



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
WYDZIAŁ ZARZĄDZANIA

KATEDRA Zarządzania Organizacjami, Kadrami i Prawa Gospodarczego

Praca dyplomowa licencjacka

*Zastosowanie metodyki Scrum w zarządzaniu projektami
informatycznymi na przykładzie*

*Usage of Scrum methodology in IT projects management on an
example*

Autor:	<i>Aleksander Gromala</i>
Kierunek studiów:	<i>Zarządzanie</i>
Opiekun pracy:	<i>dr Łukasz Wacławik</i>

Kraków, 2020



AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Wydział Zarządzania

DZIEKANAT

Aleksander Gromala
imię i nazwisko autora pracy

kierunek: Zarządzanie / ~~Zarządzanie i inżynieria produkcji~~ / ~~Informatyka i ekonometria~~ *
typ studiów : pierwszego / ~~drugiego~~ stopnia *
studia ~~stacjonarne~~ / niestacjonarne*
* niepotrzebne skreślić

Zastosowanie metodyki Scrum w zarządzaniu projektami informatycznymi na przykładzie

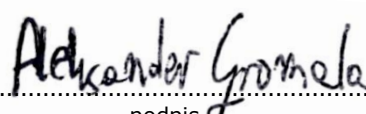
tytuł pracy dyplomowej

OŚWIADCZENIE AUTORA PRACY DYPLMOWEJ

Oświadczam, że dokumentacja pracy dyplomowej nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz. U. z 2006 r. Nr 90 poz. 631 z późniejszymi zmianami) oraz dóbr osobistych chronionych prawem cywilnym. Nie zawiera ona również danych i informacji, które uzyskałem/am w sposób niedozwolony. Wersja elektroniczna pracy dołączona przeze mnie na nośniku CD/DVD jest w pełni zgodna z wydrukiem pracy.

Zaświadczam także, że niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów wyższej uczelni lub tytułów zawodowych.

24.09.2020
data


.....
podpis

Spis treści

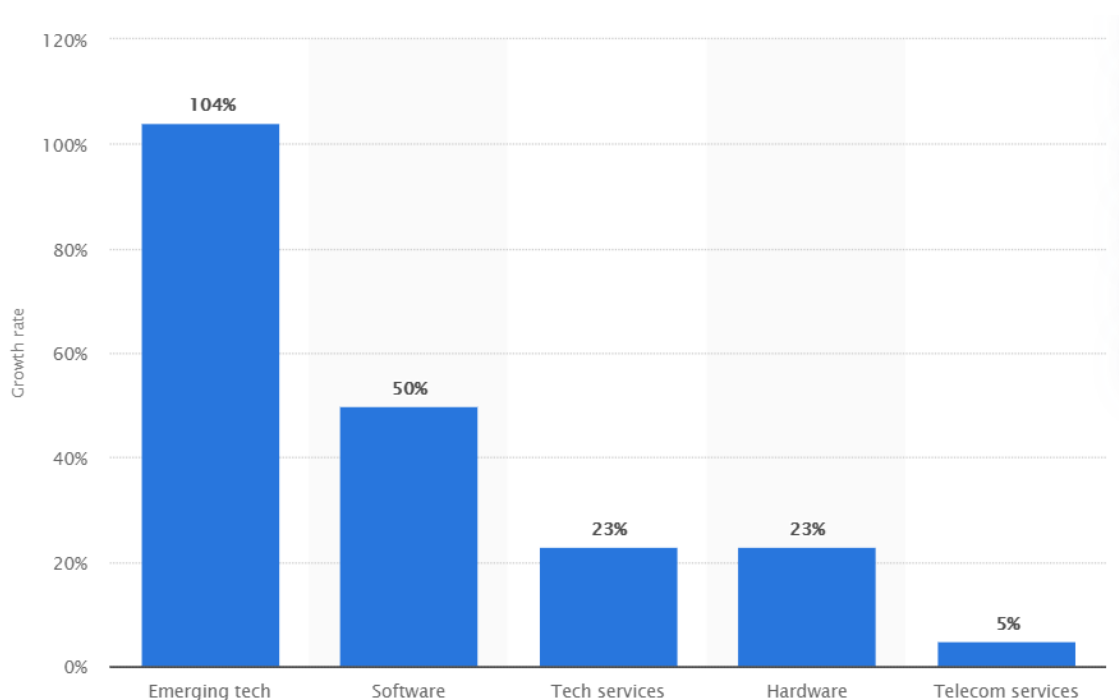
1	Wstęp do zarządzania projektami	3
1.1	Dynamiczny rozwój branży IT.....	3
1.2	Początki zarządzania projektami.....	4
1.3	Definicja projektu.....	7
1.4	Współczesne wyzwania zarządzania projektami	7
1.5	Zarządzanie projektami a branża IT	11
1.6	Cykl życia oprogramowania	12
2	Charakterystyka popularnych metod zarządzania projektami informatycznymi.....	17
2.1	Zarządzanie tradycyjne	19
2.1.1	Waterfall.....	19
2.1.2	PRINCE2.....	20
2.2	Zarządzanie zwinne.....	21
2.2.1	Wprowadzenie do Scrum – najpopularniejszej metody zwinnej.....	21
2.2.2	Programowanie ekstremalne.....	22
2.3	Inne metody wykorzystywane w IT	23
2.3.1	Model V	24
2.3.2	Kanban	25
3	Scrum – ogólny zarys i struktura	27
3.1	Role	28
3.1.1	Właściciel Produktu (Product Owner).....	29
3.1.2	Scrum Master	30
3.1.3	Zespół Deweloperski	31
3.2	Zdarzenia (Scrum Events).....	32
3.2.1	Sprint	33
3.2.2	Planowanie Sprintu (Sprint Planning)	34
3.2.3	Dzienny Scrum (Daily scrum).....	35
3.2.4	Przegląd Sprintu (Sprint Review).....	36
3.2.5	Retrospektywa Sprintu (Sprint Retrospective)	38
3.3	Artefakty	39
3.3.1	Rejestr Produktu (Product Backlog)	39
3.3.2	Rejestr Sprintu (Sprint Backlog)	42

3.3.3	Przyrost Produktu (Product increment)	43
3.4	Porównanie metod Waterfall i Scrum	44
4	Zastosowanie metody Scrum na przykładach.....	46
4.1	Scrum w cyklu życia oprogramowania w firmie z branży samochodowej.....	47
4.2	Studium przypadku: Tworzenie systemu IT dla Dutch Railways (firma Xebia) – przykład zmiany Waterfall na Scrum.....	52
5	Podsumowanie i wnioski	55
5.1	Zalety Scrum	55
5.2	Krytyka Scrum	56
5.3	Metody zwinne czy klasyczne?.....	57
	Spis rysunków i tabel	60
	Bibliografia	61

1 Wstęp do zarządzania projektami

1.1 Dynamiczny rozwój branży IT

W ciągu ostatnich dekad można było zaobserwować znaczący wzrost dynamiki rozwoju branży IT. Coraz większy popyt na nowe produkty i usługi oraz obecność coraz większej ilości konkurujących ze sobą podmiotów na rynku wymusza na firmach podejmowanie odpowiednich środków w celu zapewnienia, że kluczowe projekty zostaną pomyślnie i na czas ukończone. W związku z tym zarządzanie projektami odgrywa coraz większą rolę w dzisiejszych przedsiębiorstwach technologicznych.



Rysunek 1: Prognozy wzrostów branży IT w latach 2018-2023 z uwzględnieniem podziału na sektory¹

Jak widać na powyższym wykresie w najbliższych latach trend wzrostowy branży będzie zachowany. Co ważne, znacząco zwiększy się zapotrzebowanie na oprogramowanie, bo aż o 50%. Przewidywania wskazują, że największy wzrost osiągnie rynek związany

¹ Statista, IT industry growth rate forecast worldwide from 2018 to 2023, <https://www.statista.com/statistics/967095/worldwide-it-industry-growth-rate-forecast-segment/> (dostęp: 10.08.20)

z nowymi technologiami (robotyzacja, rozwój sztucznej inteligencji, tworzenie w pełni autonomicznego samochodu, praktyczne wykorzystanie technologii Virtual Reality). Wraz z rozwojem tych branż można spodziewać się wzrostu zapotrzebowania na projekty, które będą wymagały od kadry menedżerskiej odpowiedniego przygotowania i podejścia.

Zgodnie z raportem przygotowanym przez PMI na temat trendów na rynku pracy prognozuje się, że sektor IT będzie za drugim co do wielkości pod względem zapotrzebowania na menedżerów projektu w latach 2017-2027².

Ponadto, aktualny raport Głównego Urzędu Statystycznego pokazuje, że w Polsce w 2017 roku w sektorze ICT (technologii informacyjno-komunikacyjnych) działało 2230 firm zatrudniających 10 osób lub więcej. Spośród tych firm blisko 75% stanowiły przedsiębiorstwa oferujące usługi informatyczne. Dodatkowo raport pokazuje stały trend corocznego wzrostu nakładów ponoszonych na działalność B+R. W 2017 roku było to 5,9% względem roku poprzedniego³.

1.2 Początki zarządzania projektami

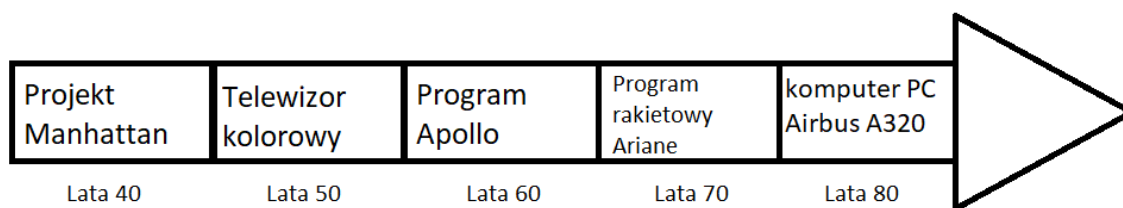
Naukowe podejście do zarządzania projektami swój początek miało w wojskowości. Jako początek uważa się pracę nad metodami planowania i kontroli w 1942 w USA na potrzebę projektu Manhattan. Mając na uwadze tajny charakter projektu, metody przez długi czas nie były upubliczniane, natomiast wykorzystywano je dalej przy kolejnych strategicznych amerykańskich projektach, w tym przy Programie Apollo w latach 60.

Prace nad rozwojem teorii były kontynuowane w kolejnych latach. Tempo rozwoju zostało znacząco przyspieszone dzięki pojawieniu się wydajniejszych komputerów w latach 80 i 90. W ostatnich latach teoria zarządzania projektami została wzbogacona także o zagadnienia personalne i kulturowe⁴.

² Job Growth and Talent Gap Report 2017-2027, PMI, USA 2017, s. 3

³ Społeczeństwo informacyjne w Polsce. Wyniki badań statystycznych z lat 2014-2018, GUS, Szczecin 2018, s. 13

⁴ Trocki M.: Zarządzanie projektami, PWE, Warszawa 2009. s 28-29



Rysunek 2: Wydarzenia, które miały historyczny wpływ na zarządzanie projektami⁵.

Jako główny czynnik mający znaczący wpływ na rozwój zarządzania projektami uważa się potrzebę planowania i kontroli realizacji wielkich projektów niemożliwych do zarządzania za pomocą tradycyjnych metod⁶. Niemniej jednak wymienia się również takie powody jak⁷:

- coraz krótszy cykl życia produktów, w wyniku czego przedsiębiorstwa są zmuszone znacząco przyspieszać proces tworzenia swoich ofert, w dodatku pamiętając o redukcji kosztów.
- rozwój technologiczny, który stwarza możliwość szybszej i tańszej produkcji a także znacząco poprawia komunikację poprzez sieci lokalne i globalne.
- potrzeba zapewniania coraz wyżej jakości oferowanych produktów, która stymuluje badania nad wpływem warunków pracy na efektywność pracowników i ich zdolność do zarządzania czasem,
- wzrost konkurencyjności, który zmuszą dzisiejsze przedsiębiorstwa do elastyczności i ciągłej czujności na zachodzące zmiany,
- możliwość realizacji projektów w zespołach rozproszonych po całym świecie dzięki globalizacji i rozwojowi sieci komputerowych,

⁵ Opracowanie własne na podstawie: Trocki M.: Zarządzanie projektami, PWE, Warszawa 2009. s. 28

⁶ Trocki M.: Zarządzanie projektami, PWE, Warszawa 2009, s. 28

⁷ Pawlak M.: Zarządzanie projektami, PWN, Warszawa 2006, s. 13-17

- panujący trend otwierania się dostawcy i klienta, dążenie do tworzenia trwałych relacji partnerskich z innymi przedsiębiorstwami oraz z klientami,
- koncepcja reengineeringu, polegająca na analizie podstawowych procesów przedsiębiorstwa (produkcyjnych, sprzedażowych, itd.) oraz przeprowadzania ich całkowitej restrukturyzacji w celu poprawy działań, ograniczenia kosztów, lepszego gospodarowania zasobami. Realizacja tej koncepcji zazwyczaj wymaga wielu projektów, które angażują wiele różnych działów, które do tej pory nie miały bezpośredniego kontaktu,
- w wyniku reengineeringu często dochodzi do decyzji o ograniczeniu liczby pracowników w celu ograniczeniu kosztów i poprawy efektywności. Pracownicy, którzy pozostali w przedsiębiorstwie muszą z reguły realizować więcej funkcji, a zespoły projektowe są mniejsze i bardziej samodzielne,
- delegowanie władzy, w celu osiągnięcia płaskiej struktury organizacyjnej oraz poprawy elastyczności, ale również niosące ryzyko chaosu, jeśli wszystkie komórki organizacyjne nie będą stosować spójnej koncepcji działania,
- zmiana orientacji przedsiębiorstw z produkcyjnej na marketingową. Postawienie w centrum zainteresowania klienta i jego potrzeby wymaga przygotowywania kreatywnych, zindywidualizowanych ofert.

Obecnie zarządzanie projektami jest powszechnie uważane za osobną i kompletną dziedzinę. Najbardziej popularnym i prestiżowym stowarzyszeniem zrzeszającym teoretyków i praktyków zarządzania projektami jest Project Management Institute (PMI). Kluczowe obszary i procesy zarządzania projektami z perspektywy wymagań branży IT zostaną szerzej omówione w rozdziale 2.

1.3 Definicja projektu

Przed przejściem do szczegółów warto najpierw poświęcić chwilę na jasne określenie co rozumiemy poprzez słowo *projekt*. Definicja w słowniku PWN określa projekt jako *plan działania lub wstępną wersję czegoś*⁸.

Inna definicja przedstawiona w literaturze naukowej mówi, że:

*Projekt jest to przedsięwzięcie, na które składa się zespół czynności, które są charakterystyczne przez to, że mają datę rozpoczęcia, specyficzne cele i limity, ustalone odpowiedzialności (obowiązki) realizatorów, budżet, rozkład czynności oraz datę ich ukończenia*⁹.

Z kolei PMI definiuje projekt jako:

*Tymczasowe przedsięwzięcie podejmowane w celu wytworzenia unikalnego wyrobu lub dostarczenia unikalnej usługi. Przez tymczasowość należy rozumieć więc, iż każdy projekt ma swój początek i koniec*¹⁰.

Należy mieć na uwadze, że projekt nie jest procesem. Jest jednorazowym, niepowtarzalnym przedsięwzięciem, które musi być na bieżąco kontrolowane. Projekty można podzielić według pochodzenia zlecenia (wewnętrzne, zewnętrzne), rozmiaru (małe, duże), stopnia skomplikowania (strategiczne, taktyczne, operacyjne), obszaru działań (inżynieryjne, produkcyjne, IT, naukowe)¹¹.

1.4 Współczesne wyzwania zarządzania projektami

Zarządzanie projektami jest dość rozległą dziedziną, na którą ma wpływ wiele czynników związanych z zarządzaniem ludźmi, czasem i ryzykiem. Poniższa grafika przedstawia

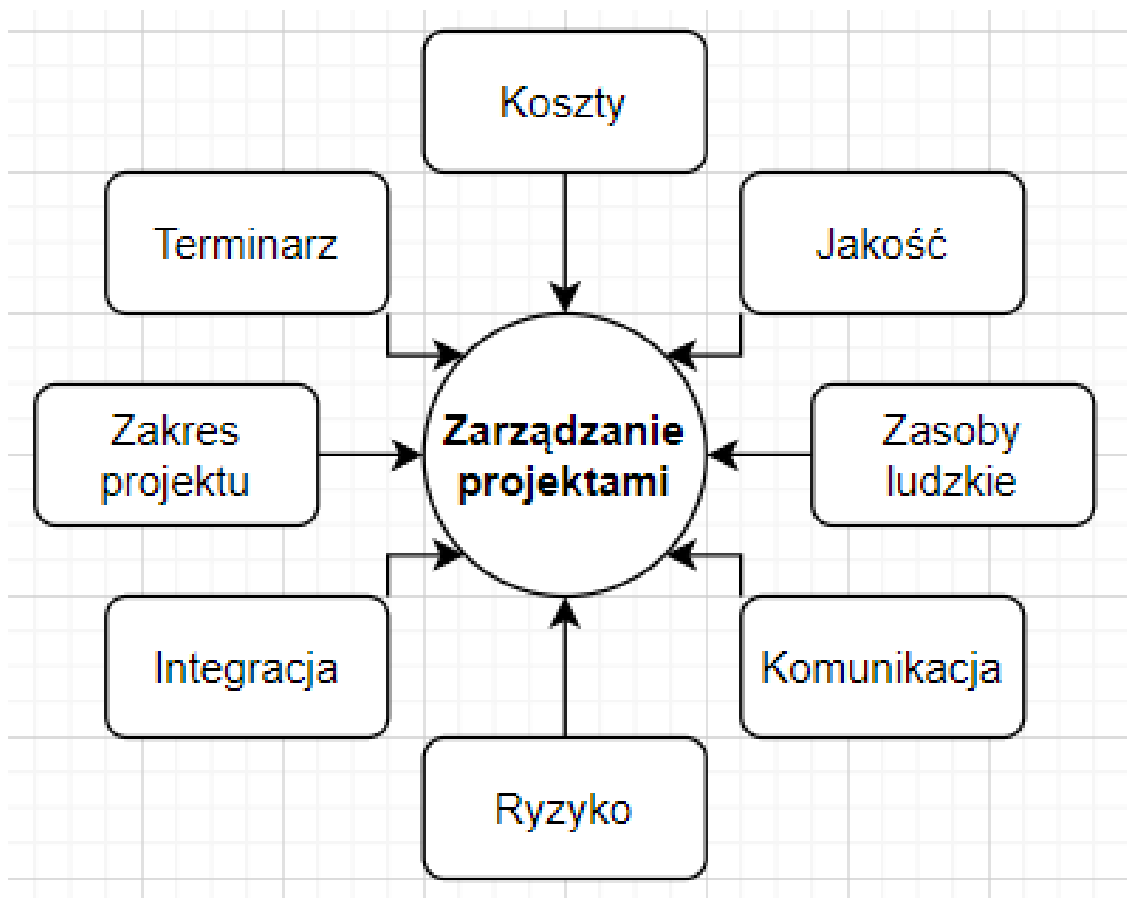
⁸ SJP, Definicja słowa „projekt”, <https://sjp.pwn.pl/> (dostęp 10.08.20)

⁹ Frączkowski K.: Zarządzanie projektem informatycznym, Politechnika Wrocławska, Wrocław 2003, s. 11

¹⁰ PMI, What is project management, <https://www.pmi.org/about/learn-about-pmi/what-is-project-management> (dostęp: 11.08.20)

¹¹ Lock D.: Podstawy zarządzania projektami, PWE, Warszawa 2009. s. 12-14

wszystkie czynniki, mające bezpośredni wpływ na organizowanie pracy projektowej nad produktem¹².



Rys. 2 Kluczowe obszary i procesy zarządzania projektami¹³.

Osoba odpowiedzialna za projekt musi być na tyle zorganizowana, żeby być w stanie pogodzić ze sobą wszystkie te aspekty, utrzymać pozytywną atmosferę w zespole w celu utrzymania efektywnej pracy nad produktem, który w dodatku usatysfakcjonuje klienta. Ponadto, menedżer projektu powinien posiadać cechy lidera, oraz umieć stworzyć takie środowisko, które nie dość, że sprzyja efektywnej pracy nad projektem, to sprawia, że ludzie chcą w nim pracować. Obszary, z którymi taka osoba powinna być zaznajomiona to kwestie: organizacyjne (relacje wewnątrz organizacji i zewnętrzne biznesowe),

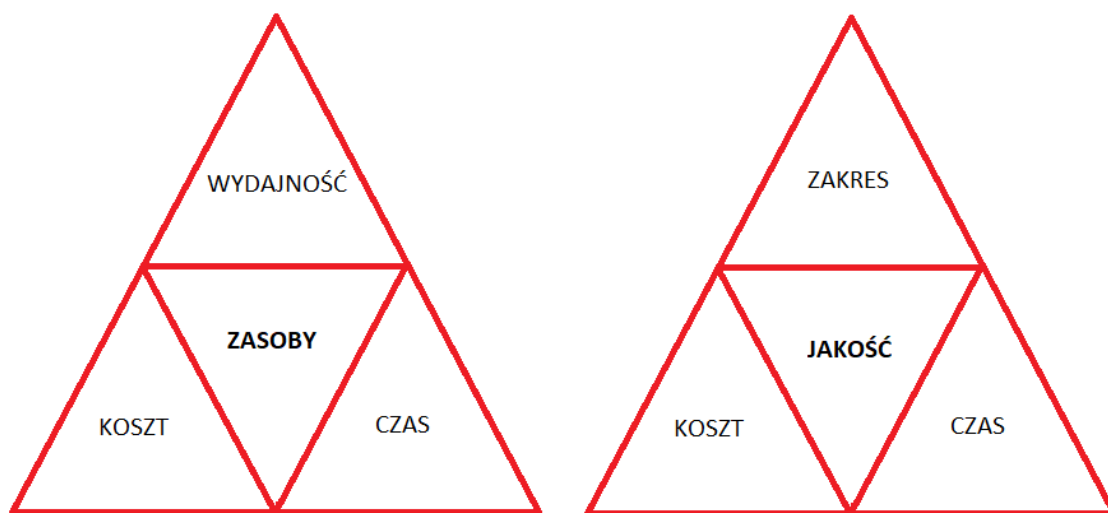
¹² Duncan W.R.: A Guide to the Project Management Body of Knowledge, PMI, USA 1996, s. 7

¹³ Opracowanie własne na podstawie: Duncan W.R.: A Guide to the Project Management Body of Knowledge, PMI, USA 1996, s. 7

zespołowe (motywacja, rozwiązywanie konfliktów), inżynieryjne (standardy jakościowe, wzorce projektowe, podstawowa wiedza techniczna) oraz procesowe (cel, KPI¹⁴, metodyka, spotkania, organizacja zadań)¹⁵.

Do tej pory powstało wiele opracowań wzajemnych relacji poszczególnych czynników, które bezpośrednio wpływają na projekt. Harold Kerzner przedstawił zależność pomiędzy czasem i kosztem w projekcie w postaci trójkąta ograniczeń projektowych. Obok czasu i kosztów dodatkową zmienną jest wydajność. Zdaniem Kerznera istotą zarządzania projektem jest kontrolowanie zasobów mając na uwadze wpływ tych trzech zmiennych. Zakładając stałe zasoby niemożliwa jest zmiana jednego parametru bez zmiany przynajmniej jednego innego parametru. Zmiany te zależą od charakteru projektu i są jego naturalną częścią¹⁶.

Na przestrzeni lat powstało kilka wersji trójkąta ograniczeń projektowych. Obecnie przyjmuje się inne parametry: zakres zamiast wydajności oraz jakość zamiast zasobów. Poniżej przedstawiono obie wersje trójkąta, oryginalną Kerznera i współczesną.



Rysunek 3: Dwa warianty trójkąta ograniczeń projektowych¹⁷

¹⁴ Key Performance Indicators – kluczowe wskaźniki efektywności stosowane do pomiaru stopnia realizacji celów projektu

¹⁵ Sierackiewicz M.: Technical Leadership, Helion, Gliwice 2016, s.14-17

¹⁶ Kerzner H.: Project Management: A systems approach to Planning, Scheduling and Controlling, Wiley, USA 2003, s. 625-626

¹⁷ Opracowanie własne na podstawie:

Kerzner H.: Project Management: A systems approach to Planning, Scheduling and Controlling, Wiley, USA 2003, s. 625-626;

Poprzez **Zakres** rozumie się wszystkie produkty i rezultaty¹⁸, w przypadku projektów IT mogą to być systemy informatyczne, programy użytkowe (produkty główne projektu) lub dokumentacja oprogramowania czy analiza wymagań (produkty dodatkowe). Innymi słowy jest to zbiór wszystkich prac, które należy wykonać, aby uzyskać produkt. Menedżer projektu powinien dbać, żeby zespół wykonywał tylko przewidziane prace a nie tracił czasu na nieuzgodnione i niewymagane czynności. Należy mieć na uwadze, że w trakcie trwania projektu bardzo często występują naciski na zmianę (zwiększenie lub zmniejszenie) zakresu prac zarówno ze strony klienta jak i samego zespołu projektowego¹⁹.

Koszt projektu to cały budżet potrzebny na zrealizowanie projektu. Projekt powinien być wyceniony z uwzględnieniem czynności projektowych, zasobów ludzkich i wymaganych narzędzi. Jeśli projekt jest zarządzany przy pomocy tradycyjnych metod takich jak model kaskadowy, wycena projektu jest przeprowadzana raz, na początku prac podczas analizy i planowania po zaakceptowaniu ostatecznej listy funkcjonalności produktu. W przypadku pracy w środowisku zwinnym, na przykład wykorzystując metodologię Scrum, należy liczyć się z częstymi zmianami wymagań poszczególnych funkcji w projekcie, co będzie miało znaczący wpływ na wahania całkowitej wyceny projektu.

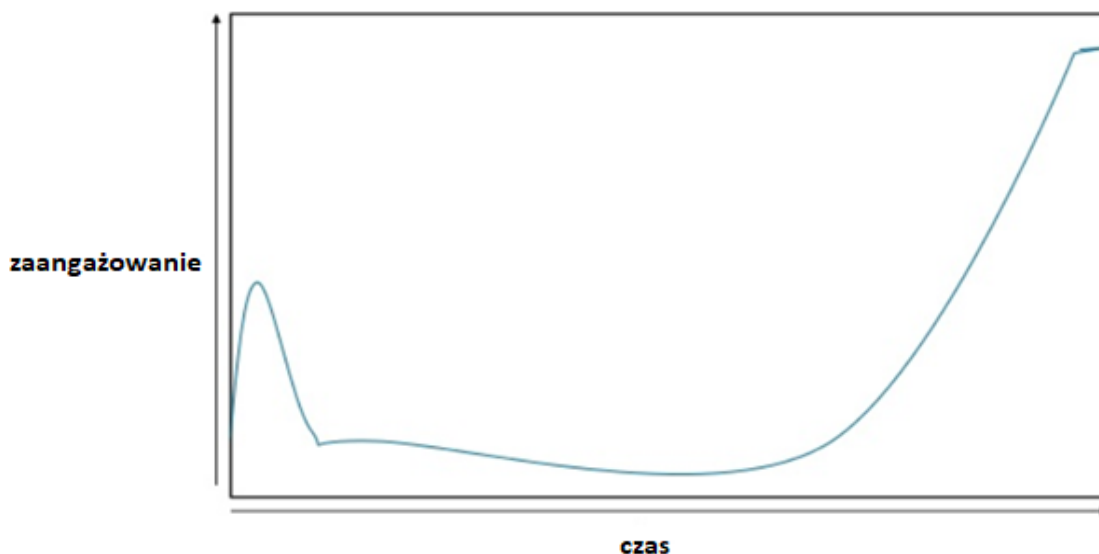
Ostatnim filarem piramidy jest **czas** poświęcony na realizację projektu. Podobnie jak w przypadku kosztu, powinno się go szacować, ponieważ to właśnie nakład pracy w czasie – roboczogodziny generują największy koszt. Przy estymacjach całościowego czasu należy pamiętać o uwzględnieniu dodatkowej rezerwy na potencjalne opóźnienia np. w postaci dodatkowych dziesięciu procent czasu przewidzianego na cały projekt. Przy opracowywaniu harmonogramu projektu należy pamiętać o częstym zjawisku

Zarządzanie Projektami Online, Trójkąt Projektowy, <https://zarzadzanieprojektami.online/2018/trojkat-projektowy> (dostęp 12.08.20)

¹⁸ Zarządzanie Projektami Online, Trójkąt Projektowy, <https://zarzadzanieprojektami.online/2018/trojkat-projektowy> (dostęp 12.10.20)

¹⁹ Kloze M.: Kluczowe aspekty zarządzania projektami – Deloitte, Prezentacja szkoleniowa Akademii Konsultingu 2009, s. 31-34

nazywanym syndromem studenta. Jest to tendencja pracowników do intensyfikowania pracy tuż po otrzymaniu zlecenia lub dopiero tuż przed terminem realizacji.



Rysunek 4: Wykres przedstawiający efekt tzw. syndromu studenta²⁰

1.5 Zarządzanie projektami a branża IT

Szybki rozwój branży IT i specyfika informatycznych projektów, różniących się od tych z klasycznych branż produktowo-usługowych zmusił menedżerów projektu do opracowania nowych metod kierowania projektami. Podstawowym aspektem, który wyróżnia projekt informatyczny jest ciężki do ustalenia zakres projektu. Jest to typowe dla często abstrakcyjnych systemów informatycznych, gdzie znane są tylko potrzeby klienta a sposób realizacji pozostaje do ustalenia. W pracy nad projektem związanym z tworzeniem oprogramowania naturalną rzeczą jest intensywna komunikacja pomiędzy zespołem projektowym a klientem²¹. Ma to na celu zapewnienie, że klient dostanie dokładnie taki produkt jaki potrzebuje. Z drugiej strony powoduje to konieczność częstej zmiany wymagań, które ewoluują podczas pracy nad produktem, a których źródłem mogą

²⁰ Korzonkowski K.: CCPM jako metoda planowania i kontroli projektów, Capgemini, Wrocław 2014, s. 7

²¹ O'Neill E.: User-Developer Cooperation in Software Development. Springer-Verlag, London 2001, s. 1-2

być czynniki zewnętrzne jak nagle zaistniałe zmiany na rynku czy wewnętrzne, na przykład problemy budżetowe.

Kierownik projektu IT musi wykazać się zarówno wysokimi umiejętnościami zarządzania zasobami ludzkimi jak i znajomością branży oraz związanych z nią aspektów technicznych, które pozwolą mu oszacować wymagany czas i budżet, dzięki czemu prace projektowe będą lepiej zaplanowane i kontrolowane²². Naturalne w branży IT jest wspomaganie się specjalistycznym oprogramowaniem, które wspiera pracę zespołową. Do najpopularniejszych przykładów należą narzędzia do zarządzania projektem, takie jak Microsoft Project, Jira (wykorzystywana często w metodyce Agile) oraz systemy kontroli wersji typu GitHub i GitLab, niezbędne, gdy wielu programistów pisze wspólny projekt²³.

1.6 Cykl życia oprogramowania

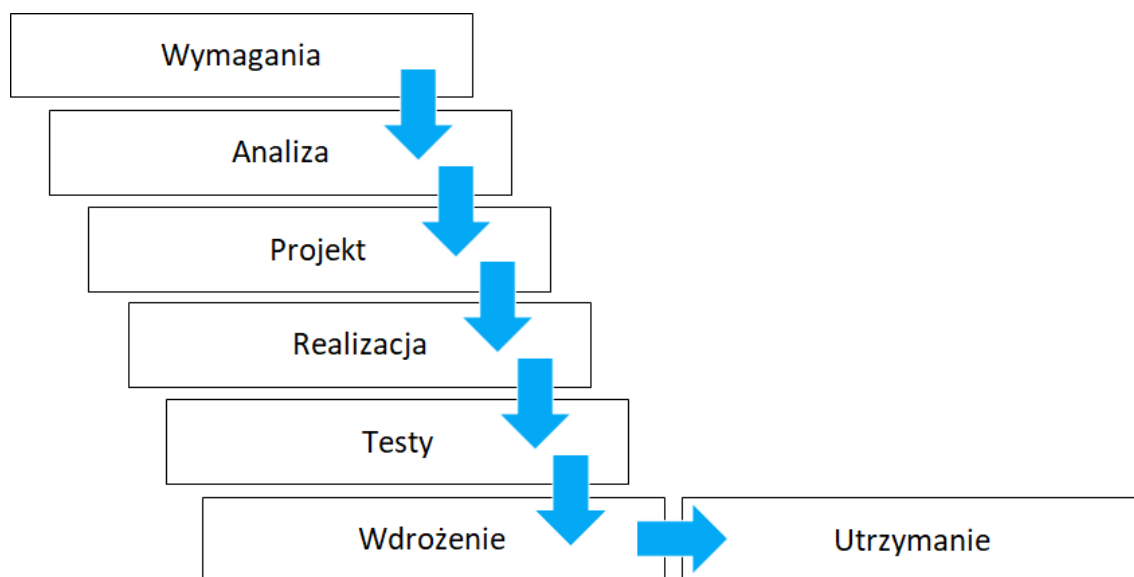
Zgodnie z definicją poprzez cykl życia oprogramowania rozumie się: *Okres czasu rozpoczynający się, kiedy pojawił się pomysł na oprogramowanie i kończący się, gdy oprogramowanie nie jest już dostępne do użytku. Zazwyczaj cykl życia oprogramowania zawiera fazę koncepcji, fazę wymagań, fazę projektowania, fazę implementacji, fazę testów, fazę instalacji i zastępowania, fazę wykorzystania produkcyjnego i pielęgnowania oraz czasami, fazę wycofania. Uwaga: te fazy mogą na siebie zachodzić lub mogą być wykonywane iteracyjnie*²⁴.

W latach 90 dominowało podejście kaskadowe (Waterfall). Założeniem było trzymanie się stałego, fazowego cyklu życia oprogramowania w procesie wytwórczym. Poniższy obrazek obrazuje kolejność etapów, charakterystycznych dla podejścia kaskadowego.

²² Dziurzańska A., Dżega D., Olejniczak W.: Polski menedżer projektu informatycznego, Akademia Ekonomiczna, Katowice 2007, s. 8

²³ Mielcarek K., Pławiak-Mowna A.: Wspomaganie realizacji grupowego projektu IT w oparciu o system SVN. Metody Informatyki Stosowanej, nr 2(27), Polska Akademia Nauk Oddział w Gdańsku, Szczecin 2011, s. 82

²⁴ Słownik wyrażen związanych z testowaniem, Definicja cyklu życia oprogramowania, <https://sjsi.org/slowo/cykl-zycia-oprogramowania> (dostęp 14.08.2020)



Rysunek 5: Przykład cyklu życia oprogramowania w podejściu tradycyjnym (model kaskadowy – Waterfall)²⁵

Takie podejście wiąże się z wystąpieniem licznych problemów, które mogą pojawić się na różnych etapach prac projektowych, powodując znaczące opóźnienia lub nawet porzucenie całego projektu. Między innymi są to:

- wymóg zebrania wszystkich wymagań i ustalenia ostatecznych założeń projektu już w początkowej fazie projektu,
- bardzo długi czas pomiędzy ustalaniem założeń projektu a ich dostarczeniem powoduje ryzyko zmiany potrzeb klienta np. w wyniku innej sytuacji rynkowej,
- wdrażanie wszystkich funkcjonalności na raz, co zwłaszcza w przypadku projektów programistycznych stwarza ryzyko zaistnienia problemów z integracją systemu

Wzrost zapotrzebowania na projekty informatyczne doprowadził do opracowania tzw. Manifestu Agile (Agile Manifesto). Jest to zbiór dwunastu wypunktowanych reguł pracy w podejściu zwinnym²⁶:

²⁵ Źródło: <https://agilegroup.pl/agile-mindset> (dostęp 16.08.2020)

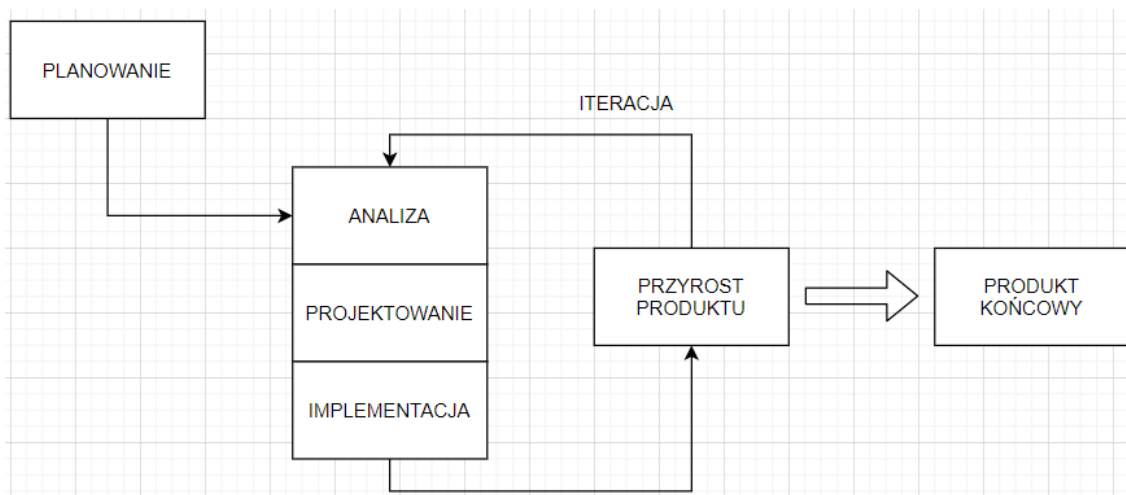
²⁶ Agile Manifesto, <https://agilemanifesto.org/iso/pl/principles.html> (dostęp 16.08.2020)

- 1) **zadowolenie klienta jako najwyższy priorytet** osiągnięte dzięki wczesnemu i ciągłemu wdrażaniu wartościowego oprogramowania,
- 2) **gotowość na zmiany** wymagań na każdym etapie projektu w celu zapewnienia klientowi konkurencyjności,
- 3) **częste dostarczanie funkcjonującego oprogramowania**, w około kilku tygodniowych odstępach czasu,
- 4) **ściśła współpraca zespołów deweloperskich i biznesowych** przez cały czas trwania projektu
- 5) **dbanie o wysoką motywację zespołu** poprzez stworzenie odpowiedniego środowiska pracy oraz zapewnianie wsparcia,
- 6) **rozmowa twarzą w twarz** zalecanym sposobem przekazywania informacji
- 7) **działające oprogramowanie** jako podstawowa miara postępu
- 8) **zrównoważony rozwój** dzięki utrzymaniu stabilnego tempa pracy na każdym szczeblu
- 9) skupienie się na **technicznej doskonałości i dobrym projektowaniu** zwiększa zwinność
- 10) dbanie o **prostotę** – minimalizowanie ilości koniecznej pracy
- 11) **samoorganizujące się zespoły** źródłem najlepszych rozwiązań architektonicznych, wymagań i projektów
- 12) organizacja **regularnych spotkań** zespołu w celu **analizy możliwości poprawy** oraz **dostrojenia i dostosowania swoich działań** do wyciągniętych wniosków

Mimo że powszechnie używa się określenia „metodyki zwinne”, twórcy Agile kładą nacisk, że to nie jest poprawne rozumowanie, ponieważ Agile to raczej sposób myślenia i odpowiedniego podejścia do projektu, wyrażony w powyższych zasadach a nie ściśle narzucona metoda. Scrum jest najpowszechniej używanym podejściem zwinnym w zarządzaniu projektami, których produktem jest oprogramowanie.

Cykl życia oprogramowania w podejściu zwinnym składa się z wielu iteracji, podczas których tworzony jest ustalony fragment gotowego produktu – jego jedna funkcjonalność zgodna z wybraną wcześniej specyfikacją. W efekcie końcowym powstaje zintegrowany

produkt końcowy. Takie rozwiązanie pozwala na wprowadzenie potencjalnych zmian do wymagań tworzonego systemu, nawet podczas trwania projektu.

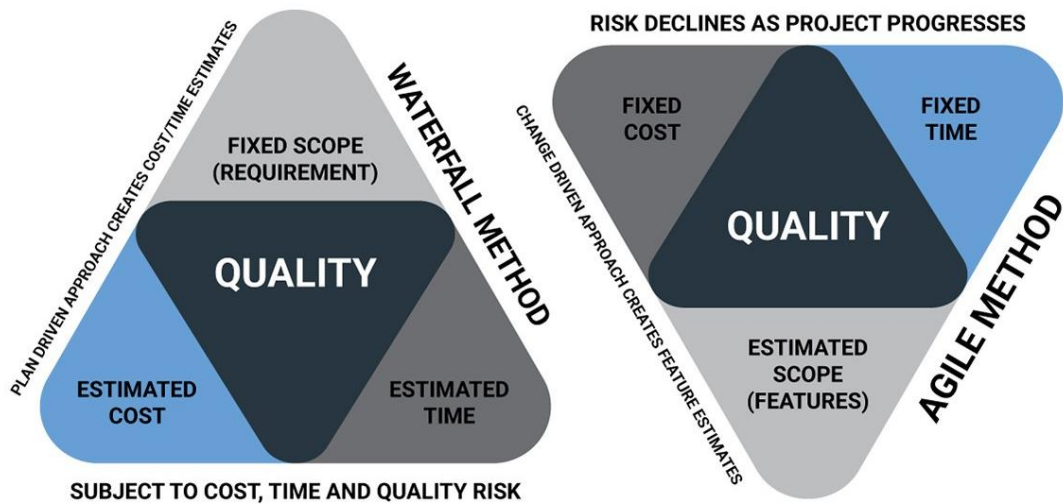


Rysunek 6: Typowy cykl życia oprogramowania w podejściu zwinnym²⁷

Z myślą o projektach prowadzonych w zgodzie z podejściem Agile powstała unikalna wersja już wspomnianego w poprzednim rozdziale Trójkąta ograniczeń projektowych. W tym przypadku trójkąt przedstawia się wierzchołkiem do dołu, ze względu na występujące różnice w porównaniu z tradycyjnym podejściem kaskadowym. Metodyka Waterfall bazowała na z góry określonym planie ze stałym zakresem rzeczy do zrobienia, który po uzgodnieniu pozwalał na przybliżone szacunki potrzebnego czasu i budżetu na realizację projektu. Natomiast w podejściu zwinnym zakłada się z góry ustalony budżet i stały czas na realizację projektu, które po uzgodnieniu pozwalają określić funkcjonalności produktu, jakie zespół jest w stanie stworzyć w takich ramach czasowych i finansowych. W trakcie trwania projektu podlegają one ciągłym szacunkom i zmianom. Należy pamiętać o ścisłym uczestnictwie interesariuszy projektu podczas negocjacji, bo ostatecznie to zadowolenie klientów jest na pierwszym miejscu i to oni powinni godzić się na zakładane założenia projektowe. Wraz z postępem prac w Agile jednocześnie

²⁷ Opracowanie własne na podstawie: IT Consulting, Cykl życia oprogramowania a wybór metodologii, https://it-consulting.pl/autoinstalator/wordpress/2011/03/22/co-wybrac-czyli-cykl-zycia-projektu-tworzenia-oprogramowania/#Wybor_wlasciwej_metodologii (dostęp: 17.08.2020)

zmniejsza się ryzyko a szansa na dostarczenie ostatecznej wersji produktu wzrasta. Podczas prac projektowych wymagania są elastyczne i mogą podlegać zmianom, jeśli takie wnioski zostaną wyciągnięte ze wspólnych dyskusji zespołu projektowego z klientem.



Rysunek 7: Porównanie trójkątów wyzwań projektowych w podejściu tradycyjnym i zwinym²⁸

Oba trójkąty w swoim centrum przedstawiają jakość, ponieważ niezależnie od podejścia czy strategii celem każdego zespołu powinno być dostarczenie produktu jak najwyższej jakości.

²⁸ Źródło: <https://www.baytechconsulting.com/how-do-you-estimate-an-agile-software-development-project/>

2 Charakterystyka popularnych metod zarządzania projektami informatycznymi

Istnieje wiele metodologii i strategii w zarządzaniu projektami. Wiele z nich powstało zanim jeszcze nastąpił rozwój cyfryzacji i zapotrzebowanie na zarządzanie projektami IT i wymagało odpowiedniego przystosowania do realiów nowej branży. Poza typową charakterystyką jak podział na podejścia klasyczne i zwinne, można wyróżnić pomocnicze metody, którymi można wspomóc procesy zarządzania projektem. Od menedżera projektu, jego wiedzy i dotychczasowego doświadczenia zależy jakie metodyki zostaną wybrane. Najczęściej jest ona wybierana w sposób intuicyjny²⁹. W niniejszym rozdziale zostaną przedstawione najczęściej rozważane przez menedżerów metody wraz z ich krótkimi charakterystykami. Metodyka Scrum, której niniejsza praca poświęca najwięcej miejsca zostanie szczegółowo przedstawiona w rozdziale 5.

Mając jako parametry znajomość celu i potencjalnego rozwiązania, poszczególne podejścia można odzwierciedlić na „Macierzy wyboru metodyki zarządzania projektem wg Roberta Wysockiego. Wyróżnia się cztery podejścia:

- Tradycyjne (TPM – Traditional Project Management)
- Zwinne (APM – Agile Project Management)
- Ekstremalne (xPM – Extreme Project Management)
- *Emertxe* (MPx – Emertxe Project Management)³⁰

²⁹ Orłowski C., Ziółkowski A.: Wsparcie doboru metod zarządzania , politechnika Gdańska, Gdańsk 2010, s. 199

³⁰ Nazwa *emrtxe* powstała z zapisania angielskiej nazwy podejścia *extreme* wspak.

		ROZWIĄZANIE	
		jasne	niejasne
CEL	niejasny	Podejście "emertxe"	Podejście ekstremalne
	jasny	Podejście tradycyjne	Podejście zwinne

Rysunek 8: Macierz wyboru metodyki zarządzania projektem wg. Wysockiego³¹.

Poprzez określenia celu i rozwiązania jako jasny/niejasny należy rozumieć w jak dużym stopniu są zdefiniowane wymagania projektowe i metody dojścia do rozwiązania. Taka macierz może pomóc menedżerom projektu w ustaleniu odpowiedniego podejścia przed przystąpieniem do realizacji projektu.

Podejście tradycyjne zakłada jasno określony cel oraz znany sposób na osiągnięcie go. W tym modelu obowiązuje prosty cykl życia projektu z etapami następującymi jeden po drugim. Na początku planuje się zakres wymagań, kosztorys, potrzebne zasoby a później następuje realizacja projektu i jego zakończenie³².

³¹ Opracowanie własne na podstawie: Wysocki R. K.: Effective Project Management: Traditional, Agile, Extreme, Wiley Publishing Inc., Indianapolis USA 2009, s. 8

³² Wysocki R. K., McGary R.: Efektywne Zarządzanie Projektami: Tradycyjne, zwinne, ekstremalne, wyd. 3, Helion, Gliwice 2005, s. 67

Podejście zwinne kładzie nacisk na rozwój funkcjonalności dostarczanego rozwiązania i spełnienie potrzeb klienta. Jasno określony jest cel projektu, natomiast sposób rozwiązania a tym samym budżet i harmonogram podlegają zmianom³³.

Podejście ekstremalne jest skrajnym przypadkiem podejścia zwinnego, w którym nawet cel projektu nie jest ściśle określony. Przykładem tej metody jest Programowanie Ekstremalne, które znajduje zastosowanie w innowacyjnych projektach badawczych i informatycznych³⁴.

Podejście emertxe jest przeciwieństwem podejścia zwinnego. Nie jest znany cel projektu, za to sposób rozwiązania jest określony. Choć koncepcja może się wydawać nielogiczna to jest coraz częściej wykorzystywana w przypadku prób znalezienia zastosowania dla nowej technologii³⁵.

Przykładami tradycyjnego podejścia są Waterfall i PRINCE2. Najpopularniejszą zwinną metodą jest Scrum, ale Programowanie ekstremalne jest nadal często rozważane. Są jeszcze metody, których nie można bezpośrednio skategoryzować w wyżej wymienionych podejściach jak Model V, czy Kanban., Są one często używane dodatkowo do wspierania zarządzania procesami. W kolejnych podrozdziałach zostaną przedstawione krótkie charakterystyki wyżej wymienionych metod.

2.1 Zarządzanie tradycyjne

2.1.1 Waterfall

Jest to model kaskadowy, propaguje podejście do zarządzania projektami polegające na wykonywaniu poszczególnych etapów projektowych jeden po drugim w stałej kolejności. W przypadku projektów IT te etapy to:

- **Planowanie** i specyfikacja wymagań systemu

³³ Wysocki R. K.: Effective Project Management: Traditional. Agile, Extreme, Wiley Publishing Inc., Indianapolis USA 2009, s. 327-329

³⁴ Wysocki R. K.: Effective Project Management: Traditional. Agile, Extreme, Wiley Publishing Inc., Indianapolis USA 2009, s. 351-353

³⁵ Wysocki R. K.: Effective Project Management: Traditional. Agile, Extreme, Wiley Publishing Inc., Indianapolis USA 2009, s. 353-354

- **Analiza** systemu
- **Projektowanie** systemu
- **Realizacja** założeń projektowych (tworzenie kodu i przypadków testowych, pamiętając o dokumentowaniu działań)
- **Testowanie** stworzonego systemu (sprawdzenie czy poszczególne elementy i całość systemu spełniają wszystkie założenia projektowe, walidacja i weryfikacja oprogramowania)
- **Wdrożenie i utrzymanie** systemu (instalacja systemu u klienta wraz z testami na jego środowisku, przygotowanie dokumentacji dla użytkownika, zapewnienie ewentualnego wsparcia w przypadku wystąpienia problemów)

Była to pierwsza metoda powszechnie wykorzystywana w projektach IT, Ciężko wskazać skąd się wziął pomysł na wykorzystanie metody Waterfall, zakłada się, że pomysł na implementację powstał w wyniku inspiracji dotychczasowymi branżami (budownicza, samochodowa czy jakakolwiek przemysłowa). Jest to najbardziej jasny i prostoliniowy model. Podstawową cechą jest to, że wszystkie wymagania i szkice projektu są wykonywane przed rozpoczęciem prac produkcyjnych nad stworzeniem oprogramowania. Dodatkowo, nie przewiduje się zmian projektowych w trakcie prac, ponieważ cofnięcie się w etapie projektu jest praktycznie niemożliwe i generuje zbyt duże koszty.

2.1.2 PRINCE2

Przykładem kolejnej, popularnej metody, która bazuje na podejściu tradycyjnym jest PRINCE2. Nazwa znaczy *P*ROjects *I*N *C*ontrolled *E*nvironments czyli *projekty w sterowalnym środowisku*. PRINCE2 powstał jako ustrukturyzowana metodyka zarządzania projektami, oparta na doświadczeniach pochodzących z tysięcy projektów³⁶. Zgodnie z PRINCE2 zarządzanie projektami to cztery zintegrowane elementy³⁷:

³⁶ OGC, PRINCE2 - Skuteczne zarządzanie projektami, TSO, Norwich UK 2009, s. 3

³⁷ OGC, PRINCE2 - Skuteczne zarządzanie projektami, TSO, Norwich UK 2009, s. 5-6

- Pryncypia (zasady) – nakazy przewodnie i dobre praktyki zarządzania projektem zgodnie z metodyką PRINCE2. Istnieje siedem pryncypiów i wszystkie muszą zostać zastosowane w projekcie.
- Tematy – opisują aspekty zarządzania projektem, które należy mieć na uwadze przez cały czas trwania projektu, jest ich siedem.
- Procesy – działania jakie trzeba podejmować krok po kroku w cyklu życia projektu.
- Środowisko projektu – PRINCE2 jako metoda posiadająca elastyczną strukturę powinien być dostosowany do środowiska projektu.

Metoda jest bardzo popularna w Wielkiej Brytanii. Menedżer projektu biorący udział w dużym, skomplikowanym projekcie w UK powinien się liczyć, że zalecane będzie zastosowanie tej metody³⁸. PRINCE2 kładzie nacisk na prowadzenie dokładnej dokumentacji działań oraz wykorzystywaniu dotychczasowego doświadczenia, nabytego w pracy nad wcześniejszymi projektami w celu zmniejszenia ryzyka.

2.2 Zarządzanie zwinne

2.2.1 Wprowadzenie do Scrum – najpopularniejszej metody zwinnej

Scrum został stworzony przez Kena Schwabera i Jeffa Sutherlanda, którzy po raz pierwszy zaprezentowali go w Austin podczas konferencji OOPSLA (obecnie Splash) w 1995 roku³⁹. Głównym celem było stworzenie metodyki zarządzania procesem powstawania nowych produktów zgodnej z zasadami zwinnymi. Typowymi zastosowaniami Scrum są⁴⁰:

- badania i rozpoznawanie potencjału rynków, technologii i produktów pod względem ich rentowności,

³⁸ Sobestiańczyk T.: Metodyka Zarządzania Projektami – PRINCE2, Politechnika Częstochowska, Częstochowa 2012, s. 17

³⁹ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 4

⁴⁰ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 4-5

- prace rozwojowych nad produktami (zwłaszcza oprogramowaniem w branży IT),
- wypuszczanie poprawek i ulepszeń produktów raz dziennie lub nawet częściej,
- rozwój i utrzymywanie rozwiązań bazujących na technologii Cloud, zarówno prywatnych jak i ogólnodostępnych,
- utrzymywanie i przedłużanie życia dotychczasowych produktów.

Na przestrzeni lat Scrum stał się podstawową metodyką zarządzania projektami w sektorach, gdzie wymagania końcowego produktu nie są nigdy do końca ustalone i wprowadzane są częste poprawki, a współpraca pomiędzy klientem a firmą odpowiedzialną za stworzenie produktu musi stać na bardzo wysokim poziomie. Dotyczy to zwłaszcza prac nad produktami branży IT lub pośrednio powiązanymi, w tym opracowywaniu rozwiązań dotyczących zarówno sprzętu (hardware) i oprogramowania (software), a także urządzeń na bazie mikrokontrolerów i ich oprogramowania (embedded software). Scrum obecnie coraz częściej pojawia się w branży automotive. Popularność wynika ze zwiększania poświęcanych środków wśród czołowych dostawców na badania i rozwój rozwiązań półautonomicznej i autonomicznej jazdy.

Podstawowym założeniem Scrum są małe zgrane zespoły, elastyczne i zdolne do szybkiej adaptacji, jeśli zajdzie taka potrzeba. Zespół jest najmniejszą komórką organizacyjną. Natomiast wiele małych zespołów wymieniających się efektami swojej pracy tworzy jedną dużą strukturę przypominającą model organizacji sieciowej⁴¹.

Scrum posiada jasno określone zasady dotyczące ról pracowników, etapów działań i dokumentowania pracy. Zostaną one dokładnie omówione w rozdziale 5.

2.2.2 Programowanie ekstremalne

Metoda opracowana pod koniec lat 90 XX wieku, obok Scruma jest jedna z najpopularniejszych metod Agile stosowanych w projektach IT⁴². Jest znacznie bardziej rygorystycznym podejściem do zarządzania projektami niż typowe metody zwinne, przez

⁴¹ Struktura organizacji, w której poszczególne komórki jest powiązana z każdą inną, nie posiada centrum ani stałych relacji.

⁴² Extreme Programming, Introduction to XP, <http://www.extremeprogramming.org> (dostęp: 25.08.2020)

co bywa klasyfikowana jako osobna kategoria – podejście ekstremalne⁴³. Programowanie ekstremalne największy nacisk kładzie na szereg założeń, między innymi na⁴⁴:

- stosowanie refaktoryzacji przy budowaniu systemu IT (projektowanie systemu w taki sposób, żeby pojedyncze zmiany nie zmieniały funkcjonalności),
- częste, małe wydania (częste aktualizacje serwera klienta o nowe zmiany),
- ciągła integracja (poprzez stosowanie systemów kontroli wersji),
- czterdziestogodzinny tydzień pracy (potrzeba nadgodzin jest wskaźnikiem problemów procesu projektowego i wymaga interwencji),
- współpraca z klientem (zachęta do obecności klienta w czasie pracy programistów),
- ustalenie stałych metafor do komunikacji programistów z klientem (w celu uproszczeniu wymiany informacji i łatwiejszego przekazywania myśli).

W praktyce Programowanie Ekstremalne jest podobnym stylem prowadzenia projektu do tego proponowanego w Scrum. Zakładana jest ścisła współpraca z klientem, podział projektu na mniejsze, niezależne funkcjonalności, duży nacisk kładzie się na testy. Podstawową różnicą jest fakt, że Programowanie Ekstremalne skupia się przede wszystkim na aspektach technicznych i wymaga wysokich kompetencji od uczestników projektu, oraz ich pełnego zaangażowania od początku prac⁴⁵. Za przewagę Scruma można uznać lepiej zorganizowany przepływ pracy (Workflow) i łatwiejszy próg wejścia dla gorzej zorganizowanego zespołu.

2.3 Inne metody wykorzystywane w IT

Zarządzanie projektami nie narzuca jednej określonej metodyki w obrębie pojedynczego projektu. Zdarza się, że w trakcie prac projektowych zostaje podjęta decyzja o zmianie metodologii prowadzenia projektu. Pomimo że wiąże się to zazwyczaj ze sporymi kosztami nie zawsze jest to decyzja negatywna, ponieważ stwarza możliwość

⁴³ Tak jak ma to miejsce w opracowaniu Wysockiego.

⁴⁴ Extreme Programming, Rules of XP, <http://www.extremeprogramming.org/rules.html> (dostęp: 25.08.2020)

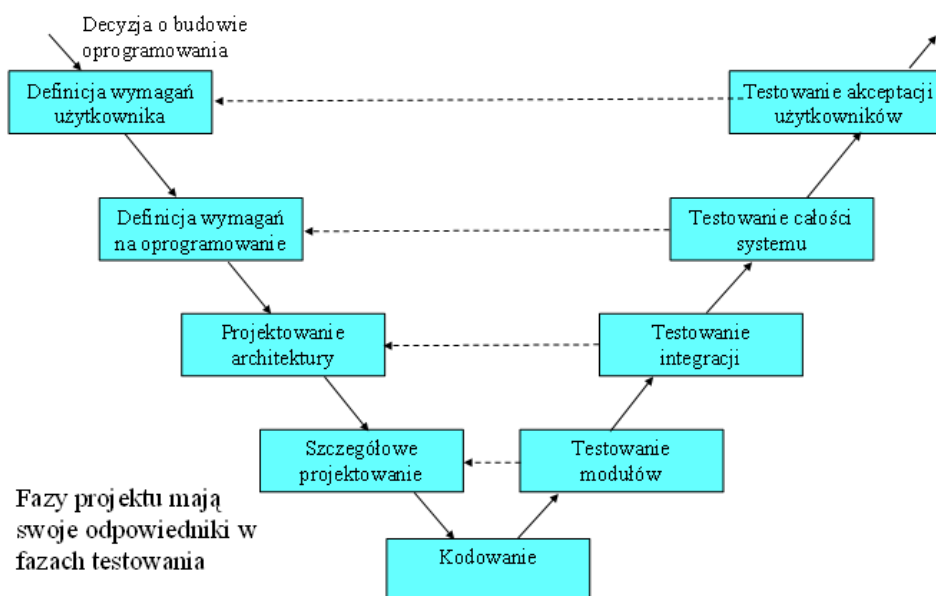
⁴⁵ Chromatic.: Extreme Programming Pocket Guide: Team-Based Software Development, O'Reilly Media 2013, s. 7-10

dostarczenia produktu, nad którym prace mogłyby zostać porzucone, jeśli do tej pory efekty nie były zadowalające. Zdarza się też, że w ramach firmy używa się więcej niż jednej metody np. metod zwinnych typu Scrum do zarządzania procesem powstawania produktu, a Waterfall do wprowadzania zmian w strukturze organizacyjnej.

W ramach jednego projektu połączenie metody Waterfall ze Scrum'em wydaje się nierealne ze względu na przeciwstawne założenia dotyczące logiki obu metodologii (iteracja vs kaskada etapów). Popularnym trendem jest wspieranie głównej metodyki rozwiązaniami zaczerpniętymi z mniej popularnych strategii zarządzania projektami. Najczęściej wykorzystuje się Model V (konceptcja rozbudowanych, wieloetapowych testów) i Kanban (tablica zadań, kaizen).

2.3.1 Model V

Jest to sekwencyjny model wytwarzania oprogramowania, który domyślnie wyróżnia cztery poziomy cztery poziomy rozwoju oprogramowania, każdy z odpowiadającym mu poziomem testów weryfikacyjnych i walidacyjnych. W praktyce ilość etapów może ulec zmianie i zależy od założeń projektowych⁴⁶.



Rysunek 9: Struktura Modelu V⁴⁷

⁴⁶ ISTQB, Certyfikowany tester – Plan poziomu podstawowego, 2011, s. 24

⁴⁷ Źródło: https://edux.pjwstk.edu.pl/mat/200/lec/wyklady/11_3.html

Model V może być wprowadzony do projektu zarządzanego metodyką zwinną w celu zapewnienia lepszej jakości tworzonego systemu. Wczesne włączenie testowania do etapów projektu umożliwi lepsze przygotowanie przypadków testowych co będzie skutkowało lepszym pokryciem wymagań projektowych. Ponadto spodziewane jest znaczne przyspieszenie pozytywnego dostarczenia produktu⁴⁸. Dzięki zastosowaniu Modelu V proces integracji modułów i poszczególnych funkcjonalności systemu IT przebiega sprawniej, a ten etap stwarza duże ryzyko wystąpienia niezgodności, których wykrycie jest czasochłonne i generuje koszty.

2.3.2 Kanban

Kanban obecnie w zarządzaniu projektami IT jest traktowany jako osobna metodyka zwinna. Stworzona w pierwszym dziesięcioleciu XX, na wzór japońskiego Kanbana z lat 50, wykorzystywanego w zarządzaniu produkcją. Tak jak w oryginale, także tutaj głównymi wartościami są eliminacja marnotrawstwa i zapewnienie płynności realizacji zadań. Metoda może być alternatywą dla Scrum. Podstawowe zasady Kanban to⁴⁹:

- wizualizacja postępów prac na tablicy,
- ustalenie limitów WIP (Work In Progress), czyli ile maksymalnie zadań w toku dany dział może mieć przed rozpoczęciem kolejnych,
- Kaizen – ciągłe doskonalenie poprzez wprowadzanie małych, częstych zmian,
- propagowanie efektywnej komunikacji.

Podobnie jak w przypadku Programowania Ekstremalnego metodyka Kanban wymaga wysokich kompetencji pracowników oraz dobrej organizacji pracy⁵⁰. Charakterystyczną rzeczą dla niej jest tablica Kanban (Kanban Board), czyli zapewnienie transparentnego miejsca do zapisywania zadań i zarządzania nimi. Taka tablica zawiera przynajmniej trzy kolumny:

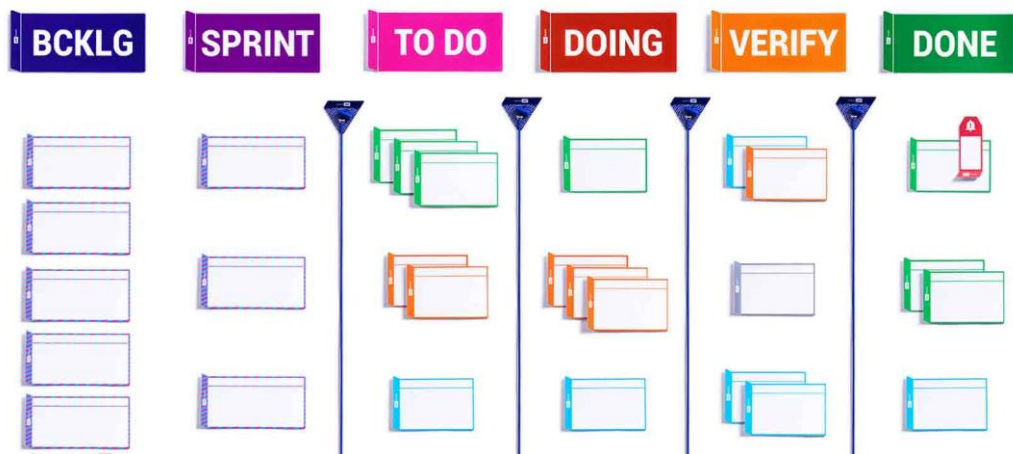
⁴⁸ Pawlak R.: Testowanie Oprogramowania. Podręcznik dla Początkujących, Helion, Gliwice 2014, s. 63-64

⁴⁹ Kanbanize, Introduction to Kanban, <https://kanbanize.com/kanban-resources/getting-started> (dostęp: 26.08.2020)

⁵⁰ Stabryła A., Wawak S.: Metody badania i modele rozwoju organizacji, UEK, Kraków 2012, s. 277-279

- Do Zrobienia (To Do)
- W trakcie (Doing)
- Zrobione (Done)

Zadania są przedstawiane często w formie samoprzylepnych kartek, po jednej na zadanie. Każdy pracownik jest zobligowany do częstych aktualizacji stanu swoich zadań poprzez przeklejanie kartek do odpowiedniej kolumny. Takie tablice są często zapożyczane przez menedżerów projektu pracujących w metodzie Scrum, ponieważ oferują jasny wgląd w aktualny stan prac podczas Sprintu oraz wspomagają przepływ informacji. Jira, oprogramowanie wspomagające zarządzanie projektami w środowisku zwinnym, o którym już była mowa oferuje taką tablicę. Ilość kolumn może zostać zwiększona na potrzeby projektu.



Rysunek 10: Przykład tablicy Kanban, stworzonej w stylu metodyki Scrum⁵¹

Próba połączenia Kanbana ze Scrumem jest Scrumban – podejście, które ma łączyć najlepsze cechy obu metodyk, czyli usprawnić zadaniową naturę Scrum z jasnym podziałem ról o metody usprawniania procesu Kanban⁵².

⁵¹ Źródło: <https://www.patboard.com/shop/scrum-kanban-board-magnets-patboard-scrum-board-tool-set/>

⁵² Kanbantool, Wprowadzenie do Scrumban, <https://kanbantool.com/pl/scrumban-kanban-i-scrum> (dostęp 27.08.2020)

3 Scrum – ogólny zarys i struktura

Metoda Scrum wywodzi się z empirycznego podejścia do rozwiązywania problemów, skuteczność metody w dużej mierze zależy od wiedzy i doświadczenia uczestników biorących udział w procesie. Scrum propaguje podejście inkrementalne, czyli podział wytycznych na podzadania, co zapewnia przewidywalność oraz kontrolę ryzyka. Ponadto metoda ma charakter iteracyjny – wyróżnia jasno zdefiniowane etapy pracy, które tworzą pętlę. Scrum jest w pełni zgodny z manifestem Agile, więc jest zaliczany do zwinnych metod zarządzania projektami. W celu ułatwienia sprawnego zaimplementowania metodyki, twórcy stworzyli jasno określoną strukturę oraz zasady, których przestrzeganie ma gwarantować pomyślną pracę nad projektem.

Trzema filarami, na których opiera się Scrum są⁵³:

- Przejrzystość,
- Inspekcja,
- Adaptacja.

Przejrzystość – mówi, że ważne aspekty projektu i czynności z nimi związane powinny być jasno zdefiniowane i znane dla wszystkich pracowników zespołu projektowego. W szczególności dotyczy się to definicji słowa „Ukończone”. Zarówno pracujący (np. realizatorzy projektu) jak i obserwujący (np. klienci) powinni być zgodni co do końcowego efektu, który planuje się osiągnąć. W praktyce polega to na wprowadzeniu ogólnie przyjętej Definicji Ukończenia (DOD – Definition of Done).

Inspekcja – od użytkowników Scruma wymaga się częstych kontroli osiąganych efektów z ustalonymi wymaganiami w celu ograniczenia a najlepiej całkowitego uniknięcia kumulacji niezgodności i błędów. Dotyczy to w szczególności Artefaktów (więcej w podrozdziale 5.3). Najlepszym rozwiązaniem jest powierzenie inspekcji osobom do tego wykwalifikowanym lub już mającym doświadczenie w celu przyspieszenia procesu i nie marnowaniu przez pracowników czasu, wymaganego na pracę nad samym produktem.

⁵³ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 4-5

Adaptacja – jeśli okaże się, że występują znaczące różnice pomiędzy tworzoną produktem a założonymi wymaganiami, w wyniku czego grozi ryzyko nieakceptacji produktu przez klienta, należy jak najszybciej wprowadzić poprawy lub po konsultacji z klientem dokonać zmian w wymaganiach.

Poza filarami struktura Scrum składa się z:

- Ról – obowiązków poszczególnych członków zespołu,
- Zdarzeń – etapów pracy,
- Artefaktów – sposobu raportowania efektów pracy. Zostaną one dokładnie omówione w kolejnych podrozdziałach.

3.1 Role

Typowy Zespół Scrum składa się z jednego Właściciela Produktu (Product Owner), jednego Scrum Master'a⁵⁴ i Zespołu Deweloperskiego (Development Team). Każdy zespół jest samoorganizującą się i wielofunkcyjną jednostką. Sam organizuje swoją pracę i decyduje w jaki sposób ją wykona. Taki model zespołu ma na celu pobudzić kreatywność i produktywność członków zespołu⁵⁵.

Efekty pracy zespołu są dostarczane stopniowo i iteracyjnie. Duży nacisk kładzie się na komunikację pomiędzy poszczególnymi rolami i klientem⁵⁶.

⁵⁴ Ze względu na powszechnie używany język angielski w branży IT i brak zdefiniowanych oficjalnych polskich tłumaczeń dla niektórych terminów (przykładowo: „Mistrz Młyna” jako dosłowne tłumaczenie Scrum Master praktycznie nie jest używane) niektóre nazwy w niniejszej pracy będą zgodne z panującym nazewnictwem w branży.

⁵⁵ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 6-7

⁵⁶ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 7



Rysunek 11: Typowa struktura Zespołu Scrum⁵⁷

3.1.1 Właściciel Produktu (Product Owner)

Jest to osoba bezpośrednio odpowiedzialna za produkt – jego design, wykonanie i implementację. Głównym zadaniem Właściciela Produktu jest nadzorowanie prac Zespołu Deweloperskiego w celu dopilnowania osiągnięcia wymaganej wartości i jakości produktu przy zachowaniu wydajnej pracy zespołu.

Właściciel Produktu jest odpowiedzialny za zarządzanie Rejestrem Produktu (Product Backlog), co sprowadza się do⁵⁸:

- pilnowania, żeby elementy Rejestru były dostępne, przejrzyste i zrozumiałe dla wszystkich, a zwłaszcza dla Zespołu Deweloperskiego,
- porządkowania kolejności elementów, w sposób zgodny z założeniami,
- optymalizowania wartości pracy wykonanej przez Zespół Deweloperski,
- dopilnowanie, żeby założenia Sprintu zostały spełnione.

⁵⁷ Źródło: <https://www.mendix.com/blog/the-road-to-adopting-scrum-team-composition/>

⁵⁸ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 6-7

Obowiązków Właściciela Produktu nie należy rozumieć jako pracy indywidualnej. Oczekuje się od niego ścisłej współpracy z resztą Zespołu Scrum. Podczas gdy Właściciel produktu pilnuje, żeby wszystkie niezbędne prace zostały wykonane, Zespół Deweloperski i Scrum Master wspierają go poprzez ewentualne poprawki w Rejestrze⁵⁹.

3.1.2 Scrum Master

Zadaniem Scrum Master'a jest promowanie zasad Scrum i zapewnianie wsparcia każdej osobie związanej z projektem w zrozumieniu teorii, praktyk, reguł i wartości metodologii⁶⁰. Przewodnik Scrum określa Scrum Master'a jako przywódcę służebnego (servant leader). Oznacza to, że powinien służyć reszcie zespołu radą i sugestią co do kierunku działań w zgodzie z metodami zwinnymi, ale jednocześnie być gotowym pomóc w razie pojawienia się wątpliwości dotyczących mechaniki metody Scrum. Powinien nie tylko dopilnować, że wszystkie spotkania scrumowe się odbywają zgodnie z harmonogramem, ale także, że są one efektywne. Pomimo tego, że jest odpowiedzialny za realizację książkowego Scrum'a, w jego kompetencjach nie leży organizacja pracy – to już zadanie zespołu.

Pracę Scrum Master'a, ze względu na dotyczący podmiot można podzielić na 3 części⁶¹:

- wspieranie Właściciela Produktu:
 - zapewniając, że cele, zakres i domena produktowa są zrozumiałe dla wszystkich w Zespole Scrum,
 - w efektywnym zarządzaniu Rejestrem Produktu,
 - zapewniając, że Właściciel Produktu wie, jak porządkować Rejestr Produktu, aby maksymalizować wartość,
 - w rozumieniu i praktykowaniu zwinności (zasad Agile),
- wspieranie Zespołu Deweloperskiego:
 - w zakresie wykorzystania zasad samoorganizacji zgodnie z zasadami zwinnymi,

⁵⁹ Pichler R.: Zarządzanie projektami ze Scrumem, Helion, Gliwice 2014, s. 30

⁶⁰ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 7

⁶¹ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 8

- wspomagając przebieg wydarzeń scrumowych, kiedy jest to konieczne lub kiedy jest o to proszony,
- wspieranie organizacji:
 - przewodząc procesom wdrażania Scruma oraz prowadząc szkolenia osób w ten proces zaangażowanych,
 - planując wykorzystanie Scruma wewnątrz organizacji,
 - wspierając pracowników i interesariuszy⁶² w zrozumieniu i stosowaniu Scruma,
 - współpracując z innymi Scrum Masterami w celu zwiększenia efektywności wykorzystania Scruma w organizacji.

3.1.3 Zespół Deweloperski

Zgodnie z Przewodnikiem Scrum, Zespół Deweloperski powinni tworzyć profesjonaliści, którzy wykonują wymaganą pracę, żeby dostarczyć ukończony (Done) Przyrost⁶³ (Increment) produktu na koniec każdego Sprintu. Efekty pracy są podsumowywane podczas Przeglądu Sprintu (Sprint Review). Nikt inny poza zespołem deweloperskim nie jest odpowiedzialny za tworzenie i dostarczenie Przyrostu⁶⁴.

Zespoły deweloperskie są to samodzielne jednostki, ustanowione z góry przez organizację. Oczekuje się od nich samodzielnego organizowania swojej pracy, nikt nawet Scrum Master nie może decydować za nich w jaki sposób przekształcać Rejestr Produktu w Przyrost gotowej do wydania funkcjonalności⁶⁵.

Zespoły Deweloperskie posiadają wiedzę i umiejętności niezbędne do wytworzenia Przyrostu produktu.

Scrum nie wnika wewnątrz Zespołu Deweloperskiego, stanowi on jedną komórkę, która sama organizuje swoją wewnętrzną strukturę (np. wyznaczenie Lidera Zespołu oraz podział zadań pomiędzy członków)

⁶² Poprzez interesariuszy rozumie się osoby bezpośrednio zainteresowane projektem i rezultatami prac.

⁶³ Spotykana polska nazwa to także *inkrement*

⁶⁴ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 7

⁶⁵ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 7

Odpowiedzialność za wykonywaną pracę ponosi cały Zespół Deweloperski⁶⁶.

Jeśli chodzi o optymalną wielkość zespołu deweloperskiego Przewodnik radzi, żeby był on wystarczająco liczny, żeby mógł skutecznie wykonywać pracę w ramach Sprintu ale jednocześnie na tyle mały, by mógł zachować zasady zwinne.

Zespół trzyosobowy stwarza ryzyko braku wystarczających kompetencji uniemożliwiające dostarczenia Przyrostu produktu na czas. Z kolei zbyt duży zespół liczący dziesięciu członków wymaga poświęcenia znacznie większych nakładów na koordynację i może spowodować opóźnienia w pracy produkcyjnej.

3.2 Zdarzenia (Scrum Events)

Opisane niżej Zdarzenia zostały stworzone w celu wprowadzenia systematyczności w spotkaniach projektowych i zminimalizować do minimum potrzebę innych spotkań bezpośrednio niezdefiniowanych przez Scrum. Czas trwania każdego Zdarzenia powinna być ustalona na początku prac projektowych, a ramy czasowe powinny być respektowane przez wszystkich uczestników Scruma. Poza Sprintem, który jest zawiera w sobie pozostałe Zdarzenia, każde zdarzenie powinno być wykorzystywane jako szansa na przeprowadzenie kontroli nad pracami projektowymi i podjęcia odpowiednich działań naprawczych jeśli zostaną znalezione problemy lub ryzyko niespełnienia oczekiwanych przez klienta wymagań.

⁶⁶ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 7

wynikać z nacisku interesariuszy, Zespołu Deweloperów czy nawet Scrum Master'a. Zakłada się, że głównym powodem anulowania całego Sprintu jest wykrycie, że cel sprintu jest nieaktualny⁶⁸. Może to wynikać ze zmian w budżecie projektu, warunkach rynkowych lub najgorszym razie rezygnacji klienta z projektu. Decyzja o anulowaniu Sprintu jest podejmowana bardzo rzadko z powodu dużego pochłonięcia zasobów oraz dezorganizacji pracy.

Po zakończeniu Sprintu następuje przegląd ukończonych elementów Backlogu produktu, które następnie są akceptowane przez Właściciela Produktu. Elementy nie spełniające założeń są ponownie szacowane i umieszczane z powrotem w rejestrze produktu⁶⁹.

3.2.2 Planowanie Sprintu (Sprint Planning)

Jest to etap planowania i organizacji pracy do wykonania podczas Sprintu. Uczestniczy w nim cały Scrumowy zespół. Domyślnie dla miesięcznego Sprintu przewiduje się ośmiogodzinne spotkanie. Osobą odpowiedzialną za organizację spotkania i dopilnowanie, że odbywa się zgodnie z zasadami Scrum jest Scrum Master. Przewodnik wyszczególnia dwa pytania jakie powinny być zadane podczas wydarzenia⁷⁰:

- Co może zostać dostarczone w ramach Przyrostu w nadchodzącym Sprincie?
- W jaki sposób to osiągniemy i ile wymaga to pracy?

Zespół Deweloperski pracuje nad ustaleniem funkcjonalności, która zostanie opracowana podczas Sprintu. Właściciel Produktu omawia cel, który powinien osiągnąć Sprint oraz elementy Rejestru Produktu (Product Backlog), które, jeśli zostaną ukończone w Sprincie, spełnią Cel Sprintu. Cały Zespół Scrum współpracuje nad zorganizowaniem pracy Sprintu. Danymi wejściowymi tego spotkania są Rejestr Produktu, najnowszy Przyrost osiągnięty w poprzednim Sprincie oraz wnioski i opinie Zespołu Deweloperskiego wyciągnięte z dotychczasowej pracy. To właśnie deweloperzy wybierają, które elementy Rejestru stworzą podczas nadchodzącego Sprintu mając na uwadze swoją dotychczasową efektywność i umiejętności.

⁶⁸ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 9-10

⁶⁹ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 10

⁷⁰ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 10-11

Najważniejszym punktem Planowania Sprintu jest wspólne określenie przez wszystkich osoby biorące udział w projekcie Celu Sprintu (Sprint Goal)⁷¹ – wytycznych zgodnych z Rejestrem, które powinny zostać spełnione podczas Sprintu.

Co do drugiego punktu, po ustaleniu Celu i wybraniu elementów Rejestru do zaimplementowania podczas Sprintu Zespół Deweloperów przystępuje do analizy wymagań i dyskusji w jaki sposób osiągnie zamierzony Przyrost. Elementy Rejestru Produktu wybrane do implementacji podczas Sprintu trafiają do Rejestru Sprintu⁷².

Właściciel Produktu jako osoba sprawująca pieczę nad wymaganiami może być poproszona o pomoc przy wybieraniu elementów Rejestru do implementacji lub wyjaśniania wszelkich wątpliwości z nimi związanymi. Do niego również powinny być zgłaszane wszelkie sugestie dotyczące nakładów pracy zwłaszcza jeśli dany pojedynczy element Rejestru jest zbyt skomplikowany żeby go ukończyć podczas jednego cyklu Sprintu.

Od poprawnie przeprowadzonej fazy Planowania zależy z jaką efektywnością Zespół będzie pracował nad implementacją wybranych wymagań. W przypadku niepoświęceniu na analizę Sprintu wystarczająco dużo czasu istnieje ryzyko, że ten okres zostanie niedostatecznie wykorzystany (wybrano zbyt mało elementów Rejestru, lub tylko te najprostsze do realizacji) lub przeszacowano możliwości zespołu (wybrano zbyt dużo elementów zwłaszcza tych najbardziej wymagających). Zespół Deweloperski powinien pilnować żeby praca podczas każdego Sprintu była jak najbardziej optymalnie rozłożona w ustalonych ramach czasowych.

3.2.3 Dzienny Scrum (Daily scrum)

Dzienny Scrum jest około piętnastominutowym spotkaniem Zespołu Deweloperskiego, które odbywa się przynajmniej raz każdego dnia podczas Sprintu. Podstawowym celem spotkania jest podsumowanie dotychczasowych postępów oraz zaplanowanie pracy

⁷¹ Założenia zaplanowane do osiągnięcia w ramach pojedynczego Sprintu.

⁷² Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 10-11

członków zespołu na kolejny dzień. Ponadto dokonuje się przeglądu problemów, jeśli jakieś wystąpiły oraz sprawdza się czy praca odbywa się spójnie z Rejestrem Sprintu.

Zespół deweloperów jest zobligowany organizować codzienne spotkania w swoim zakresie. Rolą Scrum Master'a jest nadzorowanie czy spełnione są zasady co do czasu trwania i formy spotkania.

Rekomendowana struktura podana w przewodniku Scrum zakłada znalezienie odpowiedzi na trzy pytania⁷³:

- Co zrobiłem wczoraj, co pomogło Zespołowi Deweloperskiemu przybliżyć się do osiągnięcia Celu Sprintu?
- Co zrobię dzisiaj, co pomoże Zespołowi Deweloperskiemu przybliżyć się do osiągnięcia Celu Sprintu?
- Czy widzę jakiegokolwiek przeszkody mogące uniemożliwić mi lub Zespołowi Deweloperskiemu osiągnięcie Celu Sprintu?

Każdy członek Zespołu powinien po kolei odpowiedzieć na te pytania a jakiegokolwiek problemy powinny zostać rozwiązane we wspólnym gronie.

Nie należy mieszać Dziennego Scrum'a ze spotkaniem raportującym progres Właścicielowi Produktu. To spotkanie powinno wносить wartość merytoryczną do pracy Zespołu a nie być zwykłym przedstawieniem postępów⁷⁴.

3.2.4 Przegląd Sprintu (Sprint Review)

Jest to spotkanie, które powinno być zorganizowane na końcu każdego Sprintu. Dokonuje się na nim przeglądu Przyrostu w celu aktualizacji Rejestru Produktu. Poprzez przegląd Przyrostu rozumie się podsumowanie i analizę wszystkich rzeczy dokonanych podczas ostatniego Sprintu. To wydarzenie również ma swoje ramy czasowe. Zakłada się poświęcenie czterech godzin w przypadku miesięcznego Sprintu.

⁷³ Schwaber K., Sutherland J.: The Scrum Guide 2017, s. 12

⁷⁴ Agile247, Daily Scrum, <https://agile247.pl/daily-scrum-standup> (dostęp: 27.08.20)

Tym razem najważniejszymi uczestnikami spotkania są interesariusze, którzy biorą udział we wspólnej dyskusji z deweloperami na temat dokonań podczas tego Sprintu. Bazując na dyskusji następnie, wyciąga się wnioski co do tempa pracy a także prognozuje dalsze postępy.

Przegląd sprintu składa się z następujących etapów⁷⁵:

- zebranie Zespołu Scrum i kluczowych interesariuszy zaproszonych przez Właściciela Produktu,
- zapoznanie się z punktami Rejestrem Produktu, zarówno tymi „Ukończonymi” i tymi nie,
- Zespół Deweloperski przedstawia podsumowanie prac w trakcie Sprintu, informuje o sukcesach i napotkanych problemach,
- Zespół Deweloperski prezentuje „Ukończoną” pracę oraz odpowiada na pytania odnośnie do Przyrostu,
- Właściciel Produktu omawia aktualny Rejestr Produktu, jeśli to konieczne aktualizuje przewidywany termin zakończenia prac, mając na uwadze bieżące postępy,
- wspólna dyskusja w celu przygotowania wartościowych materiałów na następujące Planowanie kolejnego Sprintu,
- analiza potencjalnych zmian spowodowanych wpływem rynku i innych czynników zewnętrznych na produkt,
- rewizja zasobów pieniężnych i czasu dla kolejnego przewidywanego wydania produktu.

W wyniku Przeglądu powstaje odświeżony Rejestr Produktu, który zawiera elementy do zaimplementowania podczas kolejnego Sprintu⁷⁶. Jeśli stwierdzono, że wpływ czynników zewnętrznych np. rynkowych się zmienił, uwzględnia się to poprzez odpowiednie zmiany w Rejestrze.

⁷⁵ Schwaber K., Sutherland J.: The Scrum Guide 2017, s 11-12

⁷⁶ Schwaber K., Sutherland J.: The Scrum Guide 2017, s 13

Należy pamiętać, że każdy uczestnik spotkania może zabrać głos a decyzje podejmowane są wspólnie. Wszyscy powinni kierować się chęcią osiągnięcia maksymalnej wartości tworzonego produktu.

Podczas spotkań w ramach Przeglądów, należy pamiętać o możliwości wprowadzenia zmian do projektu, jeśli zajdzie taka potrzeba jak np. w przypadku wpływu działań konkurentów czy zaistniałych nagłych zmian na rynku.

3.2.5 Retrospektywa Sprintu (Sprint Retrospective)

Wydarzeniem, które odbywa się bezpośrednio po Przeglądzie jest Retrospektywa Sprintu. Jest to spotkanie całego Zespołu Scrum. Spotkanie służyć ma analizie pracy podczas ostatniego Sprintu, osiągniętych rezultatów i wydajności i stworzeniu możliwości na wprowadzenie ulepszeń przed nadchodzącym Sprintem. Przewodnik Scrum zaleca trzygodzinne spotkania w przypadku miesięcznego Sprintu. Jak przy każdym wydarzeniu w metodologii Scrum za poprawne przeprowadzenie spotkania wraz z zachowaniem wszystkich zasad odpowiada Scrum Master.

Wydarzenie składa się z trzech podpunktów⁷⁷:

- inspekcja poprzedniego Sprintu pod względem zaangażowania ludzi, relacji procesów i używanych narzędzi,
- identyfikacja i uporządkowanie aspektów mających pozytywny wpływ na dotychczasową pracę,
- stworzyć plan wprowadzenia ulepszeń do dotychczasowej pracy, jeśli takie znaleziono.

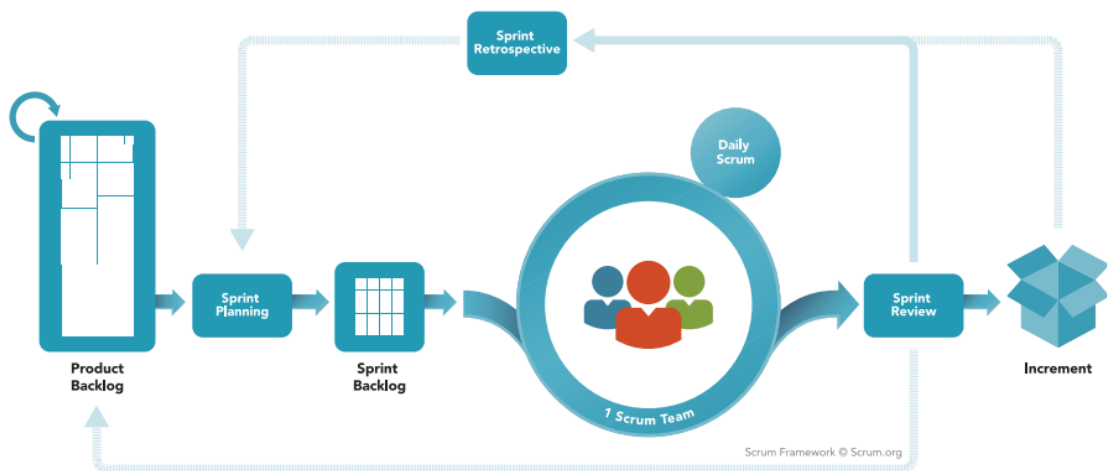
Choć Scrum zezwala na wprowadzania zmian kiedykolwiek, Retrospektywa stwarza oficjalne okno czasowe na zastanowienie się nad ulepszeniami.

⁷⁷ Schwaber K., Sutherland J.: The Scrum Guide 2017, s 14

3.3 Artefakty

Artefakty Scrum w przejrzysty sposób reprezentują pracę lub wartość. Umożliwiają przeprowadzenie inspekcji i adaptacji warunków pracy. Wyróżnia się trzy Artefakty, są to⁷⁸:

- Rejestr Produktu
- Rejestr Sprintu
- Przyrost



Rysunek 13: Artefakty Scrum na schemacie⁷⁹

3.3.1 Rejestr Produktu (Product Backlog)

Rejestr Produktu to nic innego jak uporządkowana lista wszystkich wiadomych wymagań odnośnie do rozwoju produktu. Osobą odpowiedzialną za zawartość, poszeregowanie i dostępność Rejestru są Właściciel Produktu⁸⁰.

Rejestr podlega zmianom i aktualizacjom przez cały okres prac projektowych. Dynamiczne zmiany mają na celu zapewnienie, że produkt usatysfakcjonuje klienta zarówno pod względem funkcjonalności jak i jakości.

⁷⁸ Schwaber K., Sutherland J.: *The Scrum Guide* 2017, s 14-17

⁷⁹ Źródło: <https://www.scrum.org/resources/scrum-framework-poster>

⁸⁰ Schwaber K., Sutherland J.: *The Scrum Guide 2017*, s 15

Elementy Rejestru składają się z następujących cech: opisu, kolejności, oszacowania i wartości⁸¹.

Z każdym wydaniem kolejnej wersji produktu w wyniku analiz i wspólnej dyskusji obraz wymagań produktu staje się coraz bardziej klarowny, z tego powodu Właściciel Produktu powinien zadbać o jego częste aktualizacje. Może się zdarzyć taka sytuacja, że nad tym samym produktem pracuje więcej niż jeden Zespół Scrum. Tym bardziej należy wtedy zachować szczególną dbałość o utrzymywanie Rejestru w jak najświeższym stanie wraz z porządkiem, który zespół odpowiada za które wymaganie.

Przewodnik Scrum wyróżnia takie działanie jak Doskonalenie (Refinement) Rejestru Produktu. Jest to ciągły proces, w którym Właściciel produktu wspólnie z Zespołem Deweloperskim opracowują szczegółów elementów Rejestru⁸². Zespół powinien wydzierżawić około 10% swojego czasu pracy na Doskonalenie, Za zgodą Właściciela Produktu elementy Rejestru mogą być zaktualizowane w każdej chwili.

jest działaniem polegającym na dodawaniu szczegółów, oszacowań i porządkowaniu elementów Rejestru Produktu. Jest to ciągły proces, podczas którego Właściciel Produktu wraz z Zespołem Deweloperskim opracowują szczegóły elementów Rejestru Produktu. Podczas doskonalenia Rejestru Produktu jego elementy są przeglądane i korygowane. To, jak i kiedy prowadzone jest doskonalenie, zależy od Zespołu Scrumowego. Doskonalenie zazwyczaj zajmuje nie więcej niż 10% czasu Zespołu Deweloperskiego w Sprincie. Nie zmienia to faktu, że elementy Rejestru Produktu mogą być uaktualniane w każdej chwili przez Właściciela Produktu lub za jego zgodą.

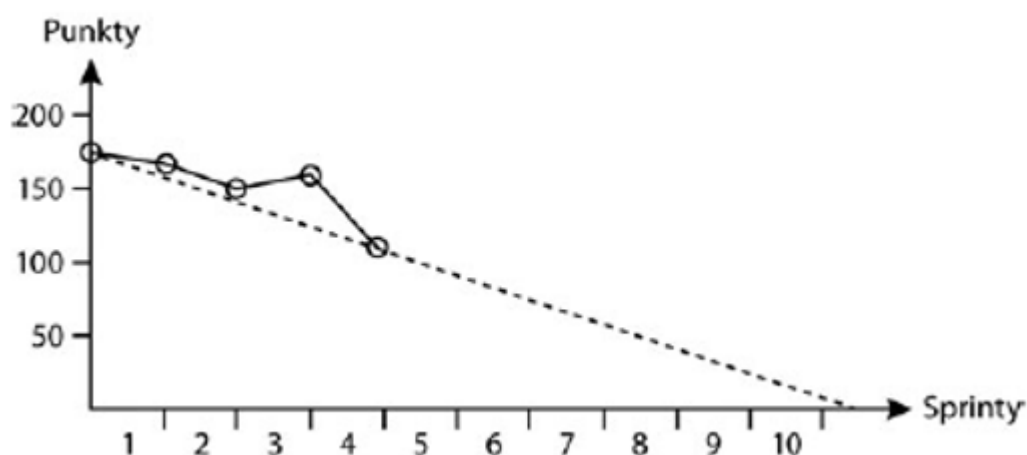
Organizacja uporządkowania Rejestru ma na celu wyszczególnienie tych elementów, które są ważniejsze i już w pełni zdefiniowane. Im niżej dany element jest na liście tym posiada mniej szczegółów. Na kolejny Sprint przewiduje się realizację najdokładniejszych wymagań, ułożonych na samej górze Rejestru, które mogą zostać „Ukończone” podczas jednego Sprintu.

⁸¹ Schwaber K., Sutherland J.: The Scrum Guide 2017, s 15-16

⁸² Schwaber K., Sutherland J.: The Scrum Guide 2017, s 15

Zespół Deweloperski jest odpowiedzialny za wszystkie oszacowania. Właściciel Produktu może wpływać na Zespół Deweloperski pomagając dostrzegać kompromisy i dokonywać odpowiednich wyborów, ale ostatecznie pracę szacują osoby, które będą ją wykonywać⁸³.

Właściciel Produktu przynajmniej raz podczas każdego Przeglądu Sprintu powinien przygotować podsumowanie pozostającej do wykonania pracy w celu porównania go z szacunkowymi przybliżeniami dokonanyymi przed rozpoczęciem Sprintu bazując na dotychczasowych wynikach. Dzięki takiemu porównaniu Właściciel Produktu może jasno określić, jak kształtuje się postęp pracy, jaka jest szansa na ukończenie zadań na czas, zgodnie z założonym celem. Następnie ocena postępów jest prezentowana wszystkim interesariuszom. Przy prognozowaniu postępów prac używa się różnych sposobów monitorowania trendu takich jak: spalanie/wypalanie (burn-down/burn-up) albo wykresów kumulatywnych⁸⁴.



Rysunek 14: Przykładowy wykres spalania wydania⁸⁵

Wykres składa się z dwóch linii. Linia ciągła to spalanie faktyczne, jest to odzwierciedlenie aktualnego postępu oraz pozostałej ilości pracy. Na powyższym, przykładowym wykresie można zauważyć, że początkowo tempo prac projektowych było

⁸³ Schwaber K., Sutherland J.: The Scrum Guide 2017, s 15

⁸⁴ Schwaber K., Sutherland J.: The Scrum Guide 2017, s 15-16

⁸⁵ Pichler R.: Zarządzanie projektami ze Scrumem, Helion, Gliwice 2014, s. 121

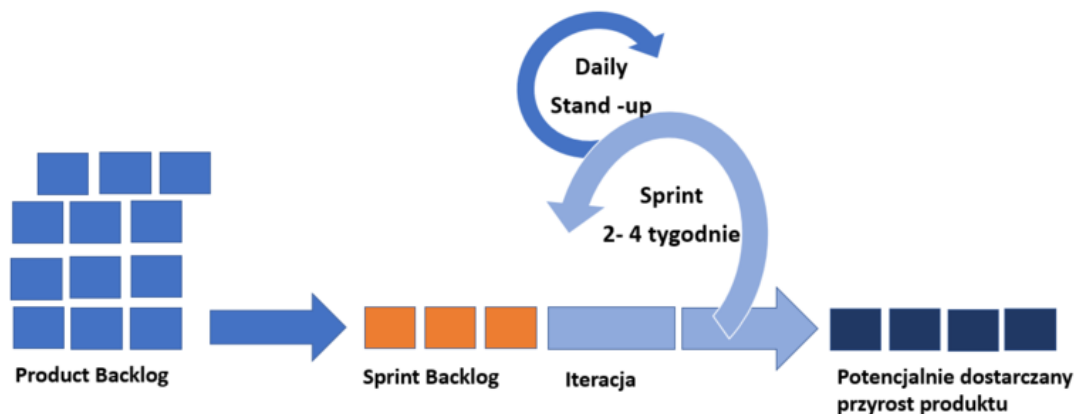
wolne. Mogłoby to być spowodowane problemami komunikacyjnymi bądź technicznymi. W trzecim Sprincie ilość pracy była o wiele większa niż dotychczas. Takie efekt mogłoby spowodować ponowne oszacowanie elementów rejestru przez zespół, lub ewentualnie znalezienie nowych wymagań, niezbędnych do spełnienia z punktu widzenia produktu. W czwartym sprincie spalanie wyraźnie przyspieszyło. Linia przerywana obrazuje trend spalania – prognozę postępów w następnych sprintach. Wynika z niej, że przy stałym tempie pracy ukończenie projektu zajmie 10 Sprintów. Dysponując tą wiedzą, menedżer projektu może podjąć odpowiednie decyzje w celu utworzenia harmonogramu i ustalenia tempa prac⁸⁶.

Z reguły największą wagę przywiązuje się do danych empirycznych z poprzednich projektów, wychodząc z założenia, że w złożonych środowiskach tylko to co już się wydarzyło, może być wykorzystywane przy podejmowaniu decyzji dotyczących przyszłości.

3.3.2 Rejestr Sprintu (Sprint Backlog)

Rejestr Sprintu jest listą zadań, które Zespół Scrum ma do wykonania w nadchodzącym Sprincie. Innymi słowy jest to plan na wytworzenie Przyrostu i realizację Celu Sprintu. Zadania są przenoszone z Rejestru Produktu przez Zespół Deweloperski, zgodnie z priorytetami ustalonymi przez Właściciela Produktu.

⁸⁶ Pichler R.: Zarządzanie projektami ze Scrumem, Helion, Gliwice 2014, s. 120-123



Rysunek 15: Rejestr Sprintu na schemacie⁸⁷

Ważne jest, aby to Zespół Deweloperski dobierał zadania mając na uwadze swoje możliwości produkcyjne, ponieważ to oni będą te zadania realizować. Istnieje możliwość zmiany zawartości Rejestru Sprintu w przypadku problemów z implementowaniem założonych wymagań, dlatego deweloperzy powinni na bieżąco kontrolować swoją pracę, szacować możliwe wyniki i być w kontakcie z Właścicielem Produktu. Podczas wspólnych negocjacji co do ostatecznej zawartości Rejestru Sprintu na uwadze powinien być Cel Sprintu, który powinien pozostać niezmienny podczas całej iteracji Sprintu.

Aby Zespół Developerski mógł ciągle polepszać swoją pracę, w Rejestrze Sprintu powinno znaleźć się przynajmniej jedno, najważniejsze zadanie z Retrospektywy⁸⁸.

3.3.3 Przyrost Produktu (Product increment)

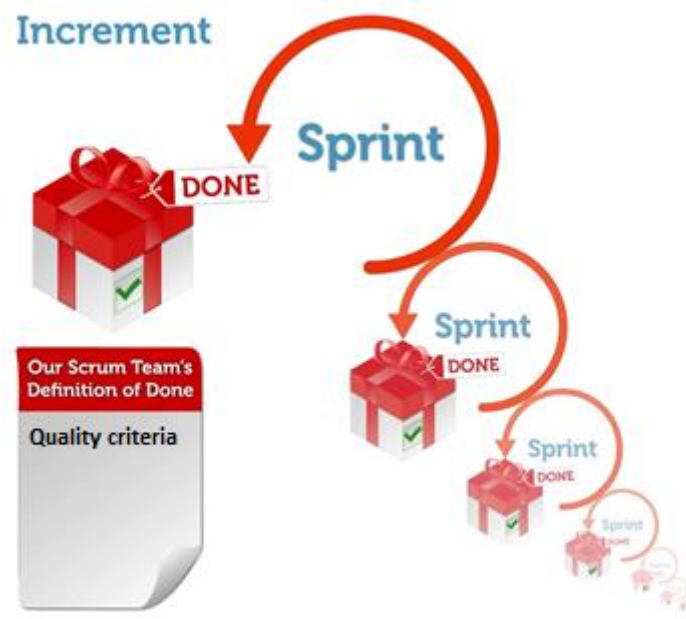
Poprzez Przyrost rozumie się wszystkie ukończone podczas Sprintu elementy Rejestru Produktu zsumowane z przyrostami osiągniętymi w poprzednich iteracjach. Każdy element powinien być zgodny z Definicją Ukończenia (Definition of Done), która jest ustalana na początku projektu przez Zespół Scrum⁸⁹.

⁸⁷ Źródło: https://mfiles.pl/pl/index.php/Backlog_produkту

⁸⁸ AgileHunters, Artefakty Scruma, <https://agilehunters.com/3-najwazniejsze-artefakty-scruma/> (dostęp: 28.09.2020)

⁸⁹ Schwaber K., Sutherland J.: The Scrum Guide 2017, s 16

Przyrost, powstały po Sprincie powinien być gotowym do potencjalnego wydania fragmentem produktu. Cały Zespół Scrum bierze udział w powstawaniu Przyrostu: deweloperzy odpowiada za budowanie i jakość, Właściciel Produktu dopilnowuje zgodności wymagań i ogólnej wizji produktu a Scrum Master pomaga wszystkim we wdrażaniu i poprawnej realizacji zasad zwinnych i reguł metodyki Scrum.



Rysunek 16: Przyrost Produktu w metodzie Scrum⁹⁰

3.4 Porównanie metod Waterfall i Scrum

W ramach podsumowania dotychczas zaprezentowanej teorii, poniżej zaprezentowano tabelę, która obrazuje najważniejsze charakterystyki dla metody tradycyjnej (na przykładzie Waterfall) i zwinnej (Scrum).

⁹⁰ Źródło: <https://youtu.be/0MTWysPmaoU?t=70>

Tabela 1: Porównanie podejścia tradycyjnego i zwinnego na przykładzie metodyk Waterfall i Scrum⁹¹

	Waterfall	Scrum
Kto zarządza projektem i podejmuje decyzje?	Kierownik projektu	Właściciel Projektu i Scrum Master w obrębie jednego Zespołu Scrum
Jak rozumiany jest sukces projektu?	Dostarczenie produktu	Maksymalne zadowolenie klienta
Jak wygląda kwestia wymagań klienta?	Sprecyzowane przed realizacją projektu	Stale aktualizowane (przed, w trakcie, a czasami po projekcie)
Jak wygląda zarządzania ryzykiem?	Realizowane wyłącznie z inicjatywy kierownika projektu	Realizowane przez cały zespół, codzienna identyfikacja ryzyka
Kontrola projektu a komunikacja?	metody ustalone przez kierownika projektu	Codziennie spotkania Scrumowe (daily Scrum)
Struktura zadaniowa projektu?	Struktura podziału pracy ⁹²	Dynamiczny Rejestr Produktu i Sprints
Współpraca z klientem podczas trwania projektu?	Niska	Wysoka
Kto jest odpowiedzialny za projekt?	Kierownik projektu	Zespół Scrumowy
Zakres projektu?	Stały	Zmienny
Cykl zarządzania projektem?	Sekwencyjno – kaskadowy	Iteracyjny, adaptacyjny, przyrostowy

⁹¹ Strojny K., Szmigiel M.: Analiza porównawcza podejść w zakresie zarządzania projektami, „Modern Management Review” Vol. XX, No. 22(3), 2015, s. 6-10

⁹² Jedną z technik zarządzania projektami, zorganizowanie zakresu prac przy pomocy hierarchicznej struktury drzewa

Jak można zauważyć metody zwinne charakteryzują się dużo większą elastycznością. Projekt nie jest zarządzany w prostoliniowy sposób, pojawia się możliwość wprowadzania zmian i częstych dyskusji z klientem na temat wymagań i założeń projektowych. Dzięki takim cechom, dobrze zaimplementowana metoda Scrum jest w stanie dostarczyć produkt najwyższej jakości, który spełni oczekiwania wymagającego klienta. Należy się liczyć, że wymaga to wysiłku zarówno po stronie całego Zespołu Scrumowego jak i samego klienta.

4 Zastosowanie metody Scrum na przykładach

W tym rozdziale zostaną przedstawione dwa przykłady praktycznego zastosowania Scrum w dwóch przedsiębiorstwach produkujących oprogramowanie. Pierwszy przykład będzie dotyczył dużej korporacji prowadzącej liczne projekty. Klientami firmy są człowie producenci samochodów a podstawowym produktem oprogramowanie samochodowe. Ten przykład pokaże, że zasady zwinne i metodyka Scrum mogą zostać skutecznie zaimplementowane nawet w dużym przedsiębiorstwie, gdzie pracuje wiele zespołów porzucanych po całym świecie. Ten przykład zostanie wzbogacony o doświadczenie autora z pracy w Zespole Scrum i wnioski dotyczące pracy w takim środowisku.

Drugim opisanym przykładem będzie analiza przypadku firmy Xebia, która przygotowywała system IT dla Dutch Railways. Celem tego przykładu jest zobrazowanie ryzyka związanego z wyborem złej metody zarządzania projektem. Firma była zmuszona zmienić założenia projektowe, a dzięki wdrożeniu Scruma w cykl tworzenia oprogramowania, była w stanie dostarczyć końcowy produkt.

4.1 Scrum w cyklu życia oprogramowania w firmie z branży samochodowej⁹³

Firma zatrudnia około czterdziestu tysięcy pracowników i posiada swoje placówki w wielu krajach. Na pewnym etapie swojego rozwoju zarząd firmy zdecydował się na wprowadzenie zasad zwinnych i metodyki Scrum – odpowiednio dostosowanej pod panujące warunki korporacyjne tj. wiele zespołów rozbitych po całym świecie. Jak się wydaje, jest to dobra decyzja, ponieważ obecnie, w wyniku rozwoju technologicznego do branży IT można zaliczyć wiele do tej pory niezwiązanych z nią dziedzin, w tym również branżę samochodową. Dzisiejsze samochody zawierają coraz więcej elektroniki i czujników, co znacząco kładzie nacisk na rozwój oprogramowania.

Model wprowadzony do firmy nazywa się AutoScrum i bazuje zarówno na zasadach Agile oraz Lean. Pozwala to dostosować metodykę zwinną do zorganizowania pracy, gdy liczba zespołów znacząco przekracza początkowe założenia Scrum’a oraz zapewnić, że są one zgodne z procesami produkcyjnymi w branży samochodowej obowiązującymi od lat.

AutoScrum ze względu na swoją elastyczność umożliwia bezkolizyjne wykorzystanie Modelu V oraz tablicy Kanban w celu jeszcze lepszego zarządzania projektem. Model V jest szczególnie wskazany w branży samochodowej, gdzie dużą wagę przykładą się do jakości końcowego produktu, który musi spełniać restrykcyjne wymagania bezpieczeństwa. Wprowadzenie wieloetapowego projektowania do cyklu życia produktu, z dedykowanymi testami weryfikacyjnymi i walidacyjnymi dla każdego etapu pozwala wyłapać wszelkie krytyczne błędy na etapie produkcyjnym. Z kolei wykorzystanie tablicy charakterystycznej dla Kanbana w ramach pracy zespołu pozwala utrzymać zaangażowanie i koncentrację zespołu na właściwym celu. Trzeba mieć na uwadze, że taka tablica wymaga wprowadzenia kilku sztywnych reguł zgodnych z zasadami pracy w Scrum⁹⁴ jak brak możliwości zmiany zadań w trakcie Sprintu. Wykorzystywanym narzędziem, wspomagającym zarządzanie zadaniami jest wspomniana już Jira –

⁹³ Ze względu na brak ustaleń z przedstawicielami firmy na temat wykorzystaniu wizerunku firmy, nazwa firmy i wszelkie poufne dane zostały pominięte.

⁹⁴ Taka tablica ma swoją nazwę – Scrum Board.

popularne oprogramowanie służące do organizacji pracy i śledzenia postępu projektów prowadzonych zgodnie z Agile.

Projekty prowadzone przez firmę są duże, cykl życia trwa średnio od trzech do pięciu lat, a klientami są czołowe firmy samochodowe. Przykładowym projektem może być zaprojektowanie we współpracy z klientem a następnie samodzielne stworzenie komputera pokładowego dla konkretnego modelu samochodu, którego debiut klient zaplanował jest w niedalekiej przyszłości.

Pierwszym wyzwaniem związanym z wprowadzeniem Scrum w korporacji jest utrudniona komunikacja. Metodyki zwinne kładą na nią duży nacisk i oczekują częstych interakcji pomiędzy pracownikami i interesariuszami. W typowym projekcie uczestniczy wiele zespołów porzucanych po całym świecie. Współpraca w takich warunkach wymaga ustalenia stałego zakresu godzin, w których można się komunikować. Każdy projekt ma uzgodnione ramy czasowe – część wspólną czasu pracy współpracowników z różnych stref czasowych (np. z USA, Indii i Polski). Przykładowo może to być zakres od godziny 10 do 15. Ludzie pracujący w danym projekcie są zobligowani do bycia dyspozycyjnym w tym czasie.

Kolejnym wyzwaniem jest przekonanie pracowników do podejścia zwinnego, zgodnego z Manifestem Agile. Firmie udało się to zrealizować dzięki licznym szkoleniom. Dało to podstawę do dalszego rozwoju zwinnego środowiska pracy. Firma zatrudniła Scrum Masterów w celu zorganizowania rzetelnych szkoleń pracowniczych z metodyk zwinnych, nadzorowania wprowadzanych zmian w strukturach firmy i rozwiązywania wątpliwości wśród pracowników, dotyczących kompetencji w nowym środowisku pracy. Bez wątpienia dobrym pomysłem było odświeżenie systemu szkoleń oraz nacisk na udział pracowników w najważniejszych warsztatach dotyczących procesów firmowych i metod pracy w cyklu życia produktu. Udział pracowników każdego szczebla w takich szkoleniach znacząco zwiększa ich wiedzę o firmie. Pozwala im lepiej zrozumieć projekt oraz cel wykonywanej przez nich pracy, a także łatwiej zaakceptować potencjalne opóźnienia i zmiany projektowe, jeśli takie nastąpią. Firma nie wymaga obowiązkowej obecności pracowników na każdym szkoleniu. Niektóre są wymagane tylko dla kadry

menedżerskiej, ale firma zachęca również tych chcących lepiej zrozumieć procesy zachodzące w firmie do wzięcia w nich udziału.

Kolejnym aspektem związanym z wprowadzeniem Scrum jest stworzenie jasnego podziału obowiązków poszczególnych zespołów. W typowym projekcie można wyróżnić trzy podstawowe techniczne zespoły, różniące się kompetencjami:

- zespół inżynierów ds. systemów – wspomagający pracę Właściciela Produktu, odpowiedzialni za kontakt z klientem i wspólne ustalanie wymagań produktu. Stanowią połączenie pomiędzy zespołem deweloperskim a klientami. W przypadku dużych projektów, jedna funkcjonalność może się składać z wielu wymagań dotyczących różnych dziedzin jak elektroniki, programowania czy mechaniki, stąd potrzeba wsparcia Właściciela Produktu w sprawowaniu pieczy nad zapewnieniem poprawności poszczególnych elementów założeń projektowych. Od tego zależy jakość ich implementacji i walidacji przez Zespół Deweloperski;
- zespół inżynierów ds. oprogramowania – odpowiedzialni za tworzenie produktu zgodnie z wymaganiami oraz podstawowe testy jednostkowe pojedynczych elementów programu. Zgodnie z zasadami Scrum można ich zaliczyć do Zespołu Deweloperskiego;
- zespół inżynierów ds. testów – odpowiedzialni za testowanie modułów i gotowych funkcjonalności, stworzonych przez inżynierów oprogramowania. Ich zadaniem jest zapewnienie, że dana funkcjonalność spełnia wymagania narzucone przez klienta. Również stanowią część Zespołu Deweloperskiego;

Zespoły tworzą pracownicy jednej specjalizacji, maksymalnie kilkunastoosobowe nadzorowane przez jednego lidera, który również jest członkiem zespołu – może to być doświadczony pracownik, który brał udział w kilku projektach, dobrze obeznany w metodach pracy i narzędziach.

W strukturze organizacyjnej bezpośrednio nad zespołem jest menedżer, który nadzoruje od jednego do kilku zespołów jednej kompetencji, biorących udział w różnych projektach. Jego zadaniem jest dbanie o komfort pracy zespołu oraz informowanie

o strategicznych zmianach w firmie, oraz postanowień projektowych na najwyższych szczeblach. Każdy pojedynczy zespół sam organizuje swoją pracę co jest zgodne z metodyką zwinnym.

Całkowicie nowym i niespodziewanym wyzwaniem jest sytuacja pandemiczna, wymuszająca warunki pracy przy projektach. Można stwierdzić, że AutoScrum funkcjonuje bardzo dobrze, również w obecnych warunkach. Dzięki regularnym spotkaniom zespołowym przeprowadzanym w formie wideokonferencji, praca zdalna nie powoduje opóźnień w przepływie informacji i realizacji zadań. Z tego powodu firma pozostawiła większości zespołom możliwość dalszej pracy z domu. Co prawda podejście zwinne zakłada, że rozwiązywanie problemów twarzą w twarz jest preferowaną opcją. Można założyć, że twórcy nie przewidywali, że w przyszłości spotkania będzie możliwość organizowania przez Internet na taką skalę bez problemów z łączem. Wideokonferencja w dużym stopniu spełnia ten punkt manifestu Agile.

Przykładowa praca nad projektem w metodyce Scrum z punktu widzenia pracownika jednego z zespołów projektowych wygląda w następujący sposób. Wymagania produktu i oczekiwane funkcjonalności są przygotowywane przez Właścicieli Produktu i zespoły inżynierów ds. systemów we współpracy z klientem. Ich obowiązkiem jest stworzenie i późniejsze edytowanie bazy danych (Rejestru Produktu) ze zbiorem właściwość tworzonego systemu. Każdy element Rejestru, poza opisem wymagania zawiera informacje o tym kto odpowiada za jego pokrycie (np. programiści czy testerzy), oraz od której wersji oprogramowania ma ono zostać spełnione.

Oprogramowanie jest przygotowywane małymi krokami, poprzez tworzenie i wprowadzanie konkretnych funkcjonalności z Rejestru Produktu, określonych przed rozpoczęciem Sprintu podczas Planowania. Implementacją zajmuje się zespół inżynierów ds. oprogramowania. Następnie nowa wersja oprogramowania trafia do zespołu testerów, których obowiązkiem jest sprawdzenie czy stworzony do tej pory fragment systemu zachowuje się zgodnie z wymaganiami zamieszczonymi w bazie danych. Może to być przykładowo sprawdzenie czy komunikacja pomiędzy komputerem pokładowym a czujnikami odbywa się poprawnie po wcześniejszym przeprowadzeniu restartu

systemu. Tworzone są przypadki testowe, które mogą dać wynik pozytywny (test dał oczekiwany wynik) lub negatywny (wynik testu nie spełnia oczekiwań). Z wyników przygotowywany jest raport, który trafia do Właścicieli Produktu. Pozwala on wyciągnąć wnioski na temat postępu prac, pokrycia wymagań, oraz tych funkcjonalności, które sprawiają największe problemy. Dzięki temu menedżerowie projektu mogą lepiej zaplanować przebieg kolejnych Sprintów i ilość oczekiwanych Przyrostów. Wprowadzane są zmiany do Rejestru i iteracja zaczyna się od nowa wraz z kolejnym Sprintem.

Warto nadmienić, że praca odbywa się przy ścisłej współpracy inżynierów systemu, oprogramowania i testu. Jakikolwiek wątpliwości, czy problemy występujące podczas pracy przy danej funkcjonalności są rozwiązywane we wspólnym gronie ludzi z nią związanych. Może się zdarzyć błąd, że wymaganie zostało przypisane do złej osoby, lub dana funkcjonalność nie jest już wspierana. Dzięki zaangażowaniu wielu pracowników o różnych kompetencjach znacznie skuteczniej przebiega wykrywanie i naprawianie takich problemów.

Poza nieregularnymi spotkaniami pracowników różnych specjalizacji w celu rozpoznania i naprawy błędów, odbywają się również codzienne spotkania w obrębie jednego zespołu (Daily Scrum). Prowadzi je lider zespołu, a ich celem jest wymiana informacji pomiędzy członkami zespołu na temat przebiegu prac nad poszczególnymi funkcjonalnościami. Lider bierze udział w spotkaniach z przedstawicielami klienta i Właścicielami Produktu, co pozwala mu przedstawić swojemu zespołowi aktualną sytuację projektową w tym kwestie aktualnych priorytetów i opóźnień. Jego zadaniem jest zapewnienie, że zespół jest w stanie realizować swoje zadania na czas, poprzez odpowiednie zorganizowanie pracy jego członków.

Decyzje podjęte na wszystkich spotkaniach są na bieżąco raportowane za pomocą specjalistycznego oprogramowania wspomagającego zarządzanie projektami. Dostęp do notatek mają wszyscy pracownicy zespołu i interesariusze.

Podsumowując, duże znaczenie w skutecznym wdrażaniu metod zwinnych ma przygotowanie rzetelnego i skutecznego systemu szkoleń dla pracowników. Początkowo większość może uważać zasady zwinne za zbyt skomplikowane i bezsensowne, ale po

wprowadzeniu i pracy w tym środowisku, pracownicy zazwyczaj przekonują się do ich skuteczności. Warto zwrócić uwagę, że metoda Scrum choć ma swoje sztywne zasady, jest na tyle elastyczna, że można ją połączyć z innymi metodykami, wspierającymi procesy jak Lean, Kanban i Model V. Pozwala to na efektywną pracę nawet przy dużych projektach, przy których pracują setki ludzi. Należy przy tym pamiętać o organizowaniu częstych spotkań – codziennych, organizacyjnych w ramach jednego zespołu, jak i tych angażujących pracowników o różnych specjalizacjach w przypadku wystąpienia problemów z tworzoną funkcjonalnością.

Ponadto z obserwacji można stwierdzić, że praca nad oprogramowaniem samochodowym wiąże się z licznymi zmianami w trakcie realizacji co sprawia, że tradycyjna metoda zarządzania projektem typu Waterfall nie byłaby wskazana. W skrajnym przypadku klient z powodów finansowych może zrezygnować z projektu, który jest już na zaawansowanym etapie, a w zamian zdecydować się na stworzenie podobnego, tańszego rozwiązania, opracowanego na bazie dotychczasowych prac. Takie wydarzenia wymagają od zespołów elastyczności i umiejętności szybkiego dostosowania się do nowej sytuacji. Dotychczasowe doświadczenia pokazują, że Scrum świetnie sprawdza się w takich warunkach pracy.

4.2 Studium przypadku: Tworzenie systemu IT dla Dutch Railways (firma Xebia) – przykład zmiany Waterfall na Scrum

Projekt dotyczył stworzenia systemu centralnej kontroli wyświetlaczy i głośników z informacją o dostępnych połączeniach kolejowych. Pierwsze podejście dostarczenia rozwiązania było zarządzane w tradycyjny sposób z wykorzystaniem Waterfall. Klient dostarczył szczegółowe specyfikacje ówczesnemu dostawcy IT i oczekiwał, że otrzyma gotowy system bez swojego większego zaangażowania. Niestety z powodu licznych problemów, pomimo kilku lat pracy nie udało się dostarczyć działającego produktu, a klient ostatecznie podjął decyzję o zerwaniu współpracy. Można stwierdzić, że wina po części leżała po stronie klienta z powodu braku jakiegokolwiek zaangażowania w pracę nad projektem. Z drugiej strony, Waterfall nie wymaga takiej interakcji po rozpoczęciu

projektu, więc ani zespół IT ani klient nie podejmowali się wspólnej dyskusji na temat rozwoju projektu w trakcie jego trwania.

Po pierwszej porażce projektowej klient podjął decyzję o zaangażowaniu firmy Xebia do zbudowania systemu od podstaw. Tym razem zdecydowano, że zostanie wdrożone podejście zwinne a projekt będzie przebiegał zgodnie z metodyką Scrum. Projekt dość sprawnie udało się doprowadzić do końca i dostarczyć klientowi wymagane oprogramowanie ⁹⁵.

Z przebiegu prac projektowych firmy Xebia można wyciągnąć szereg interesujących wniosków dotyczących pracy w metodyce Scrum:

- znalezienie odpowiedniej osoby na stanowisko Właściciela Produktu z szeroką wiedzą specjalistyczną może być ciężkie. Przy dużych, skomplikowanych projektach, gdzie jest wiele wymagań pokrycia dobrze jest wybrać kilku Właścicieli Produktu, żeby zapewnić osobę o odpowiedniej wiedzy czuwa nad założeniami projektu,
- każda estymacja jest lepsza niż żadna – nawet gdy dostępne jest niewiele informacji, ważne jest, żeby Rejestr Produktu był zawsze kompletny i oszacowany. Pomoże to lepiej ustalać terminy i organizować Sprints,
- Scrum jest doskonale przystosowany do pracy w rozproszonych zespołach. Holenderska firma Xebia była w stanie pracować efektywnie współpracując z zespołem programistów w Indiach. Wspomagano się licznymi wideokonferencjami,
- dobrą praktyką jest stworzenie osobnego zespołu nietechnicznego, który skupi się na zadaniach nie pasujących do Sprintu, dzięki czemu Zespół Deweloperski będzie mógł się w pełni skupić na swojej pracy,
- testowanie jest niezbędne, aby dostarczać oprogramowanie w sposób przyrostowy bez opóźnień wynikających z błędów znalezionych zbyt późno. Dodatkowo warto zainwestować w automatyzację testów.

⁹⁵ Pełna analiza przebiegu projektu dostępna pod linkiem: <https://www.infoq.com/articles/dutch-railway-scrum/>

Firma Xebia również napotkała problemy podczas pracy, są one nieuniknione praktycznie w każdym projekcie IT, ale tym razem udało się je rozwiązać lub pójść na kompromis dzięki elastyczności metody Scrum.

5 Podsumowanie i wnioski

W tym rozdziale zostaną przedstawione zalety i wady wykorzystania metody Scrum, opracowane bazując na zaprezentowanym opisie teoretycznym i przykładach. Następnie podjęta zostanie próba odpowiedzi na pytanie: Czy można bezrefleksyjnie wybierać Scrum do zarządzania projektami informatycznymi?

5.1 Zalety Scrum

Scrum jako metoda zgodna z podejściem zwinnym oferuje menedżerom projektu dużą swobodę w podejmowaniu decyzji o zmianach w trakcie trwania projektu. Ma to kluczowe zastosowanie zwłaszcza w projektach informatycznych, gdzie produkt jest w dużej mierze wirtualny. Klient nie zawsze jest w stanie dokładnie sprecyzować swojej potrzeby, co w połączeniu z dynamiką branży, częstymi zmianami sytuacji rynkowej oraz pojawianiem się coraz to nowszych technologii wzmacnia potrzebę bycia elastycznym i otwartym na częste zmiany planu.

Jak pokazał drugi przykład, wybór tradycyjnej metody zarządzania projektem wiąże się z ryzykiem niemożności ukończenia projektu, ponieważ budżet i czas są z góry ustalone a ich przekroczenie wiąże się z porażką. Scrum, bardziej elastyczny pod tym względem, na start bierze pod uwagę potencjalne potrzeby zmiany, które mogą wygenerować dodatkowe koszty lub spowodować przesunięcie ukończenia projektu w czasie.

Kolejną zaletą wykorzystania Scruma jest wprowadzenie jasnej struktury zespołu, gdzie każda osoba ma z góry określone kompetencje jak właściciel produktu czy członek zespołu deweloperskiego. Dodatkowo, dzięki Scrum Masterom zapewnia się, że pozostali pracownicy będą mogli być praktycznie w każdej chwili mogą rozwiązać swoje wątpliwości odnośnie do swoich obowiązków w ramach pracy zgodnie z metodą Scrum. Ostatnim argumentem przemawiającym za przewagą Scrum nad tradycyjną metodą Waterfall jest czynne uczestnictwo klienta w pracach projektowych. Zapewnia to, że dostarczony produkt będzie dokładnym odzwierciedleniem jego potrzeb. Na każdym etapie projektu klient może dostarczać swoje uwagi, które pomiędzy iteracjami Sprintu będą wspólnie dyskutowane, żeby stworzyć odpowiednie wymagania do zaimplementowania.

5.2 Krytyka Scrum

Dla równowagi warto przytoczyć również głosy krytykujące metodę Scrum. Pierwszym argumentem przeciwników jest sprawianie wrażenia przez Scrum, że poprzez szereg zasad jak, na przykład stałe iteracje produkcyjne, wymóg częstych spotkań bardzo komplikuje się pracę⁹⁶. Wśród pracowników technicznych może pojawić się obawa wprowadzenia Scrum'a, bo spodziewają się, że wiąże się to z zwiększeniem nadzoru pracy i obowiązku wykonywania wielu powtarzalnych a ich zdaniem niepotrzebnych czynności. Scrum zachęca do organizowania częstych spotkań zespołu. Nie ma pewności, że takie spotkania będą produktywne. Może być tak, że to jedna osoba wyraża swoją opinię i wpisuje zadania do Jiry, a pozostali uczestnicy biernie uczestniczą w spotkaniu i tylko się przeglądają. Takie spotkanie nie ma sensu i jest stratą czasu. Co gorsza może pogorszyć produktywność programistów poprzez częste wyrywanie ich ze skupienia.

Kolejna kwestia dotyczy zakładania zbyt ambitnych planów na początku projektu, a zwłaszcza na pierwszych Sprintach. Przy obecnych skomplikowanych systemach IT bardzo ciężko jest od razu dostarczyć dobrą, działającą funkcjonalność. Należy się liczyć z tym, że początki mogą być trudne a sama konfiguracja środowiska pracy może pochłoniąć tygodnie. Przygotowanie zbyt ambitnego Rejestru Sprintu w początkowej fazie projektu to prosta droga do popsucia morale zespołu i pogorszenia relacji z klientem. Problemy może stwarzać także Wykres spalania, który powinien być odwzorowaniem aktualnego postępu. Niestety, jest ryzyko, że nie każdy problem będzie można rozbić na mniejsze składowe, a skomplikowane zadania z powodu swojej binarności („zrobione” albo „niezrobione”) i różnym stopniom potrzebnego zaangażowania będą realizowane w sposób nierównomierny w efekcie czego wykres będzie długą linią prostą aż do zrealizowania zadań. Taki wykres nie przekazuje żadnych wartościowych informacji i nie pomaga w planowaniu kolejnych Sprintów.

Wszystkie wymienione wyżej negatywne opinie na temat Scrum powinny być wzięte pod uwagę przez zespół, przed decyzją o wykorzystaniu tej metody. Wynika z nich, że Scrum wymaga zaangażowania i wysokich kompetencji od każdego pracownika. Wprowadzenie Scrum może być o wiele bardziej skuteczne, jeśli będzie poprzedzone szkoleniami pracowników, którzy w ten sposób zrozumieją sens częstych spotkań i być może

⁹⁶ Kalkus, Scrum zabija, <https://kalkus.dev/pl/2020/03/10/scrum-zabija> (dostęp: 01.09.20)

postanowią zaplanować swoją pracę w taki sposób żeby nie kolidowała ona z Daily Scrum, w których wezmą udział w sposób aktywny. Odpowiednie przeszkolenie „zarządu projektu”, również jest bardzo wskazane. W taki sposób założenia projektowe i plan początkowych Sprintów będą przygotowane z uwzględnieniem problemów i opóźnień typowych dla każdego rozpoczynającego się projektu IT (jak na przykład przygotowanie środowiska programistycznego lub symulacji). Odnosząc się do ostatniej krytycznej opinii, Właściciel Produktu i wszyscy pomagający mu w przygotowywaniu Rejestru powinni starać się być jak najbardziej skuteczni w upraszczaniu skomplikowanych zadań, dzięki czemu wykres spalania będzie miarodajny. Niepodzielne zadania powinny być wykonywalne w obrębie jednego Sprintu, w przeciwnym razie należałoby pomyśleć nad wydłużeniem jego czasu.

5.3 Metody zwinne czy klasyczne?

Ostatecznie pojawia się pytanie czy warto inwestować w szkolenia, dużą komplikację projektu i zaryzykować z wprowadzaniem metody Scrum czy pozostać w tradycyjnym modelu kaskadowym. Odpowiedź na takie pytanie nie jest jednoznaczna, bo wybór metody powinien być uwarunkowany poprzez wiele różnych czynników, jak doświadczenie zespołu, ramy czasowe, budżet oraz co ważniejsze co jest produktem. Inna metoda sprawdzi się przy projektowaniu prostej aplikacji komórkowej a z kolei inna przy tworzeniu skomplikowanego systemu informatycznego dla banku.

Tabela 2: Porównanie użyteczności Waterfall i Scrum w zależności od charakterystyki tworzonego produktu IT⁹⁷.

Charakterystyka tworzonego systemu IT	wykorzystanie Waterfall	wykorzystanie Scrum
Niejasne wymagania klienta	słabe	bardzo dobre
Skomplikowana struktura	dobre	słabe

⁹⁷ Dennis A., Wixom B., Roth R.: System Analysis & Design 5h edition, Wiley, USA 2012, s. 59

Potrzeba wysokiej jakości i niezawodności	dobrze	dobrze
Potrzeba dostarczenia w krótkim czasie	słabe	bardzo dobre
Potrzeba zaplanowania prac nad poszczególnymi funkcjonalnościami	słabe	dobrze

Czasami może się okazać, że ani Waterfall ani Scrum nie spełniają potrzeb projektowych i możliwości zespołu. Być może warto wypróbować Programowanie Ekstremalne lub PRINCE2, lub wspomóc się metodą Kanban albo Modelem V. Jeśli to nie wystarczy istnieje więcej metod, nieprzybliżonych w niniejszej pracy. Każda ma swoje wady i zalety a decyzja o wyborze podejścia do zarządzania projektem powinna być podjęta mając na uwadze doświadczenie menedżera projektu i możliwości zespołu.

Elastyczność i umiejętność dostosowania się do sytuacji jest kluczowa. Dobrym przykładem jest British Airways, który doświadczył problemów z dostarczaniem produktów na czas pomimo zdolnego i ambitnego zespołu⁹⁸. Ówczesny, główny inżynier oprogramowania – Mike Croucher zaproponował wprowadzenie zasad Agile. Zostało to zrealizowane w bardzo przemyślany sposób, stopniowo wprowadzając podejście zwinne do pojedynczych działów i projektów. Dokonane zmiany znacząco przyspieszyły procesy, ale ostatecznie tylko 25% organizacji przestawiło się na Agile. Część zespołów preferowało pracę w dotychczasowych warunkach a zarząd widząc, że w ich projektach to działa nie zmuszano ich do zmiany. W taki sposób dało się pogodzić oba modele w obrębie jednej organizacji, osiągając przy tym korzyść, czyli przyspieszenie w realizacji krytycznych projektów, w których metodyki tradycyjne nie działały.

Doświadczony menedżer projektu powinien przewidzieć czy w danym projekcie Waterfall będzie odpowiednią metodą, ze swoim prostoliniowym procesem, stałymi założeniami i niską potrzebą wdrożenia pracowników do procesów, a kiedy Scrum z założeniem że zmiana jest jedyną stałą. W branży IT dominują projekty wykorzystujące innowacyjne technologie, które po kilku latach już są traktowane jako przestarzałe. Ta

⁹⁸ Mondelo, Daniel J.: Where agile development works and where it doesn't: A user story., <https://www.searchsoftwarequality.com> (dostęp: 05.09.2020)

zmienność stwarza idealne warunki do wprowadzenia metodyki Scrum, stąd też nacisk do pracy zgodnie z jej założeniami jest popularne wśród firm informatycznych. Należy jednak pamiętać, że źle wprowadzony Scrum może przynieść więcej szkód niż pożytku, a przede wszystkim popsuć atmosferę w zespole, dlatego proces wprowadzenia nowego podejścia do zarządzania projektami powinien odbywać się w sposób zaplanowany krok po kroku, tak jak miało to miejsce w British Airways. Odpowiadając na pytanie postawione na początku rozdziału, nie powinno się bezrefleksyjnie wybierać Scrum do zarządzania projektami IT. Mimo wszystko ta metoda jest wskazana, jeśli są spełnione odpowiednie wymienione wielokrotnie warunki a zespół jest przeszkolony.

Na koniec warto stwierdzić, że, wielu menedżerów projektu kieruje się przede wszystkim własnym doświadczeniem. Takie podejście choć co prawda nie ustrukturyzowane, będące często mieszaniną różnych stylów zarządzania, jeśli tylko się sprawdza i pozwala realizować projekty na czas może być jak najbardziej stosowane.

Spis rysunków i tabel

Rysunki:

Rysunek 1: Prognozy wzrostów branży IT w latach 2018-2023 z uwzględnieniem podziału na sektory	3
Rysunek 2: Wydarzenia, które miały historyczny wpływ na zarządzanie projektami.	5
Rysunek 3: Dwa warianty trójkąta ograniczeń projektowych	9
Rysunek 4: Wykres przedstawiający efekt tzw. syndromu studenta	11
Rysunek 5: Przykład cyklu życia oprogramowania w podejściu tradycyjnym (model kaskadowy – Waterfall)	13
Rysunek 6: Typowy cykl życia oprogramowania w podejściu zwinnym	15
Rysunek 7: Porównanie trójkątów wyzwań projektowych w podejściu tradycyjnym i zwinnym.....	16
Rysunek 8: Macierz wyboru metodyki zarządzania projektem wg. Wysockiego.	18
Rysunek 9: Struktura Modelu V	24
Rysunek 10: Przykład tablicy Kanban, stworzonej w stylu metodyki Scrum	26
Rysunek 11: Typowa struktura Zespołu Scrum.....	29
Rysunek 12: Wydarzenia Scrum na schemacie	33
Rysunek 13: Artefakty Scrum na schemacie	39
Rysunek 14: Przykładowy wykres spalania wydania	41
Rysunek 15: Rejestr Sprintu na schemacie	43
Rysunek 16: Przyrost Produktu w metodzie Scrum	44

Tabele:

Tabela 1: Porównanie podejścia tradycyjnego i zwinnego na przykładzie metodyk Waterfall i Scrum.....	45
Tabela 2: Porównanie użyteczności Waterfall i Scrum w zależności od charakterystyki tworzonego produktu IT.	57

Bibliografia

- [1] Agile Group, Model kaskadowy, <https://agilegroup.pl/agile-mindset> (dostęp 16.08.2020)
- [2] Agile Manifesto, <https://agilemanifesto.org/iso/pl/principles.html> (dostęp 16.08.2020)
- [3] Agile247, Daily Scrum, <https://agile247.pl/daily-scrum-standup> (dostęp: 27.08.20)
- [4] AgileHunters, Artefakty Scruma, <https://agilehunters.com/3-najwazniejsze-artefakty-scruma/> (dostęp: 28.09.2020)
- [5] Chromatic.: Extreme Programming Pocket Guide: Team-Based Software Development, O'Reilly Media 2013
- [6] Dennis A., Wixom B., Roth R.: System Analysis & Design 5h edition, Wiley, USA 2012
- [7] Duncan W.R.: A Guide to the Project Management Body of Knowledge, PMI, USA 1996
- [8] Dziurzańska A., Dżega D., Olejniczak W.: Polski menedżer projektu informatycznego, Akademia Ekonomiczna, Katowice 2007
- [9] Extreme Programming, Introduction to XP, <http://www.extremeprogramming.org> (dostęp: 25.08.2020)
- [10] Frączkowski K.: Zarządzanie projektem informatycznym, Politechnika Wrocławska, Wrocław 2003, s. 11
- [11] Infoq, Xebia case study, <https://www.infoq.com/articles/dutch-railway-scrum> (dostęp: 29.08.20)
- [12] ISTQB, Certyfikowany tester – Plan poziomu podstawowego, 2011
- [13] IT Consulting, Cykl życia oprogramowania a wybór metodologii, https://it-consulting.pl/autoinstalator/wordpress/2011/03/22/co-wybrac-czyli-cykl-zycia-projektu-tworzenia-oprogramowania/#Wybor_wlasciwej_metodologii (dostęp: 17.08.2020)
- [14] Jarocka K.: Projekt z kapelusza, czyli skutki zakrzywiania projektowej rzeczywistości, <https://kjarocka.pl/zarzadzanie-projektami/projekt-z-kapelusza-czyli-skutki-zakrzywiania-projektowej-rzeczywistosci> (dostęp 12.10.20)
- [15] Job Growth and Talent Gap Report 2017-2027, PMI, USA 2017

- [16] Kalkus, Scrum zabija, <https://kalkus.dev/pl/2020/03/10/scrum-zabija> (dostęp: 01.09.20)
- [17] Kanbanize, Introduction to Kanban, <https://kanbanize.com/kanban-resources/getting-started> (dostęp: 26.08.2020)
- [18] Kanbantool, Wprowadzenie do Scrumban, <https://kanbantool.com/pl/scrumban-kanban-i-scrum> (dostęp 27.08.2020)
- [19] Kerzner H.: Project Management: A systems approach to Planning, Scheduling and Controlling, Wiley, USA 2003
- [20] Kloze M.: Kluczowe aspekty zarządzania projektami – Deloitte, Prezentacja szkoleniowa Akademii Konsultingu 2009
- [21] Korzonkowski K.: CCPM jako metoda planowania i kontroli projektów, Capgemini, Wrocław 2014
- [22] Lock D.: Podstawy zarządzania projektami, PWE, Warszawa 2009
- [23] Mielcarek K., Pławiak-Mowna A.: Wspomaganie realizacji grupowego projektu IT w oparciu o system SVN. Metody Informatyki Stosowanej, nr 2(27), Polska Akademia Nauk Oddział w Gdańsku, Szczecin 2011
- [24] Mondelo, Daniel J.: Where agile development works and where it doesn't: A user story., <https://www.searchsoftwarequality.com> (dostęp: 05.09.2020)
- [25] O'Neill E.: User-Developer Cooperation in Software Development. Springer-Verlag, London 2001
- [26] OGC, PRINCE2 - Skuteczne zarządzanie projektami, TSO, Norwich UK 2009
- [27] Orłowski C., Ziółkowski A.: Wsparcie doboru metod zarządzania , politechnika Gdańska, Gdańsk 2010
- [28] Pawlak M.: Zarządzanie projektami, PWN, Warszawa 2006
- [29] Pawlak R.: Testowanie Oprogramowania. Podręcznik dla Początkujących, Helion, Gliwice 2014
- [30] Pichler R.: Zarządzanie projektami ze Scrumem, Helion, Gliwice 2014
- [31] PMI, What is project management, <https://www.pmi.org/about/learn-about-pmi/what-is-project-management> (dostęp: 11.08.20)
- [32] Schwaber K., Sutherland J.: The Scrum Guide 2017
- [33] Sierackiewicz M.: Technical Leadership, Helion, Gliwice 2016

- [34] SJP, Definicja słowa „projekt”, <https://sjp.pwn.pl/> (dostęp 10.08.20)
- [35] Słownik wyrażen̄ związanych z testowaniem, Definicja cyklu życia oprogramowania, <https://jsi.org/slowo/cykl-zycia-oprogramowania> (dostęp 14.08.2020)
- [36] Sobestiańczyk T.: *Metodyka Zarządzania Projektami – PRINCE2*, Politechnika Częstochowska, Częstochowa 2012
- [37] Społeczeństwo informacyjne w Polsce. Wyniki badań statystycznych z lat 2014-2018, GUS, Szczecin 2018
- [38] Stabryła A., Wawak S.: *Metody badania i modele rozwoju organizacji*, UEK, Kraków 2012
- [39] Strojny K., Szmigiel M.: Analiza porównawcza podejść w zakresie zarządzania projektami, „Modern Management Review” Vol. XX, No. 22(3), 2015
- [40] Trocki M.: *Zarządzanie projektami*, PWE, Warszawa 2009
- [41] Wysocki R. K., McGary R.: *Efektywne Zarządzanie Projektami: Tradycyjne, zwinne, ekstremalne*, wyd. 3, Helion, Gliwice 2005
- [42] Wysocki R. K.: *Effective Project Management: Traditional, Agile, Extreme*, Wiley Publishing Inc., Indianapolis USA 2009
- [43] *Zarządzanie Projektami Online*, Trójkąt Projektowy, <https://zarzadzanieprojektami.online/2018/trojkat-projektowy> (dostęp 12.10.20)