 2011 r.

Abstract

The article discusses practical issues related to the use of agile software development methodologies in project teams. To describe different methodologies of such type, a use of a meta model is proposed here. The article presents a systematic description of Scrum – the most popular agile methodology connecting project management and product development.

The paper discusses techniques for obtaining valuable knowledge from extensive and unstructured information. It also postulates that a problem of a credible methodology properties designation requires stable ontologies for it. It is important to be able to evaluate

different methodologies and their components in order to adapt them to the needs of companies.

Additionally, the article orders a number of solutions unspecified to full extent in Scrum, evaluating their usefulness in practice at the same time. The presented solutions are the result of research, including numerous simulations and experiments with real teams and in specific commercial ventures..

During the studies, a large number of Scrum implementations in enterprises were made, and numerous experiments and discussions with many managers of commercial agile projects were conducted.

This paper provides a systematic description of Scrum with a strong rearrangement of its unspecified elements. It also includes accurate Scrum properties and elements description, as a starting point for further considerations. Special emphasis was placed on the processes analysis involved in a software release creating. The process of monitoring and possible options of adapting the Sprint (the main stage of manufacturing defined in Scrum) were considered broadly.

Considerations of the basic postulates of agile methodologies were also carried out. Scrum was interpreted here as an empirical process, but also as a normative methodology for determining the rigid system of conduct in an IT project work.

The developed solutions were presented to a large group of respondents, receiving quite a high level of acceptance acceptance.

Adresy

Jan WEREWKA: AGH Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Automatyki, werewka@agh.edu.pl.

Michał TUREK: AGH Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Automatyki, mitu@agh.edu.pl.

Tomasz WŁODAREK, PodDrzewem.pl, tomek@poddrzewem.pl.