

ZWINNOŚĆ

**Od zwinnych zespołów
do zwinnego zarządzania**

Paweł Wyrzębowski

ZWINNOŚĆ

**Od zwinnych zespołów
do zwinnego zarządzania**

Recenzje

Mariusz Hofman

Jacek Strojny

Redakcja językowa

Jacek Skawiński

© Copyright by Paweł Wyrozębski & Szkoła Główna Handlowa w Warszawie, Warszawa 2021

Wszelkie prawa zastrzeżone. Kopiowanie, przedrukowywanie i rozpowszechnianie całości lub fragmentów niniejszej publikacji bez zgody wydawcy zabronione.

Wydanie I

ISBN 978-83-8030-519-9

Oficyna Wydawnicza SGH – Szkoła Główna Handlowa w Warszawie

02-554 Warszawa, al. Niepodległości 162

www.wydawnictwo.sgh.waw.pl

e-mail: wydawnictwo@sgh.waw.pl

Projekt i wykonanie okładki

Magdalena Limbach

Skład i łamanie

DM Quadro

Druk i oprawa

QUICK DRUK s.c.

Zamówienie 180/XII/21

SPIS TREŚCI

WSTĘP	7
-------------	---

1	WPROWADZENIE DO ZWINNEGO ZARZĄDZANIA PROJEKTAMI	11
1.1.	Geneza podejścia	11
1.2.	Agile Manifesto	17
1.3.	Charakterystyka metod zwinnych	19
1.4.	Korzyści ze stosowania metod zwinnych	25
1.5.	Istota i próba zdefiniowania pojęcia zwinności	28
1.6.	Aspekty metodologiczne zwinności	32
1.7.	Fale rozwoju zwinności	37
	Podsumowanie	45

2	PRZEGLĄD NAJWAŻNIEJSZYCH ZWINNYCH METODYK ZARZĄDZANIA PROJEKTAMI	47
2.1.	Evolutionary Project Management (EVO)	47
2.2.	Agile Project Management (APM)	53
2.3.	Programowanie Ekstremalne – Extreme Programming (XP)	64
2.4.	Crystal Light	71
2.5.	Feature Driven Development (FDD)	76
2.6.	SCRUM	85
2.7.	Adaptive Software Development (ASD)	98
2.8.	Ekstremalne zarządzanie projektami – Extreme Project Management (XPM) ...	101
2.9.	AgilePM (Dynamic System Development Method)	109
2.10.	Agile PMI – Software Extension to the PMBOK Guide Fifth Edition	114
2.11.	PRINCE2 Agile	135

3	METODY SKALOWALNEJ ZWINNOŚCI	141
3.1.	Scaled Agile Framework (SAFe)	144
3.2.	SCRUM of SCRUMs (SoS)	150
3.3.	Nexus	152
3.4.	SCRUM@Scale (S@S)	156
3.5.	LeadingAgile (LA)	160
3.6.	Sustainable Cultural Agile Release in the Enterprise (SCARE)	167

SPIS TREŚCI

4	ZWINNE TRANSFORMACJE PRZEDSIĘBIORSTW	171
	4.1. Wprowadzenie	171
	4.2. Wyzwania zwinnej transformacji	174
	4.3. Modele zwinnej transformacji	183
	4.4. Czynniki sukcesu zwinnej transformacji	195
	ZAKOŃCZENIE	199
	BIBLIOGRAFIA	201

WSTĘP

W lutym 2021 r. minęło dwadzieścia lat od ogłoszenia światu Manifestu zwinnego wytwarzania oprogramowania. Dwadzieścia lat mocno zdewaluowało prognozy formułowane na progu XXI wieku. Myślę, że również sygnatariusze owego Manifestu nie byli by w stanie przewidzieć jaki wpływ na rzeczywistość odniesie ich, niejako „garażowa”, koncepcja zwinnego programowania.

Przez dwie dekady koncepcji zwinności towarzyszył szalenie intensywny rozwój, zaś mimo początkowych obiekcji i zastrzeżeń zwinne programowanie stało się zwinnym zarządzaniem projektami, a dzisiaj również zwinnym zarządzaniem we wszystkich obszarach problemowych funkcjonowania przedsiębiorstw. Zwinność jest dziś w głównym nurcie biznesu i nauk o zarządzaniu zmagających się ze zmiennym i dynamicznym światem VUCA¹. Problematyka zwinności wywiera znaczny wpływ już nie tylko na działania w ramach tworzenia nowych rozwiązań programistycznych, ale odważnie weszła także w inne, dotychczas zarezerwowane dla bardziej klasycznych metod, obszary organizacji, w tym zarządzanie strategiczne, rachunkowość czy operacje i utrzymanie rozwiązań.

Fundamentalna zmiana nie byłaby możliwa bez wysiłku rzeszy twórców, praktyków, ale też i przedstawicieli kierownictwa firm wspólnie zaangażowanych w budowanie, tworzenie, testowanie oraz doskonalenie szeregu koncepcji wspólnie klasyfikowanych jako zwinne. Część z ich innowatorskich pomysłów nie przetrwała próby czasu i odeszła w cień, a niektóre z powodzeniem rozwinęły się, zdobywając status mniej lub bardziej oficjalnych standardów w biznesie. Ten dynamiczny rozwój miał w dużej mierze charakter pragmatyczny oparty był na pomysłowości, wiedzy, doświadczeniu i pętli uczenia się promotorów zwinności. W dużo mniejszym stopniu, a zdaniem niektórych nawet w nikłym, towarzyszyło mu wsparcie ze strony środowiska naukowego, w szczególności nauk

¹ P. Wyrozębski, H. Cichocki, *Zwinna transformacja w dużej skali*, w: *Wybrane aspekty zarządzania procesami, projektami i ryzykiem w przedsiębiorstwach*, E. Sońta-Drączkowska, I. Bednarska-Wnuk (red.), Wydawnictwo Uniwersytetu Łódzkiego, Łódź 2020, s. 149–159; O.G. Mink et al., *Change at Work: A Comprehensive Management Process for Transforming Organizations*, Jossey-Bass Publishers, San Francisco 1993, za: W.E. Whiteman, *Training and educating army officers for the 21st century: Implications for the United States Military Academy*, Defense Technical Information Center, Fort Belvoir 1998; US. Army Heritage & Education Center, <http://usawc.libanswers.com/faq/84869> (dostęp: 31.01.2020).

o zarządzaniu. Rodzi to wiele problemów i stawia bariery w zrozumieniu, dalszym rozwoju, czy obiektywnej ocenie wyników uzyskiwanych dzięki zwinności.

Taka konstatacja dała impuls do podjęcia wysiłku opracowania niniejszej publikacji. Jej celem jest podsumowanie dwudziestolecia zwinności i przedstawienie możliwie kompleksowego ujęcia problematyki zwinnego zarządzania w formie monograficznego, naukowego opracowania. Celem przyświecającym autorowi jest prezentacja genezy, idei i postulatów oraz metod, technik, podejść oraz towarzyszących im problemów i rozwiązań organizacyjnych. W obserwacji piszącego te słowa jest to nadal obszar deficytowy w polskiej literaturze naukowej, tworzący istotną lukę badawczą.

W niniejszym opracowaniu przyjęto tryb postępowania badawczego zgodny z zasadami metodycznymi obowiązującymi w naukach o zarządzaniu². Punktem wyjścia rozważań stały się pytania badawcze:

- 1) Na czym polega istota zwinności w zarządzaniu i naukach o zarządzaniu?
- 2) Czym charakteryzują się i jakie rozwiązania proponują metody klasyfikowane jako zwinne?
- 3) W jaki sposób skalować zwinność na duże zespoły, przedsięwzięcia i organizacje?
- 4) Jakie wyzwania stoją przed organizacjami pragnącymi zaadaptować do swoich potrzeb podejścia zwinne i jak przeprowadzić taką transformację z sukcesem?

Odpowiedź na tak postawione pytania badawcze oraz wypełnienie luki wiedzy nastąpi poprzez realizację szczegółowych celów badawczych takich jak:

- 1) określenie ram pojęciowych i struktury problematyki badanego zagadnienia,
- 2) przeprowadzenie kompleksowej analizy źródeł fachowych oraz naukowych zwinności
- 3) synteza wiedzy w monograficzne ujęcie poszczególnych obszarów problemowych
- 4) sformułowanie ocen i rekomendacji o charakterze teoriiotwórczym, metodycznym i pragmatycznym, użytecznych zarówno dla świata biznesu, jak i nauki.

Przyjęta strategia badawcza korzysta z metod badań literaturowych oraz empirycznych. W badaniach literaturowych wykorzystano zarówno źródła polsko-, jak i obcojęzyczne, w tym m.in.: wydawnictwa zwarte poświęcone problematyce zwinności, zwłaszcza zwinnego zarządzania, artykuły i raporty z badań obcych, materiały zawarte w bazach czasopism i publikacji naukowych (m.in. ScienceDirect, ProQuest, EBSCO i inne), materiały szkoleniowe, prezentacje i materiały z konferencji oraz inne materiały dostępne w zasobach fizycznych i sieciowych (bazy OPI, bazy Biblioteki Narodowej).

Niniejsza publikacja uporządkowana została w czterech rozdziałach. W rozdziale pierwszym przedstawiono genezę, historię oraz główne założenia koncepcji zwinne-

² J. Pieter, *Metodologia pracy naukowej*, Wydawnictwo Wyższej Szkoły Pedagogicznej w Katowicach, Katowice 1965, s. 139–142; C. Frankfort-Nachmias, D. Nachmias, *Metody badawcze w naukach społecznych*, Zysk i S-ka, Poznań 2001, s. 36; J. Kisielnicki, *Zarządzanie projektami badawczo-rozwojowymi*, Wolters-Kluwer, Warszawa 2013; Podstawy metodologii badań w naukach o zarządzaniu, W. Czakon (red.), Wolters-Kluwer, Warszawa 2011; M. Saunders, P. Lewis, *Doing research in business and management*, Prentice Hall, Hoboken 2012.

go zarządzania. Podjęto się próby zdefiniowania zwinności, uporządkowania metodologicznego koncepcji oraz przedstawienia głównych etapów jej rozwoju na przestrzeni ostatnich dekad. Rozdział pierwszy ma charakter wprowadzający i przedstawiający istotę omawianej problematyki.

Rozdział drugi poświęcony jest analizie warstwy metodycznej zwinności na poziomie zespołów i projektów zwinnych. Na kolejnych jego stronach zaprezentowano przegląd najważniejszych metod, technik i metodyk zwinnego zarządzania. Omówienie to ma na celu zilustrowanie bogactwa podejść metodycznych oraz krytyczną ocenę proponowanych narzędzi z perspektywy ich silnych i słabych stron.

W rozdziale trzecim autor rozpatruje problematykę skalowania zwinności. Sukces przedsięwzięć o relatywnie małej skali skłonił menedżerów do poszukiwania możliwości powtórzenia tychże osiągnięć również w przypadku rozwiązań dużych, a także programów, zarządzania portfelem oraz całymi organizacjami. Jednakże skalowanie dalekie jest od automatyzmu. Pojawiające się specyficzne problemy w tym zakresie, jak również odrębne rozwiązania wraz z ich oceną stanowią treść tegoż rozdziału.

Rozdział czwarty kontynuuje wątki rozpoczęte w rozdziałach poprzednich. Wobec popularyzacji podejścia zwinnego oraz chęci adaptacji go przez liczne organizacje, nie tylko z branży informatycznej, konieczne jest dokładne przestudiowanie uwarunkowań, przebiegu oraz czynników sukcesu i porażki tzw. zwinnych transformacji, czyli przedsięwzięć mających na celu fundamentalną zmianę filozofii działania organizacji w kierunku szybkości, elastyczności, zwinności. Atrakcyjność i swoista moda na zwinność powodują rosnące zainteresowanie takimi wdrożeniami, jednak naukowa analiza literatury i doświadczeń w tym względzie sugerują zachowanie daleko idącej ostrożności i uwagi przy realizacji takich przedsięwzięć ze względu na poziom skomplikowania, zasięg oddziaływania zmian oraz ryzyko porażki. W rozdziale czwartym problematyka ta została przeanalizowana szczegółowo i z należytą uwagą.

Prezentowaną publikację kończy podsumowanie, w którym autor zbiera uzyskane w całej monografii obserwacje oraz podejmuje próby wskazania scenariuszy, potencjalnych kierunków dalszego rozwoju dziedziny mającej pełne prawo określać się jako dojrzała: zarówno w wymiarze metodycznym, jak i organizacyjnym w szerokiej społeczności praktyków oraz badaczy-naukowców.

Niniejsza publikacja jest efektem kilkuletniej pracy naukowej i dydaktycznej autora w Katedrze Zarządzania Projektami, wchodzącej w skład Instytutu Zarządzania SGH, czołowego ośrodka naukowego w Polsce zajmującego się badaniem problemów zarządzania złożonymi przedsięwzięciami (projektami). Jest nowym, oryginalnym wkładem w stan wiedzy w tej dziedzinie. Autor pozostaje w nadziei na jej przychylne przyjęcie zarówno w środowisku naukowym, jak i biznesowym.

1 WPROWADZENIE DO ZWINNEGO ZARZĄDZANIA PROJEKTAMI

1.1. Geneza podejścia

Z perspektywy rozwoju zarządzania, zarówno w aspekcie praktyki biznesowej, jak i wiedzy naukowej, ostatnie lata zdominowane zostały przez zagadnienie „zwinności” – początkowo „zwinnego zarządzania projektami”, a obecnie coraz częściej, po prostu „zwinnego zarządzania”. Zwinność stała się bowiem słowem kluczem, Świętym Graalem współczesnego biznesu poszukującego rozwiązań pozwalających wzrastać w dynamicznym i coraz bardziej ucyfrowionym świecie XXI wieku. Zanim jednak zaprezentowane zostaną pogłębione studia nad fenomenem zwinności, należy przestudiować jego genezę, źródła oraz kluczowe zagadnienia związane z tym pojęciem.

W popularnym ujęciu za początek koncepcji zwinnego zarządzania uznaje się najczęściej rok 2001 oraz prezentację szerszej społeczności profesjonalistów zarządzania w IT treści tzw. Manifestu zwinnego wytwarzania oprogramowania (ang. Manifesto for Agile Software Development), popularnie nazywanego w późniejszym czasie Manifestem zwinności (ang. Agile Manifesto). Do samego Manifestu powrócimy jeszcze w dalszej części opracowania, jednak należy w tym miejscu podkreślić, iż źródła tej koncepcji należy szukać wcześniej, zaś sama deklaracja nie była początkiem, lecz ważnym kamieniem milowym ewolucji podejść zwinnych, które rozwijały się z powodzeniem od co najmniej kilku dekad. Źródłem koncepcji zwinnego zarządzania jest bowiem krytyczne podejście do obowiązujących i szeroko akceptowanych zasad zarządzania zgodnie z tzw. kaskadowym/liniowym modelem rozwoju projektu (ang. *waterfall*) oraz praktyk kierowania zespołami pracowniczymi opartych na daleko posuniętej kontroli kierowników nad pracą zespołów „ocierających się” o mikrozarządzanie.

Opinie krytyczne uwidoczniły się zwłaszcza w efekcie publikacji pierwszej (z 1994 r.) i kolejnych edycji głośnego raportu Standish Group zatytułowanego *Chaos Report*. Badanie obejmujące swoim zasięgiem kilka tysięcy projektów informatycznych unaocznilo

towarzyszący im chaos, który stał się głównym motywem raportu³. Mimo rozwijających się już wtedy od lat najlepszych praktyk zarządzania projektami i prężnie działających stowarzyszeń fachowych zbadany odsetek projektów zakończonych sukcesem (rozumianym jako realizacja projektu na czas, w planowanym zakresie i kosztach) wynosił zaledwie 16,2%. Projekty, których założenia nie zostały spełnione, stanowiły ponad połowę próby (52,7%), zaś te zamknięte przed czasem – niemal jedną trzecią (31,1%). Autorzy raportu, mimo, iż nie byli bezpośrednio związani ze środowiskiem rozwijającym podejścia zwinne, zdecydowanie zwrócili uwagę społeczności profesjonalistów w zakresie zarządzania projektami IT na niedomagania istniejących rozwiązań prowadzące do tak słabych wyników.

Tradycyjne podejście do realizacji projektów rozwoju oprogramowania oparte było na tzw. modelu kaskadowym. Historycznie autorstwo modelu przypisuje się w pierwszej kolejności H.D. Beningtonowi⁴, zaś w drugiej późniejszemu dr. Winstonowi W. Royce'owi, który zmodyfikował go w 1970 r. w artykule pt. *Managing the Development of Large Software Systems*, dodając pętle i sprzężenia zwrotne⁵. Organizacja pracy jest w nim oparta na założeniu, iż analiza musi poprzedzać kodowanie, zaś sam projekt tworzenia oprogramowania powinien być podzielony na kolejne, jasno wyodrębnione rodzajowo fazy rozwoju poprzez: gromadzenie wymagań systemowych, gromadzenie wymagań oprogramowania, analizę, projektowanie, kodowanie, testowanie i wdrożenie. Stosowanie takiego modelu miało swoje zdecydowane zalety. Należy do nich zaliczyć m.in.:

- 1) relatywną prostotę konstrukcji i łatwość we wdrożeniu,
- 2) ustrukturyzowane, łatwo zarządzalne podejście, w którym każda faza składa się z przeglądów i określonych rezultatów,
- 3) liniowość – fazy nie nakładają się i są realizowane pojedynczo, jedna po drugiej,
- 4) łatwość stosowania w przypadku małych projektów, z jasnymi i klarownymi wymaganiami,
- 5) możliwość jasnej komunikacji z klientami i odbiorcami, którzy mogą poznać spodziewane rezultaty, a także rozumieją koszt, rozmiar i harmonogram zlecanych przedsięwzięć,
- 6) solidną i wyczerpującą dokumentację, która pozwala ograniczyć wpływ rotacji pracowników na projekt.

Mimo późniejszego znacznego upowszechnienia modelu jego użytkownicy byli świadomi istniejących wad tego rozwiązania. Co ciekawe, W. Royce już we własnym ar-

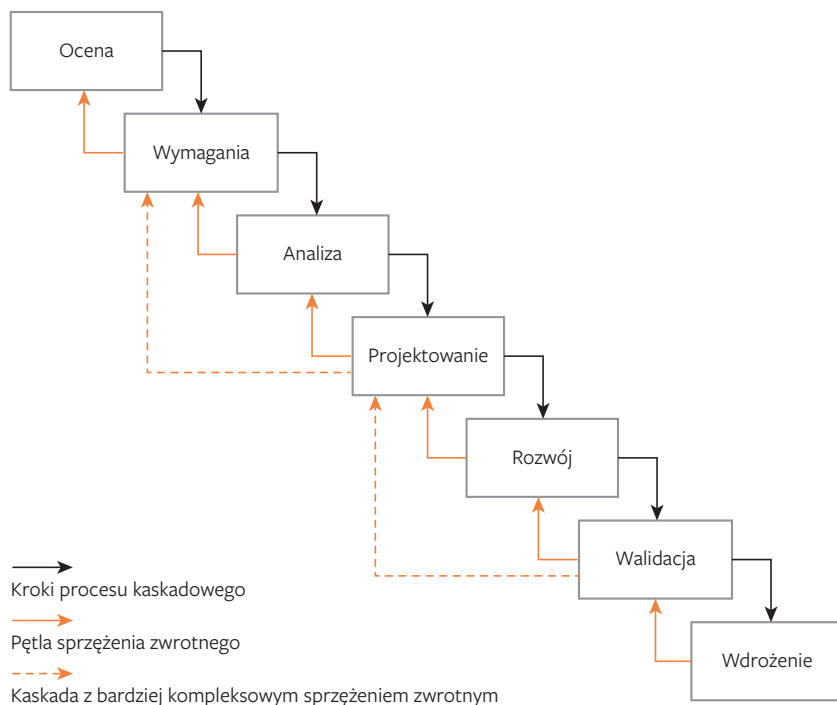
³ Standish Group, *The Chaos Report (1994)*, https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf (dostęp: 1.01.2021).

⁴ H.D. Benington, *Production of large computer programs*. In *Proceedings*, w: *ONR Symposium on Advanced Programming Methods for Digital Computers*, F.D. Rigby (ed.), Washington 1956, s. 15–27.

⁵ W.W. Royce, *Managing the development of large software systems: concepts and techniques*. In *Proceedings of the 9th international conference on Software Engineering (ICSE '87)*, IEEE Computer Society Press, Washington 1987, s. 328–338.

tykule zwracał uwagę na jego niedoskonałości. W szczególności dotyczyło to konieczności poprawy błędów wykrytych w etapie testowania⁶. W proponowanych modyfikacjach znaleźć można wprowadzenie iteracyjności i złamanie liniowości faz (patrz rysunek 1).

Rysunek 1. Kaskadowy model zarządzania projektem wg W. Royce’a



Źródło: N.B. Ruparelia, *Software development lifecycle models*, "ACM SIGSOFT Software Engineering Notes" 2010, vol. 35, no. 3, s. 8–13.

Bezpośrednim źródłem zainteresowania alternatywnymi podejściami stała się krytyka podstawowego paradygmatu podejścia kaskadowego, czyli szczegółowego, długookresowego planowania i zdolności do opisanie rezultatu końcowego już przy jego rozpoczęciu. Krytycy kontestowali celowość i wykonalność szczegółowego planowania produktów oraz przebiegu projektu, a także stosowania krępujących, ich zdaniem, norm czy standardów, wysuwając argumenty, które znalazły duży oddźwięk szczególnie wśród projektów o wysokim stopniu innowacyjności, w tym informatycznych i badawczo-rozwojowych. Głównym problemem, wyłączając typowe błędy w samym zarządzaniu projektem, wydawał się być zbyt duży poziom standaryzacji działań w projektach tego typu. Przejawia się to w podejmowaniu nieudanych prób ustalenia cyklu życia projektu, szczegółowych

⁶ W.W. Royce., *Managing the development...*

wymagań oraz projektowania na dużym poziomie szczegółowości już na wstępnych etapach. Takie podejście daje poczucie komfortu realizatorom i zamawiającym, bardzo często nie trafia jednak w rzeczywiste potrzeby. Wymagania klienta ewoluują, zwykle wraz z postępującą wizualizacją opracowywanych rozwiązań. Postęp prac nad projektem ujawnia konieczność wielokrotnej weryfikacji wielu założeń, działań i planów, ponieważ w miarę jego rozwoju zarówno zlecający, jak i zespół projektowy posiadają znacznie lepsze pojęcie o kształcie produktu finalnego. Często pewne ścieżki dochodzenia do określonych rezultatów okazują się nieefektywne dopiero po fazie testów i weryfikacji.

Dodatkowymi elementami podważającymi deterministyczny charakter przebiegu projektu są częste zmiany w jego otoczeniu: sytuacji rynkowej, uzasadnienia biznesowego, regulacji i norm prawnych oraz oczekiwań klienta. Ustalenie i konsekwentne realizowanie planów doprowadzają często do budowania funkcjonalności, w których duża część rozwiązań nie jest nigdy używana lub możliwa do użycia dopiero po kosztownych modyfikacjach. Według raportu Standish Group z 2013 r. z perspektywy wartości płynących z funkcjonalności systemów mamy do czynienia z realizacją zasady klasycznej zasady V. Pareto. Wykazano, bowiem iż:

- 1) 20% funkcjonalności jest wykorzystywanych często,
- 2) 50% wykorzystywanych jest sporadycznie lub bardzo rzadko
- 3) 30% czasami lub rzadko⁷.

Warto w tym zakresie przywołać artykuł S. Atkinsona i G. Benefielda, którzy bazując na badaniach Ala Goernera z Uniwersytetu Missouri w Kansas City, określili tzw. okres połowicznego rozkładu wymagań projektów informatycznych na okres 6 miesięcy⁸. Oznacza to, iż przy założeniu kompletności i wyczerpującego charakteru dokumentacji wymagań już po pół roku połowa z nich jest nieadekwatna do bieżącej sytuacji, zaś po roku dotyczy to trzech czwartych pierwotnego zakresu! Co więcej, czas ten (w stosunku do wcześniejszych dekad) raptownie maleje: w latach 80. XX w. było to 10–12 lat, zaś na przełomie wieków 2–3 lata.

Jako podsumowanie można przytoczyć tu wyniki badań i postawioną diagnozę opracowane przez C. Larmana⁹, który wskazał na następujące czynniki utrudniające lub wręcz uniemożliwiające precyzyjne określenie i opisanie rezultatów projektu na etapie jego definiowania:

- 1) klienci i użytkownicy nie są pewni, czego chcą,
- 2) klienci mają trudności ze sformułowaniem tego, czego chcą,

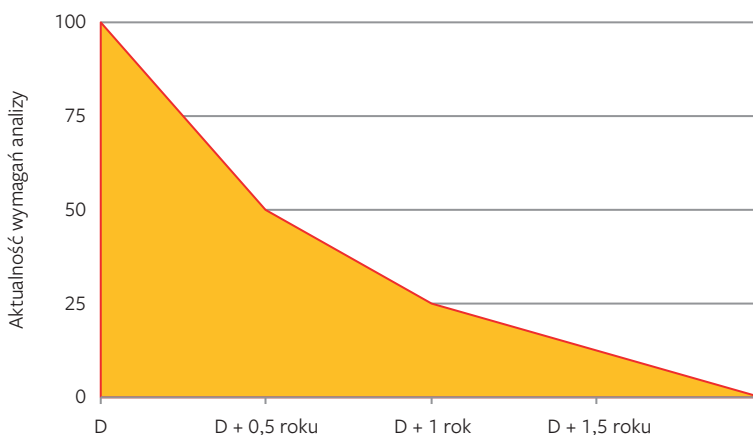
⁷ CHAOS Manifesto 2013. *Think Big, Act Small*, The Standish Group International, Incorporated, <http://versionone.com/assets/img/files/ChaosManifesto2013.pdf> (dostęp: 21.06.2020).

⁸ S. Atkinson, G. Benefield, *Software development: Why the traditional contract model is not fit for purpose*, „Proceedings of the Annual Hawaii International Conference on System Sciences”, w: *Proceedings of the Annual Hawaii International Conference on System Sciences*, Ralph H. Sprague, Jr. (ed.), Wailea 2013, s. 4842–4851.

⁹ C. Larman, *Agile and Iterative Development – A Manager's Guide*, Addison-Wesley, New Jersey 2003.

- 3) wiele szczegółów z tego, czego klienci naprawdę chcą, ujawni się dopiero podczas realizacji projektu,
- 4) szczegóły wdrożenia na samym początku realizacji nie dają się uporządkować,
- 5) wraz z tym, jak widzą powstający produkt, zmienia się sposób myślenia,
- 6) siły zewnętrzne (takie jak produkty konkurencji lub usługi) prowadzą do zmian lub rozbudowy wymagań.

Rysunek 2. Okres połowicznego rozkładu wymagań w projektach informatycznych (w %)



Źródło: S. Atkinson, G. Benefield, *Software development: Why the traditional contract model is not fit for purpose*, w: *Proceedings of the Annual Hawaii International Conference on System Sciences*, Ralph H. Sprague, Jr. (ed.), Wailea 2013, s. 4842–4851.

O ile zjawiska wymienione powyżej można uznać w dużym stopniu za uniwersalne, o tyle realizacja projektów informatycznych dodatkowo wiąże się z ryzykiem wystąpienia również innych nieprawidłowości. Philippe Kruchten wylicza kilka głównych problemów, które dotknąć mogą każdy projekt, a są to:

- 1) zarządzanie wymaganiami *ad hoc*,
- 2) niejednoznaczna i nieprecyzyjna komunikacja,
- 3) krucha architektura systemu,
- 4) przytłaczająca złożoność,
- 5) niewykryta niespójność w wymaganiach, projektach i implementacja,
- 6) niewystarczające testowanie,
- 7) subiektywna ocena statusu projektu,
- 8) niekontrolowane wprowadzenie zmian,
- 9) niewystarczająca automatyzacja¹⁰.

¹⁰ P. Kruchten, *The Rational Unified Process: An Introduction*, Addison-Wesley Pearson Education, New York 2003.

Obserwacja powyższych zjawisk oraz brak możliwości rozwiązania ich zgodnie z podejściem tradycyjnym przyczyniły się bezpośrednio do rozwoju prac nad uelastycznieniem i „odchudzeniem” tradycyjnych metodyk zarządzania projektami oraz poszukiwaniem alternatywnych podejść. Według N.B. Ruparelii można wśród nich wymienić takie, jak¹¹:

- 1) stworzony przez Birrell i Oulda B-Model łączący w sobie kaskadowe podejście do rozwoju produktu oraz cykliczne podczas jego użytkowania/utrzymania¹²,
- 2) opracowany przez NASA i zaprezentowany w 1991 r. V-Model¹³,
- 3) dążący do minimalizacji ryzyka projektu spiralny model rozwoju oprogramowania B. Boehma z 1986 r.¹⁴,
- 4) model piasta-szprycha (stanowiący rozwinięcie spiralnego), spotykany m.in. w metodzie TOGAF¹⁵,
- 5) iteracyjny model Rational Unified Process (RUP) opracowany przez Rational Software i rozwijany obecnie przez IBM¹⁶,
- 6) Rapid Application Development (RAD) – metodę opartą na prototypowaniu, opracowaną w 1991 r. przez J. Martina¹⁷.

Źródeł zwinnych metod zarządzania projektami według M. Trockiego poszukiwać można jeszcze wcześniej¹⁸. Sięga on bowiem aż do lat 30. XX w., do prac W. Shewarta, proponującego cykl Plan-Do-Study-Act (PDSA) do usprawniania procesów produkcyjnych czy późniejszego ciągłego doskonalenia PDCA W.E. Deminga, jak również do prac z lat 60. i 70. XX w. autorstwa T. Gilba. Ten ostatni autor to twórca ewolucyjnych procesów tworzenia oprogramowania o nazwie Evolutionary Project Management, w skrócie EVO, zaliczanych do najstarszych adaptacyjnych rozwiązań w tej dziedzinie¹⁹. Istotnym wkładem w rozwój podejść nieliniowych były badania H. Tekeuchiego i I. Noaki dotyczące liderów technologicznych połowy lat 80.: firm Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M, Xerox i Hewlett-Packard. Swoje wyniki opublikowali oni w 1986 r.,

¹¹ N.B. Ruparelia, *Software development lifecycle models*, „ACM SIGSOFT Software Engineering Notes” 2010, vol. 35, no. 3, s. 8–13.

¹² N.D. Birrell, M.A. Ould, *A practical handbook to software development*, Cambridge University Press, Cambridge 1988, s. 3–12.

¹³ K. Forsberg, H. Mooz, *The relationship of system engineering to the project cycle*, <http://www.damiantgordon.com/Courses/ISE/Papers/The%20Relationship%20of%20System%20Engineering%20to%20the%20Project%20Cycle.pdf> (dostęp: 12.04.2019).

¹⁴ B. Boehm, *A spiral model of software development and enhancement*, „ACM SigSoft Software Engineering Notes” 1986, vol. 2, no. 4, s. 22–42.

¹⁵ TOGAF 9: The Open Group, <http://www.opengroup.org/togaf/> (dostęp: 24.11.2021).

¹⁶ P. Kruchten, *The Rational Unified Process...*, s. 33.

¹⁷ J. Martin, *Rapid Application Development*, Macmillan, New York 1991, s. 81–90.

¹⁸ M. Trocki, *Dwadzieścia lat zwinnego zarządzania – doświadczenia i wyzwania*, w: *Zwinne zarządzanie projektami w dużych organizacjach*, P. Wyrozębski (red.), Oficyna Wydawnicza SGH, Warszawa 2020.

¹⁹ C. Larman, *Agile and Iterative Development...*, s. 211.

w artykule pod tytułem *The New New Development Game*, gdzie opisali założenia „nowych zasad gry”, używając po raz pierwszy słowa *scrum* do opisu nowego podejścia²⁰.

Tempo rozwoju adaptacyjnych metody zarządzania znacznie wzrosło w okresie intensywnego wzrostu w informatyce w latach 90. XX w. i w efekcie przygotowały grunt pod ogłoszony w 2001 r. Manifesto for Agile Software Development.

1.2. Agile Manifesto

Opracowanie i ogłoszenie Manifestu miało miejsce w dniach 11–13 lutego 2001 r., w miejscowości Snowbird w stanie Utah, gdzie odbyło się spotkanie przedstawicieli i propagatorów nowych, elastycznych, czy jak to wtedy mówiono: „lekkich”, metodyk zarządzania projektami. Udział wzięło w nim siedemnastu ekspertów²¹, późniejszych założycieli „sojuszu zwinności” – Agile Alliance. Byli to autorzy i promotorzy metod wytwarzania oprogramowania z powodzeniem rozwijanych od połowy lat 90. XX w., takich jak: Extreme Programming (XP), Cristal, Adaptive Software Development (ASD), Feature Driven Development (FDD) czy Dynamic Systems Development Method (DSDM).

W wyniku poszukiwań alternatywnych dla tradycyjnych, „ciężkich” metodyk podejść do realizacji projektów informatycznych oraz towarzyszących im dyskusji powstał manifest, który podsumowywał przesłanie, jakim mieli się oni odtąd kierować, wskazując, co jest lub co powinno być priorytetem w ich pracy. Opracowanie miało na celu zintegrowanie różnych metodyk odtąd nazywanych już w sposób konsekwentny „metodykami zwinnymi” wokół wspólnych wartości i przekonań. Manifest ten w całości brzmi następująco.

Tablica 1. Manifest programowania zwinnego

Odkrywamy nowe metody programowania dzięki praktyce w programowaniu i wspieraniu w nim innych.
W wyniku naszej pracy zaczęliśmy bardziej cenić:
Ludzi i interakcje od procesów i narzędzi.
Działające oprogramowanie od szczegółowej dokumentacji.
Współpracę z klientem od negocjacji umów.
Reagowanie na zmiany od realizacji założonego planu.
Oznacza to, że elementy wypisane po prawej są wartościowe, ale większą wartość mają dla nas te, które wypisano po lewej.

Źródło: <https://agilemanifesto.org/iso/pl/manifesto.html> (dostęp: 4.01.2021).

²⁰ H. Tekeuchi, I. Noaka, *The New New Product Development Game*, “Harvard Business Review”, Jan. – Feb. 1986.

²¹ Zgodnie z informacją zamieszczoną na stronie <https://agilemanifesto.org/> pierwszymi sygnatariuszami Manifestu byli: Kent Beck, Mike Beedle, Arie van Bennekum, Alistar Cockburn, Ward Cunningham, Martin Fowler, Robert C. Martin, Steve Mellor, Dave Thomas, James Grnning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Ken Schwaber, Jeff Sutherland.

Sygnatariusze manifestu uznali, że w odpowiedzi na wysoki stopień niepewności, występujący w dynamicznie zmieniającym się środowisku projektu, dotychczasowe priorytety powinny zostać przewartościowane. Zwrócili uwagę na fakt, iż to ludzie realizują projekty, a nie narzędzia i procesy, wobec czego należy przedłożyć ludzi, ich pracę i relacje ponad sztywny gorset zdefiniowanych narzędzi i procesów. W przypadku projektów innowacyjnych, aby uzyskać wysoki poziom kreatywności, należy pozostawić członkom zespołu możliwie dużo nieskrępowania i wolności. Autorzy krytykują, zbyt wysoki ich zdaniem, poziom biurokracji spowalniający zapał twórczy zespołów projektowych. Podkreślają, że tylko działające oprogramowanie może być miernikiem postępów projektu. Odrzucając rozbudowaną dokumentację, krytykują jej częste przekłamania i brak prawdziwie istotnych dla projektu informacji. Następnie przechodzą na poziom relacji z klientem, domagając się pełnego zaangażowania w pracę przy projekcie na każdym jego etapie. Ma to na celu lepszą komunikację, szybszą wymianę informacji i uzyskiwanie bieżących opinii na temat zmieniających się oczekiwań zleceniodawcy. Ostatnim postulatem sygnatariuszy manifestu jest konieczność adaptacyjnego budowania planów i umiejętnego reagowania na zmiany w projekcie w celu jak najlepszego spełnienia rzeczywistych, a nie deklarowanych wymagań klienta wobec produktu końcowego.

W ślad za opracowaną deklaracją autorzy, w uzupełnieniu jej krótkiej i zwięzłej formy, sformułowali szerszy zestaw 12 zasad (przykazań, pryncypiów) wspierających wyszczególnione w niej wartości. Ich kształt przedstawiono poniżej.

Tablica 2. Dwanaście zasad wspierających Manifest zwinnego programowania

1. Najwyższy priorytet ma dla nas zadowolenie klienta dzięki wczesnemu i ciągłemu wdrażaniu wartościowego oprogramowania.
2. Bądźcie gotowi na zmiany wymagań nawet na późnym etapie jego rozwoju. Procesy zwinne wykorzystują zmiany dla zapewnienia klientowi konkurencyjności.
3. Dostarczajcie funkcjonujące oprogramowanie często, w kilkutygodniowych lub kilkumiesięcznych odstępach. Im częściej, tym lepiej.
4. Zespoły biznesowe i deweloperskie muszą ściśle ze sobą współpracować w codziennej pracy przez cały czas trwania projektu.
5. Twórzcie projekty wokół zmotywowanych ludzi. Zapewnijcie im potrzebne środowisko oraz wsparcie i zaufajcie, że wykonają powierzone zadanie.
6. Najbardziej efektywnym i wydajnym sposobem przekazywania informacji zespołowi deweloperskiemu i wewnątrz niego jest rozmowa twarzą w twarz.
7. Działające oprogramowanie jest podstawową miarą postępu.
8. Procesy zwinne umożliwiają zrównoważony rozwój. Sponsorzy, deweloperzy oraz użytkownicy powinni być w stanie utrzymywać równe tempo pracy.
9. Ciągłe skupienie na technicznej doskonałości i dobrym projektowaniu zwiększa zwinność.
10. Prostota – sztuka minimalizowania ilości koniecznej pracy – ma kluczowe znaczenie.
11. Najlepsze rozwiązania architektoniczne, wymagania i projekty pochodzą od samoorganizujących się zespołów.
12. W regularnych odstępach czasu zespół analizuje możliwości poprawy swojej wydajności, a następnie dostraja i dostosowuje swoje działania do wyciągniętych wniosków.

Źródło: *ibid.*

Manifest spotkał się z życzliwą akceptacją w środowisku projektów informatycznych, jednakże w środowisku zarządzania projektami został przyjęty początkowo z rezerwą. Mimo to, stał się wspólnym sztaandarem i kamieniem milowym w dalszym rozwoju zwinnych metodyk zarządzania projektami.

1.3. Charakterystyka metod zwinnych

Zwinne metodyki zarządzania stanowią grupę metodyk charakteryzujących się „lekkim”, „zwinnym” podejściem do zarządzania projektami, realizującym postulaty opisane go wyżej manifestu. Od czasu ogłoszenia stał się on swoistym „parasolem”, pod którym zgromadzili się praktycy i promotorzy nowego podejścia rozwijający od tej pory poszczególne metodyki nazywane wspólnie zwinnymi (ang. *agile project methodologies*).

Do grupy zwinnych metodyk zarządzania projektami zalicza się m.in. następujące metodyki:

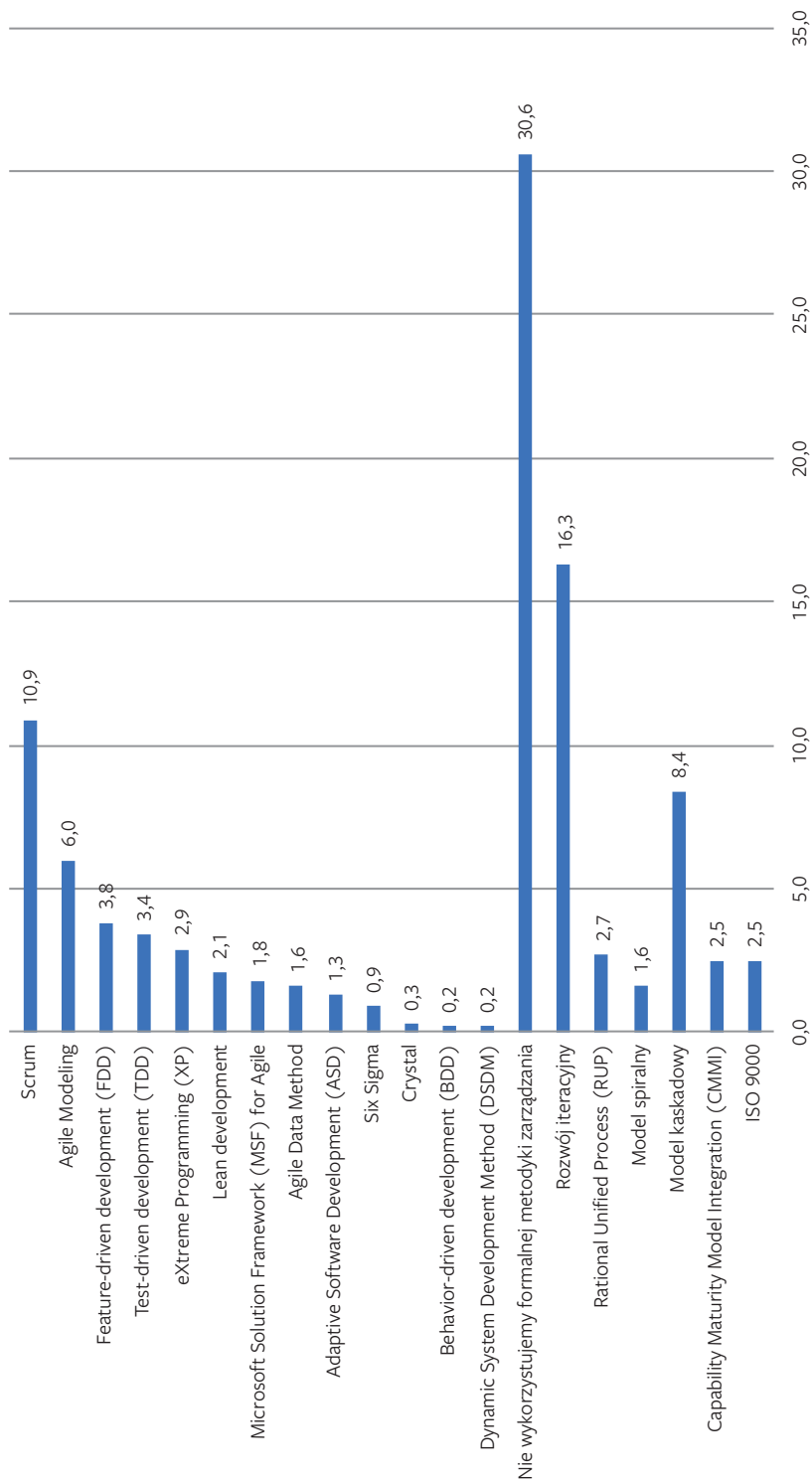
- 1) Adaptive Driven Development,
- 2) Crystal Clear Method,
- 3) Dynamic System Development Method (obecnie pod nazwą AgilePM)
- 4) Extreme Programming,
- 5) Feature Driven Development,
- 6) Lean Development,
- 7) SCRUM,
- 8) Prince2 Agile / XPrince.

W ostatnim czasie do tego grona zaczęto również dołączać metody takie jak: Lean, Lean Startup, Kanban, Design Thinking, a także i inne, które ze względu na zbieżność reprezentowanych wartości i praktyk pozostają zdaniem ich autorów i społeczności zarządzania projektami w zgodzie z ideami zwinnego zarządzania²².

Liczna rodzina metod zwinnych nie oznacza jednakowej popularności ww. rozwiązań w praktyce biznesowej. Dowodów na taką tezę dostarczają liczne badania dotyczące rozpowszechnienia i praktycznego znaczenia metod zwinnych w przedsiębiorstwach.

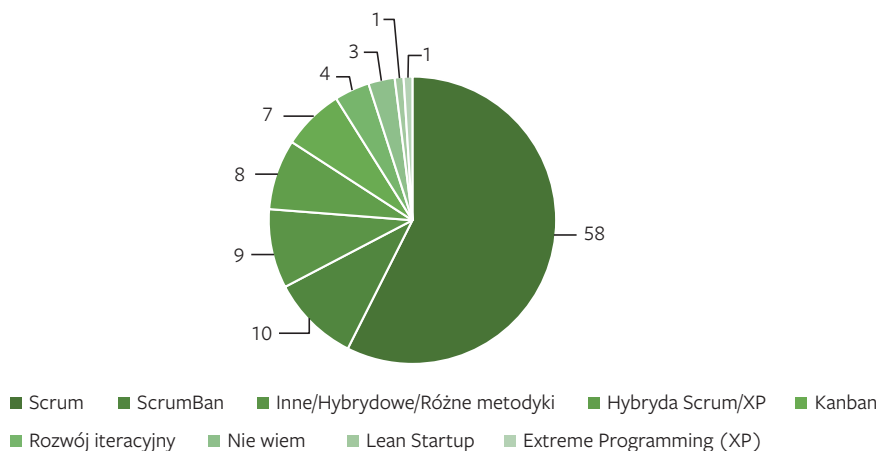
²² A. Komus et al., *Studie Status Quo Agile 2014*, https://www.hs-koblenz.de/fileadmin/media/fb_wirtschaftswissenschaften/Forschung_Projekte/Forschungsprojekte/Status_Quo_Agile/Studie_2014/2014.07.23_Bericht_Interessenten_final.v.1.01.pdf (dostęp: 24.11.2021).

Rysunek 3. Popularność zwinnych metod zarządzania projektami według „Forrester Research” (w %)



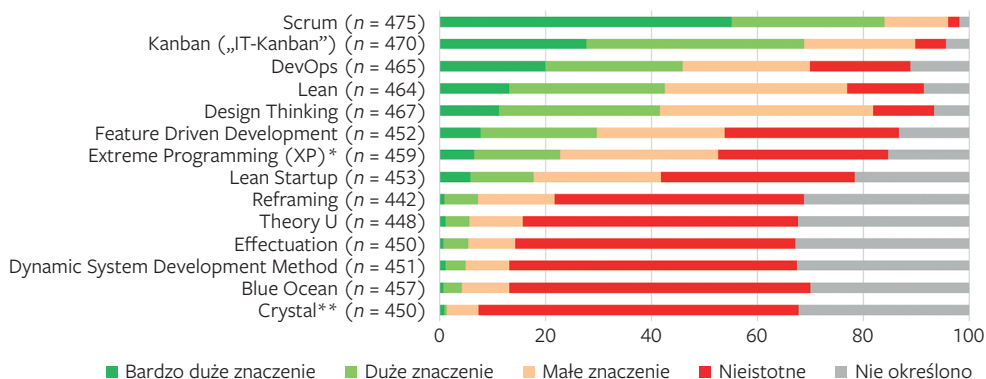
Źródło: D. West *et al.*, Agile development: Mainstream adoption has changed agility, „Forrester Research” 2020, vol. 2, no. 1.

Rysunek 4. Popularność zwinnych metodyk zarządzania projektami według VersionOne w 2020 r. (w %)



Źródło: VersionOne, 14th Annual State of Agile Report, <https://www.qagile.pl/wp-content/uploads/2020/06/14th-annual-state-of-agile-report.pdf> (dostęp: 24.11.2021).

Rysunek 5. Popularność zwinnych metodyk zarządzania projektami według badania prof. A. Komusa „Status Quo Agile” 2020 (w %)



* Włączając w to test driven development i programowanie w parach.

** Kolejność metod według liczby odpowiedzi „duże znaczenie” i „bardzo duże znaczenie”.

Źródło: A. Komus et al., Studie Status Quo Agile 2014, https://www.hs-koblenz.de/fileadmin/media/fb_wirtschaftswissenschaften/Forschung_Projekte/Forschungsprojekte/Status_Quo_Agile/Studie_2014/2014.07.23_Bericht_Interessenten_final.v.1.01.pdf (dostęp: 24.11.2021).

Z perspektywy przeszło dwóch dekad od ogłoszenia Manifestu zdecydowanie na czele jawi się metodyka SCRUM, konsekwentnie deklarowana jako najczęściej stosowana metodyka zwinna. W szczególności zdaje się to być zasługą cech merytorycznych, takich jak: koncentracja na zespole projektowym, organizacja pracy, zarządzanie w oparte na priorytetach, troska o jakość, jak i cech „technicznych”: prostoty opisu metody, zwiezłości,

niezależności od warstwy narzędziowej, łatwości przyswojenia²³. Oryginalna metoda SCRUM często łączona jest z techniką wizualizacji przepływu pracy Kanban wywodzącą się z japońskiej szkoły zarządzania produkcją. Taki miks nazywany jest fachowo IT-Kanban lub ScumBan i również cechuje się dużą popularnością. W zestawieniach odnaleźć można także metody, takie jak FDD, XP, TDD, czyli o bardziej technicznym i *stricte* programistycznym charakterze niż SCRUM. Z kolei Lean ceniony jest za efektywność i jakość rezultatów, zaś Design Thinking za zdolność tworzenia innowacyjnych rozwiązań²⁴.

Szczegółowe omówienie wiodących metodyk zwinnych zaprezentowane zostanie w osobnej części niniejszej publikacji.

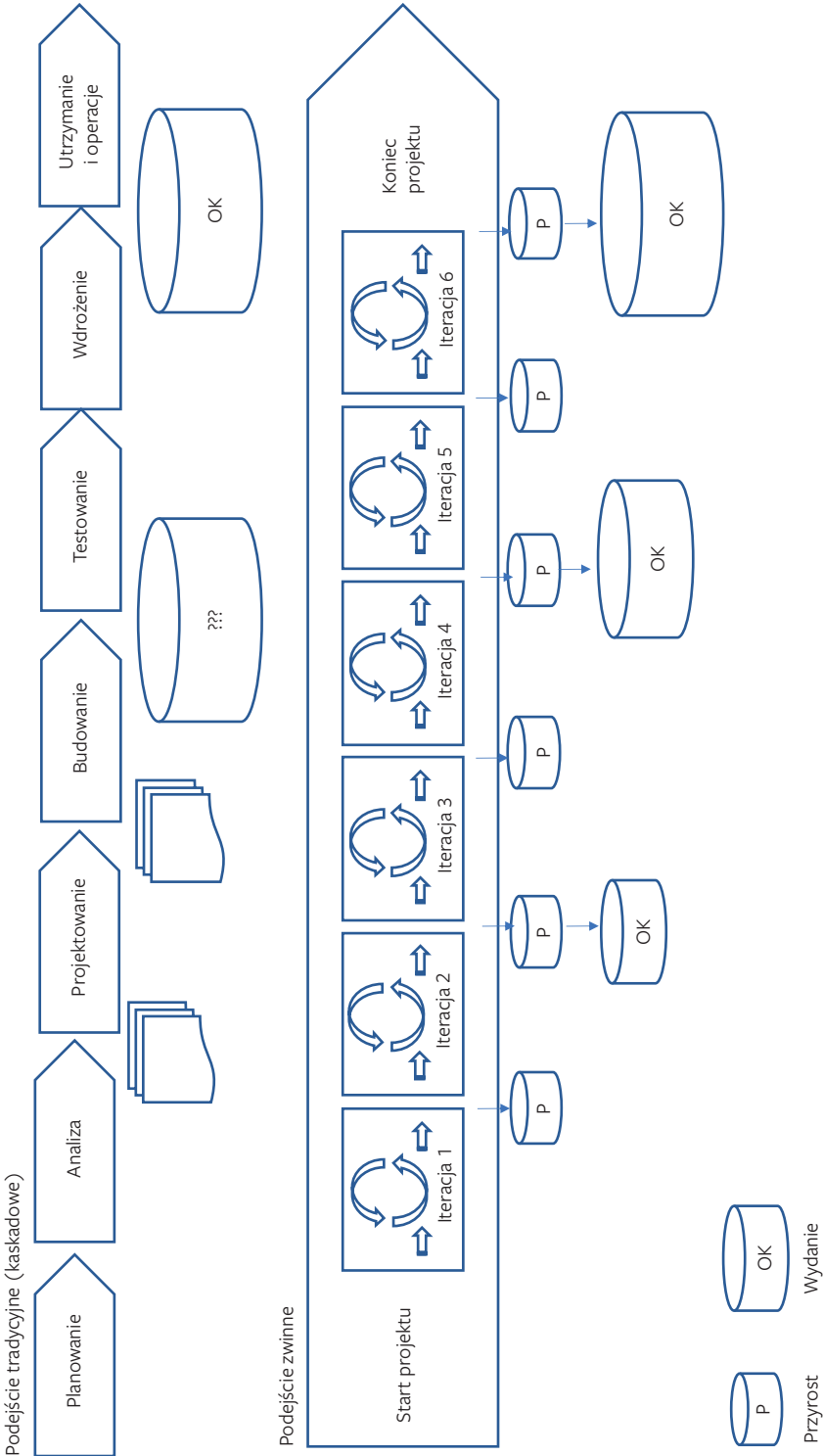
Tak duże zróżnicowanie metod składania do pytania o łączące je cechy. Pierwszą i oczywistą będzie rzecz jasna deklaracja autorów metod względem zgodności ich zaleceń z Manifestem i 12 zasadami zwinności. Wśród innych, wspólnych charakterystyk podejścia zwinnego można wyróżnić między innymi:

- 1) cykl życia projektu składający się z adaptacyjnych iteracji, zwykle o tym samym czasie trwania (ang. *time-boxing/sprint/iterations*),
- 2) krótkie (dzienne, tygodniowe, miesięczne) okresy adaptacyjnych iteracji,
- 3) planowanie zakresu projektu oparte na identyfikacji funkcjonalności produktu,
- 4) stosowanie spriorytetyzowanego dokumentu przedstawiającego zakres prac do wykonania w ramach poszczególnych iteracji (tzw. backlog),
- 5) stosowanie planowania iteracyjnego – deklaracja pracy do wykonania na podstawie elementów backloga przypisywanych do danej iteracji,
- 6) wymagania, projektowanie i sam produkt powstaje w miarę ewoluowania projektu,
- 7) regularnie dostarczane przyrosty (ang. *increment*) działającego oprogramowania,
- 8) troska o testowanie i najwyższą jakość przekazywanych przyrostów,
- 9) przedstawiciele klienta, użytkownicy pozostają stale zaangażowani w projekt; m. in. poprzez uczestniczenie w okresowych i końcowych demonstracjach produktu; przedstawiciel klienta współpracuje blisko z zespołem i dostarcza mu wiedzy odnośnie do pożądanych kierunków rozwoju produktu i istniejących ograniczeniach projektu,
- 10) realizacja projektu przez małe (do 10 osób), samoorganizujące się zespoły; w przypadku dużych projektów stosowane struktury gronowej (wiele małych zespołów),
- 11) codzienna synchronizacja pracy dzięki krótkim spotkaniom zespołu (tzw. *stand-ups*),
- 12) praca w warunkach jednozadaniowości – każdy członek zespołu pracuje tylko nad jednym projektem, w danym czasie,
- 13) zespoły projektowe łączą osoby kompetencji uniwersalnych oraz specjalistów; do zespołu, w miarę potrzeb zapraszani są eksperci funkcjonalni,

²³ *Ibid.*

²⁴ *Ibid.*

Rysunek 6. Porównanie cyklu życia projektu (modelu wytwarzania) według podejścia tradycyjnego (liniowa kaskada) oraz podejścia zwinnego (iteracyjnego i przyrostowego)



Źródło: The California Department of Technology Project Management Office (CA-PMO) <https://projectresources.cdt.ca.gov/agile/waterfall-and-agile/> (dostęp: 24.11.2021).

14) uwzględnienie uczenia się dzięki regularnym spotkaniom podsumowującym (tzw. retrospektywy).

Nowe podejście i proponowane w związku z nim zmiany stanowiły istotną nowość w stosunku do dotychczasowo praktykowanego podejścia kaskadowego. Szczegółowe zestawienie różnic przedstawiono w tabeli 1.

Tabela 1. Tradycyjna i zwinna perspektywa na projekty wytwarzania oprogramowania

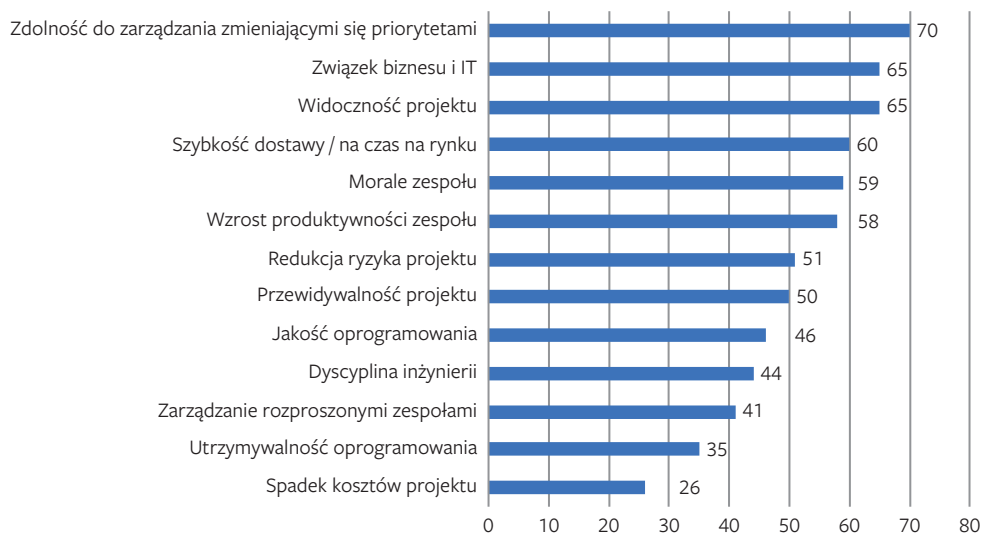
Zagadnienia	Podejście tradycyjne	Podejście zwinne
Cykl wytwarzania produktu (Charvat, 2003; Nerur, Mahapatra, Mangalaraj, 2005)	liniowy (kaskadowy, spiralny lub ich odmiana)	iteracyjny, ewolucyjny model budowania produktów
Styl rozwoju (Leffingwell, 2007)	presytuacyjny/predyktywny	sytuacyjny/adaptacyjny
Wymagania (Boehm, 2002; Boehm, Turner, 2004)	rozpoznawalne wcześniej, w dużej mierze stabilne; jasno określone i udokumentowane	wyłaniające się, szybka zmiana, nieznanne – odkrywane podczas projektu
Architektura (Boehm, 2002; Wysocki, 2009, 2011)	ciężka architektura dla obecnych i przyszłych wymagań	zasada YAGNI „Nie będzie ci to potrzebne!” (“You aren’t going to need it”)
Zarządzanie (Boehm, Turner, 2005; Vinekar, Slinkman, Nerur, 2006)	zorientowane na proces; zarządzanie i kontrola	zorientowane na ludzi; przywództwo i współpraca
Dokumentacja (Boehm, Turner, 2005)	ciężka/szczegółowa wiedza jawna	lekka (zastąpiona przez komunikację twarzą w twarz); wiedza ukryta
Cel (Dybå, Dingsøy, 2009)	przewidywalność i optymalizacja	eksploracja i adaptacja
Zmiana (Boehm, Turner, 2003)	tendencja do niechęci do zmian	akceptacja zmian
Zespół (Boehm, 2002; Sherehiy, Karwowski, Layer, 2007)	rozproszone zespoły specjalistów; zorientowany na plan, adekwatny dostęp umiejętności do wiedzy zewnętrznej	zwinny, kompetentny, skołokowany i oparty na współpracy; kolokacja starszego personelu technicznego
Organizacja zespołu (Leffingwell, 2007)	wstępnie ustrukturyzowane zespoły	samoorganizujące się zespoły; klienci
Zaangażowanie klienta (Highsmith, Cockburn, 2001)	małe zaangażowanie, pasywne	klient na miejscu i uważany za członka zespołu; aktywny/proaktywny
Kultura organizacyjna (Highsmith, 2002; Nerur, Mahapatra, Mangalaraj, 2005)	kultura dowodzenia i kontroli	kultura przywództwa i współpracy
Proces rozwoju oprogramowania (Salo, Abrahamsson, 2007)	uniwersalne podejście i rozwiązanie zapewniające przewidywalność i wysoką pewność	elastyczne podejście dostosowane do zbiorowego zrozumienia potrzeb kontekstowych w celu zapewnienia szybszego rozwoju
Miary sukcesu (Highsmith, 2010)	zgodność z planem	dostarczana wartość biznesowa

Źródło: A. B. M Moniruzzaman, S. Akhter Hossain, *Comparative Study on Agile Software Development Methodologies*, “Global Journal of Computer Science and Technology” 2013, vol. 13., no. 7.

1.4. Korzyści ze stosowania metod zwinnych

Przyjęcie powyżej wyszczególnionych zasad organizacji pracy przyniosło znaczną poprawę efektywności, jakości oraz morale zespołów projektowych oraz w dalszej konsekwencji przełożyło się na pozytywne efekty dla przedsiębiorstw jako całości. W literaturze naukowej oraz raportach dostępne są liczne zestawienia prezentujące korzyści osiągnięte z wdrożenia zwinnych metod do praktyki działania przedsiębiorstw. Wśród nich warto wskazać na cykliczne raporty firmy VersionOne: *State of Agile Report*, jak również raporty Project Management Institute oraz badania prof. A. Komusa *Status Quo Agile*. W niemal wszystkich źródłach autorzy pozostają zgodni, iż kluczową korzyścią ze zwinności pozostaje szybkość i zdolność do adaptacji. Są to charakterystyki kluczowe i pozwalające organizacjom budować trwałe przewagi konkurencyjne konkurując czasem zbudowania i dostarczenia rozwiązania oraz elastycznością w działaniu.

Rysunek 7. Korzyści wdrożenia zwinnego zarządzania według VersionOne (w %)



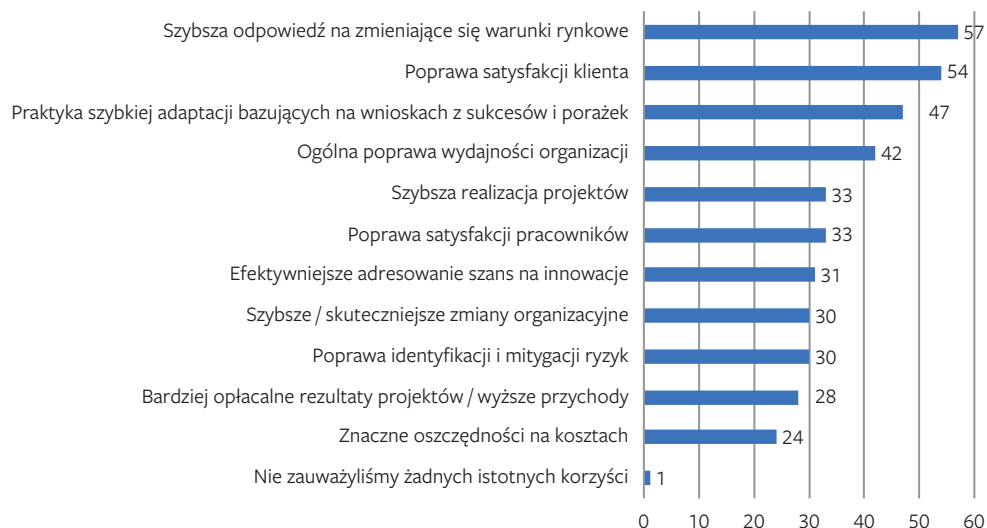
Źródło: VersionOne, 14th annual...

W badaniach PMI wykazano dodatkowo, iż współpraca, komunikacja oraz elastyczność uzyskiwane dzięki stosowaniu metod zwinnych przekładają się bezpośrednio na osiągnięte wyniki projektów²⁵. Organizacje o wysokim poziomie tychże cech osiągały w stosunku do pozostałych średnio o 30 punktów procentowych większy odsetek projektów

²⁵ Project Management Institute, *Capturing the value of project management*, <https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2015.pdf> (dostęp: 24.11.2021).

realizujących stawiane im cele biznesowe, mieszczących się w budżecie, realizowanych na czas i dających oczekiwany zwrot z inwestycji (ROI)²⁶.

Rysunek 8. Najważniejsze korzyści (czynniki sukcesu) wzrostu organizacyjnej zwinności (w %)



Źródło: Project Management Institute, *Capturing the value of project management*, <https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2015.pdf> (dostęp: 24.11.2021).

Tabela 2. Wskaźniki sukcesu projektów programistycznych (w %) wg bazy Standish Group Chaos Repo (FY2011–2015, $n > 10\,000$)

Rozmiar	Podejście	Sukces (<i>successful</i>)	Zagrożony (<i>challenged</i>)	Porażka (<i>failed</i>)
Projekty wszystkich rozmiarów	zwinne	39	52	9
	kaskadowe	11	60	29
Projekty duże	zwinne	18	59	23
	kaskadowe	3	55	42
Projekty średnie	zwinne	27	62	11
	kaskadowe	7	68	25
Projekty małe	zwinne	58	38	4
	kaskadowe	44	45	11

Źródło: Standish Group, *CHAOS Report 2015*, https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf (dostęp: 3.02.2021).

²⁶ *Ibid.* s. 10–13.

Interesujące wyniki prezentują również raporty Standish Group i analiza zgromadzonych w bazie danych na przestrzeni kilku lat ponad 10 tys. projektów²⁷.

Opracowane zestawienie ilustruje wskaźnik sukcesu w zależności od przyjętej metody pracy oraz ich rozmiaru. Z perspektywy całej próby odsetek projektów zakończonych sukcesem, w przypadku metod zwinnych był niemal czterokrotnie większy niż w metodach tradycyjnych (39% vs 11%). Analogicznie, odwrotna relacja dotyczyła projektów zakończonych porażką: 9% vs 29%. Różnice te rysowały się nawet wyraźniej, gdy wziąć pod uwagę rozmiary projektów. O ile generalnie sukces dużych projektów był rzadszy, to już stosowanie podejścia zwinnego podnosiło w stosunku do kaskady szanse sukcesu aż sześciokrotnie (18% vs 3%), przy czym stosunek ten spadał na korzyść podejść tradycyjnych wraz z zmniejszaniem się rozmiaru przedsięwzięcia. W każdej kategorii rozmiarowej było też dwukrotnie więcej tradycyjnie realizowanych projektów zakończonych porażką niż tych zarządzanych zwinnie.

Według autorów raportu kluczowe wnioski to: po pierwsze, wbrew pozorom, podejścia kaskadowe bardzo trudno skalować na większe przedsięwzięcia i *agile* jest w tym zakresie skuteczniejsze, po drugie, powinno dążyć się do organizowania mniejszych, łatwiej zarządzalnych projektów, które mają większe szanse powodzenia²⁸. Istotnym w tym temacie jest cykliczne, powtarzane od 2012 r. badanie prowadzone przez zespół pod kierunkiem prof. A. Komusa z Uniwersytetu Nauk Stosowanych w Koblencji²⁹. W najnowszej edycji raportu pod nazwą *Status Quo Agile* rysuje on całościowy obraz funkcjonowania zwinności w badanych organizacjach. Główną osią badania jest teza, iż stosowanie *agile* nie jest „zero-jedynkowe”, ale występuje w różnych odmianach. O ile 20% badanych organizacji stosuje konsekwentnie podejście zwinne, o tyle większość użytkowników używa tych metod wybiórczo lub w formie mieszanej (43% hybrydowo, 28% selektywnie)³⁰. Z perspektywy osiąganego korzyści zdecydowana większość (85%) respondentów potwierdziła jednak, iż w efekcie stosowania metod zwinnych nastąpiła poprawa wyników i wydajności projektów. Co więcej, spośród tej grupy niemal wszyscy (89%) poświadczali, iż efekty są dużo wyższe (19%) lub wyższe (70%) niż poczynione nakłady w związku ze zmianą modelu pracy. Szczegółowe zagadnienia tzw. zwinnych transformacji omówione zostaną w dalszych rozdziałach publikacji.

²⁷ Standish Group, *CHAOS Report 2015*, https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf (dostęp: 24.11.2021).

²⁸ *Ibid.*

²⁹ A. Komus et al., *Study Status Quo...* (dostęp: 14.06.2021).

³⁰ A. Komus, M. Kuberg, *Result Report: Status Quo (Scaled) Agile 2019/20*, <https://www.process-and-project.net/studien/studien-anfordern/report-status-quo-scaled-agile-2020-en-request/> (dostęp: 9.06.2021).

1.5. Istota i próba zdefiniowania pojęcia zwinności

Rozważając tak dynamicznie rozwijającą się tematykę zwinnego zarządzania projektami nie sposób pominąć zdefiniowania przedmiotu i znaczenia centralnego pojęcia jakim jest *agile*. Słowo ma swoje źródła w łacińskim czasowniku *agere* oznaczającym „działać, czynić”³¹ i po raz pierwszy pojawiło się w języku angielskim w około 1580 r. jako zaczerpnięte z języka francuskiego³². Aspekt biznesowy nadał mu opublikowany w 2001 r. *Manifesto for Agile Software Development*³³. Tłumacząc tytuł na język polski, można przyjąć go jako Manifest zwinnego wytwarzania oprogramowania, natomiast oficjalna polska wersja zamieszczona na rzeczonyj stronie internetowej posługuje się skróconą nazwą *Manifest programowania zwinnego*³⁴.

Z językowego punktu widzenia angielskie słowo *agile* jest przymiotnikiem, a więc cechą, charakterystyką przypisywaną rzeczownikom (osobom, przedmiotom, zjawiskom itd.). Internetowy słownik *Cambridge Dictionary* podaje polskie znaczenie tego słowa jako: „zwinny” (tj. „zdolny do poruszania swoim ciałem łatwo i szybko”) oraz „sprawny” (tj. „zdolny do szybkiego myślenia w inteligentny sposób”)³⁵. Takie rozumienie wydaje się zgodne z intencjami sygnatariuszy Manifestu piszących o atrybutach osób przedkładających pewne cechy działania ponad inne. Takie rozumienie tego słowa (właściwość ruchu oraz umysłu) występuje również w *Internetowym Słowniku MacMillan*. Należy jednak zaznaczyć, iż słownik ten dodaje jeszcze trzecią definicję pojęcia, wprost odnoszącą się do rozumienia go z perspektywy zarządzania: „w biznesie, odnoszący się do sposobu zarządzania projektami opartego na podziale pracy na szereg małych zadań i wykorzystaniu regularnej informacji zwrotnej na każdym etapie w celu wprowadzenia zmian w sposobie wykonywania pracy na późniejszych etapach”³⁶. Ta trzecia definicja pokazuje zmianę znaczenia słowa wobec postępu metod zarządzania projektami, ale dowodzi też zmiany jego paradygmatu fleksyjnego z przymiotnika na rzeczownik: konkretny sposób postępowania stosowany w dziedzinie zarządzania projektami. W miarę rozwoju dziedziny i popularyzacji metod zwinnego programowania *agile* stało się odrębną dziedziną wraz z własną terminologią, pryncypiami, metodami i narzędziami pracy oraz organizacjami wspierającymi jej rozwój. „Zwinne” zaczęło być utożsamiane ze „zwinnością”, a *agile* stało się niejako odrębnym bytem. W mowie potocznej pojawiły się określenia typu *doing agile* – „robić agile” kontrastowane z pojęciem *being agile* – „być zwinnym”³⁷. Mimo pozornej trywialności fakt, czy

³¹ <https://pl.wiktionary.org/wiki/agere> (dostęp: 15.06.2021).

³² <https://www.macmillandictionaryblog.com/agile> (dostęp: 15.06.2021).

³³ <http://agilemanifesto.org/> (dostęp: 24.11.2021).

³⁴ <http://agilemanifesto.org/iso/pl/manifesto.html> (dostęp: 24.11.2021).

³⁵ <https://dictionary.cambridge.org/pl/dictionary/english-polish/agile> (dostęp: 15.06.2021).

³⁶ <https://www.macmillandictionaryblog.com/agile> (dostęp: 15.06.2021).

³⁷ I. Marinova, *Doing Agile vs. Being Agile. Why it matters?*, <https://kanbanize.com/blog/doing-agile-vs-being-agile/> (dostęp: 15.06.2021).

agile jest przymiotnikiem, czy rzeczownikiem jest przedmiotem dyskusji praktyków oraz specjalistów wielu dziedzin, gdyż stanowi o istocie tego pojęcia³⁸. Czy zwinność to często złożony system działania organizacji ze wszystkim jego elementami? Czy cecha zespołów, wartości, kultura pracy bliższa pojęciu „zwinnego myślenia” (ang. *agile mindset*). W ostatnich latach, w szczególności w świetle coraz bardziej skomplikowanych metod skalowania zwinności oraz tzw. zwinnych transformacji, podnoszone są pytania o istotę dziedziny oraz formułowane postulaty powrotu do korzeni³⁹. Piszący te słowa nie podejmuje się jednoznacznego rozstrzygnięcia owego dylematu. W niniejszej monografii, ze względu na chęć kompleksowego ujęcia tematyki, stosowane będą obydwa rozumienia tego słowa.

Jak jednak w takim razie rozumieć bycie zwinnym i zwinność jako taką? W ślad za M. Trockim „zwinność” możemy traktować jako pojęcie wieloznaczne⁴⁰. Za *Słownikiem Języka Polskiego PWN* „zwinny” to⁴¹: 1. «wykonujący szybkie, zręczne ruchy», 2. «o ruchach, poruszaniu się kogoś lub czegoś: szybki i zgrabny». *Wielki słownik języka polskiego* opracowany w Instytucie Języka Polskiego PAN dodaje do tej definicji jeszcze, iż jest to określenie człowieka „takiego, którego ruchy są szybkie, dokładne i lekkie”⁴².

Zwinność będzie zatem cechą organizmów żywych i charakterystyką właściwą jednostkom. Cechę tę, przypisywaną wykonawcy działania, definiuje już T. Kotarbiński w *Traktacie o dobrej robocie*, porównując „zwinność” do „zręczności”⁴³ i pisząc, iż „Zręczniej [czyt. zwinniej – przyp. aut.] porusza się ten, kto porusza się dokładniej wedle zamierzenia, robi to z mniejszym wysiłkiem i mięśni i uwagi, z większym poczuciem łatwości, szybciej o ile zechce, i jeśli wcześniejsze fazy jego ruchu jakoś gładziej przechodzą w fazy późniejsze”⁴⁴. Indywidualne postrzeganie zwinności użyteczne jest z perspektywy prakseologicznej, jednak nie stanowi istoty rozpatrywanego podejścia zwinnego.

Kluczowym jest tutaj podejście do zwinności na poziomie zespołowym. Według sze-rokich badań literaturowych i lingwistycznych przeprowadzonych przez E.C. Conforto na tym poziomie zwinność oznaczać będzie „zdolność **zespołu projektowego do szybkiej zmiany planu projektu w odpowiedzi na potrzeby klientów**, interesariuszy, rynku lub

³⁸ C. Worrall, <https://www.bcsconsulting.com/blog/agile-its-an-adjective-not-a-noun/>; D. Dame, <https://www.linkedin.com/pulse/agile-verb-noun-david-dame/>; A. Cleff, <https://www.andycleff.com/2016/09/agile-is-not-a-noun/> (dostęp: 17.06.2021).

³⁹ K. Schwaber, *unSAFE at any speed*, <https://kenschwaber.wordpress.com/2013/08/06/unsafe-at-any-speed/> (dostęp: 24.11.2021); R. Freedman, *Back to the Roots: Where Agile Came From*, <https://dzone.com/articles/back-to-the-roots-where-agile-came-from> (dostęp: 24.11.2021); S. Williams, *Getting Back to the Roots of Agile*, <https://www.pragmaticinstitute.com/resources/articles/product/getting-back-to-the-roots-of-agile/> (dostęp: 5.01.2021).

⁴⁰ M. Trocki, *Dwadzieścia lat zwinnego zarządzania...*, s. 14–15.

⁴¹ Słownik Języka Polskiego PWN, hasło *zwinny*, <https://sjp.pwn.pl/sjp/zwinny;2547753.html> (dostęp: 21.06.2021).

⁴² Wielki Słownik Języka Polskiego, hasło: *zwinny*, https://wsjp.pl/index.php?id_hasla=39341&id_znaczenia=4230797&l=29&ind=0 (dostęp: 21.06.2021).

⁴³ T. Kotarbiński, *Traktat o dobrej robocie*, Wydawnictwo Uniwersytetu Łódzkiego, Łódź 2019, s. 131.

⁴⁴ *Ibid.*, s. 404.

technologii w celu **poprawy wyników projektów i dostarczania produktów** w innowacyjnym i dynamicznym środowisku”⁴⁵. Zwinność zatem jest cechą zespołów, które taką zdolność do szybkiej reakcji w zmiennym otoczeniu wykształciły samodzielnie lub na- były przy pomocy z zewnątrz. Ponownie dostrzega się, iż „zwinność jest umiejętnością/ zdolnością bardziej niż atrybutem metody lub praktyki”⁴⁶. M. Trocki dodaje, iż „cechami takiej zwinności są między innymi: ścisła i partnerska współpraca stron, iteracyjny i przy- rostowy sposób dostarczania rezultatów oraz otwartość na zmiany, poprzez elastyczną adaptację do zmieniających się okoliczności”⁴⁷.

Tabela 3. Interpretacje pojęcia zwinności

Zwinność w potocznym/ słownikowym rozumieniu	Zwinność jednostkowa	Zwinność zespołowa	Zwinność organizacyjna/ biznesowa
Zwinny – „mający szybkie, zręczne ruchy”: <i>Z. jak małpa, Z. palce</i> (MSJP). Także zwinność jako: ▪ wdzięk w poruszaniu się ▪ delikatność w wykonaniu czegoś	Zwinność jako: ▪ swoboda w wykonywaniu czegoś ▪ efektywne wykonywanie czegoś ▪ sprawne wykonywanie czynności	Zwinność jako: „cecha pracy zespołowej realizowanej w warunkach zmieniających się wymagań użytkowników lub klientów i asymetrii informacji nakierowanej na maksymalizację dostarczanej im wartości”. Cechami takiej zwinności są między innymi: ▪ ścisła i partnerska współpraca stron ▪ iteracyjny i przyrostowy sposób dostarczania rezultatów oraz ▪ otwartość na zmiany, poprzez elastyczną adaptację do zmieniających się okoliczności	Zwinność jako: ▪ „zdolność organizacji do uzyskania przewagi konkurencyjnej poprzez inteligentne, szybkie i proaktywne określenie szans oraz reagowanie na zagrożenia” (Bessant et al., 1999) ▪ „zdolność organizacji firmy (projektu, firmy, działu) do elastycznego i szybkiego dostosowywania się do zmieniających się czynników w trakcie jej procesów – nie tracąc z oczu nadrzędnego celu (C. Husselmann)

Źródło: M. Trocki, *Dwadzieścia lat zwinnego zarządzania – doświadczenia i wyzwania*, w: *Zwinne zarządzanie projektami w dużych organizacjach*, P. Wyrozębski (red.), Oficyna Wydawnicza SGH, Warszawa 2020.

Takie rozumienie zwinności towarzyszyło twórcom koncepcji u jej źródeł i jest rozu- mieniem pierwotnym dla niej centralnym, fundamentalnym całej dziedziny zwinnego zarządzania. Stanowi ono punkt wyjścia do formułowania wszelkich innych zaleceń i jed- nocześnie kryterium oceny proponowanych rozwiązań. Taka teza znajduje potwierdzenie nie tylko w literaturze naukowej, ale także w praktyce działania organizacji. Najlepszym tego przykładem zdaje się być dominacja metody SCRUM, orientującej się na zespół, oraz

⁴⁵ E.C. Conforto et al., *The agility construct on project management theory*, “International Journal of Project Management” 2016, vol. 34, no. 4, s. 660–674.

⁴⁶ *Ibid.*

⁴⁷ M. Trocki, *Dwadzieścia lat zwinnego zarządzania...*

powszechne niemal traktowanie jej jako podstawy dla bardziej złożonych rozwiązań zarządczych (np. model Spotify, SAFe, Nexus itp.).

Przyjęcie, iż tak jak planety wokół Słońca, wszystko kręci się wokół zwinnych zespołów, stawia pytanie o możliwość i warunki powodzenia przeniesienia cechy z niewielkich i zintegrowanych grup ludzi na duże, zinstytucjonalizowane byty, jakim są przedsiębiorstwa. Pytanie to wprost nawiązuje bowiem do trzeciego rozumienia zwinności, jakim jest zwinność organizacyjna lub biznesowa. Sukces zwinnych zespołów przyczynił się chęci przypisania cechy zwinności całym organizacjom. Już w 2004 r. według raportu Global CEO Study 90% dyrektorów naczelnych (ang. *chief executive officers*, CEO) deklarowało dążenie do transformowania organizacji na bardziej responsywną w ciągu najbliższych lat⁴⁸. W raporcie czytamy: „CEO na całym świecie zidentyfikowali responsywność, zwinność i elastyczność organizacji jako niezbędne kompetencje. Rozwijanie zdolności organizacji do nie tylko wyczuwania, ale także przewidywania i reagowania na zmieniające się rynki i wymagania klientów to jedno z największych wyzwań dla dzisiejszych CEO”⁴⁹. „Zwinne przedsiębiorstwo” stało się od tej pory nowym paradygmatem biznesowym⁵⁰, bytem niemalże idealnym⁵¹: „zdolnym do uzyskania przewag konkurencyjnych poprzez inteligentne, szybkie i proaktywne określenie szans i reagowanie na zagrożenia”⁵², a w efekcie uzyskującym ponadprzeciętne wyniki w kategorii szybkości, kosztów, jakości i spełniania potrzeb klientów⁵³. Zwinność organizacyjna to „zdolność firmy do dynamicznej modyfikacji i/lub rekonfiguracji poszczególnych procesów biznesowych w celu dostosowania do wymaganych i potencjalnych potrzeb firmy”⁵⁴. H. Sharifi i Z. Zhang formułują cechy zwinnego przedsiębiorstwa takie jak: zdolność reagowania, adaptacyjność elastyczność oraz konkurencyjność⁵⁵. Według G. Bełza i A. Baranasza dodatkowo są to m.in.:

- 1) umiejętność szybkiej identyfikacji szans rynkowych,
- 2) właściwa interpretacja zagrożeń wynikających z otoczenia biznesowego,
- 3) umiejętność kategoryzacji sytuacji na korzystne lub niekorzystne,

⁴⁸ IBM Business Consulting Services, *Companion guide to Your Turn: The Global CEO Study 2004*, <https://www.hospitalitynet.org/file/152001463.pdf> (dostęp: 24.11.2021).

⁴⁹ *Ibid.*

⁵⁰ *Agile Enterprise Concepts and Some Results of Research*, R. Trzcieliński (red.), University of Technology and IEA Press, Poznań 2007; A.M. Hormozi, *Agile manufacturing: the next logical step*, “Benchmarking: An International Journal” 2001, vol. 8, no. 2, s. 132–143.

⁵¹ W. Aghina, A. de Smet, K. Weerda, *Agility: It rhymes with stability*, “McKinsey Quarterly”, December 2015; Brosseau D. et al., *The journey to an agile organization*, “McKinsey Insights”, May 2019.

⁵² J. Bessant et al., *Developing the agile enterprise*, “International Journal of Technology Management” 2002, vol. 24, no. 5/6, s. 484–497.

⁵³ A. Ganguly, R. Nilchiani, J.V. Farr, *Evaluating agility in corporate enterprises*, “International Journal of Production Economics” 2009, vol. 118, no. 2, s. 410–423.

⁵⁴ R. Raschke, J.S. David, *Business process agility*, w: *Proceedings of the 11th Americas Conference on Information Systems*, Omaha 2005, s. 355–360.

⁵⁵ H. Sharifi, Z. Zhang, *A methodology for achieving agility in manufacturing organisations: An introduction*, “International Journal of Production Economics” 1999, no. 62, s. 7–22.

- 4) umiejętnie wykonywanie zadań i kontrola ich realizacji,
- 5) łączenie „zarządzania wizjonerskiego z operacyjnym”, co oznacza szerzenie idei i osadzanie ich w działaniach organizacji,
- 6) sprawna ocena adekwatności zasobów,
- 7) umiejętność pozyskiwania zasobów z otoczenia⁵⁶.

Według McKinsey&Company zwinne organizacje są „otwarte, inkluzywne i niehierarchiczne, ewoluują nieustannie bez częstych destrukcyjnych restrukturyzacji wymaganych w bardziej mechanicznych organizacjach; z większą pewnością przyjmują niepewność i dwuznaczność”⁵⁷. Według autorów raportu taki sposób działania pozwala wierzyć, iż organizacje te są znacznie lepiej przygotowane na przyszłość⁵⁸.

1.6. Aspekty metodologiczne zwinności

Istotnym aspektem koniecznym do rozstrzygnięcia są kwestie metodologiczne dotyczące zwinności. Zarządzanie projektami, a w szczególności zwinne zarządzanie projektami są dziedzinami silnie osadzonymi w praktyce. Powoduje to, iż w mniejszym stopniu przykładało się w nich uwagę do porządku terminologicznego, a bardziej do prakseologicznego, tj. skutecznego działania. Zwykle dopiero na dalszych etapach zaangażowanie badaczy i naukowców prowadzi do uporządkowania i nadania dziedzinie bardziej ustrukturyzowanej postaci⁵⁹. W literaturze spotkać można wprost zarzuty, iż: „definicje zwinności występujące w dyscyplinach zarządzania projektami (PM) i zwinnego zarządzania projektami (APM) są niespójne, niekompletne i niejasne”⁶⁰. Inni autorzy podkreślają także brak solidnych podstaw teoretycznych tworzonych rozwiązań⁶¹.

Czym zatem z perspektywy metodologii zarządzania będzie *agile*? Na to pytanie nie ma również jednej dobrej odpowiedzi. P. Paterek i A. Kozarkiewicz konstatują, iż w li-

⁵⁶ G. Belz, A. Barbasz (red.), *Research Papers. Management Forum 4*, Wydawnictwo Uniwersytetu Ekonomicznego we Wrocławiu, Wrocław 2014.

⁵⁷ McKinsey, *Leading agile transformation: The new capabilities leaders need to build 21st-century organizations*, 2018, nr October, s. 27, <https://www.mckinsey.com/~media/mckinsey/business%20functions/organization/our%20insights/leading%20agile%20transformation-the-new-capabilities-leaders-need-to-build-21st-century-organization> (dostęp: 20.04.2020).

⁵⁸ McKinsey & Company, *Leading agile transformation: The new capabilities leaders need to build 21st-century organizations*, <https://www.mckinsey.com/~media/mckinsey/business%20functions/people%20and%20organizational%20performance/our%20insights/leading%20agile%20transformation%20the%20new%20capabilities%20leaders%20need%20to%20build/leading-agile-transformation-the-new-capabilities-leaders-need-to-build-21st-century-organizations.pdf> (dostęp: 24.11.2021).

⁵⁹ P. Wyrozębski, *PMO in Focus of Management Studies*, w: *Kommunikation in multikulturellen Projektteams*, S. Grucza, J. Alnajjar (Hrsg.), Peter Lang Verlag, Frankfurt am Main 2015.

⁶⁰ E.C. Conforto et al., *The agility construct...*

⁶¹ J. Sheffield, J. Lemétayer, *Factors associated with the software development agility of successful projects*, „International Journal of Project Management” 2013, vol. 31, s. 459–472.

teraturze funkcjonują „bardzo często równoznacznie traktowane i zamiennie stosowane podjęcia określające podejście zwinne”⁶². W oryginalnych źródłach stosuje się nieraz zamiennie określenia typu *software development approaches* (podejścia), *agile frameworks* (w języku polskim najlepszym chyba odpowiednikiem będzie słowo „model”, ale stosuje się też określenie „rama postępowania”), jak również *agile development methodologies* (metodyki).⁶³ O ile sygnatariusze Manifestu nie mieli nic przeciwko stosowaniu słowa „metodyka” jako takiego⁶⁴, o tyle jednak pojęcie to było kojarzone ze starym porządkiem, ciężkimi metodykami planistycznymi i raczej odstawiane na bok. I tak np. SCRUM, który popularnie określa się jako metodykę, przez samych autorów nazywany jest „ramą postępowania” (ang. *framework*) i pozycjonowany jako „rama dla innych technik, metodyk i praktyk”⁶⁵. Wobec różnorodności rozwiązań i odmiennych nazw nawet profesjonalni tłumacze i wydawniczy konsultanci merytoryczni napotykają pewne trudności⁶⁶. Za zestawieniem opracowanym przez P. Paterek i A. Kozarkiewicz można zatem mówić o związanym z tym szerokim zakresem pojęciowym (tablica 3).

Tablica 3. Pojęcia stosowane w literaturze fachowej w dziedzinie zwinnego zarządzania projektami

- podejście zwinne (*agile methods*)
- metody lub metodyki zwinne (*agile methodologies* lub *agile methods*)
- procesy zwinne (*agile processes*)
 - zwinne zarządzanie projektami (*agile project management*)
 - zwinne wytwarzanie oprogramowania (*agile software development*)
 - zwinne wytwarzanie systemów (*agile system development*)
 - zwinność jako cecha (*agility*) przypisywana w różnych kontekstach zarządzania
 - podejście lekkie
 - koncepcja i filozofia zwinna
 - metody i metodyki lekkie

Źródło: P. Paterek, A. Kozarkiewicz A., *Zwinne zarządzanie zespołami projektowymi*, C.H. Beck, Warszawa 2020, s. 27.

Z punktu widzenia metodologii nauk o zarządzaniu przedstawiane w publikacji rozwiązania bez względu na intencję ich autorów spełniają definicję „**metod zarządzania**”, czyli „**celowo i świadomie opracowanych zespołów zaleceń postępowania, nadających się do wielokrotnego stosowania przy rozwiązywaniu problemów określonej klasy**”⁶⁷.

⁶² P. Paterek, A. Kozarkiewicz, *Zwinne zarządzanie zespołami projektowymi*, C.H. Beck, Warszawa 2020, s. 27.

⁶³ <https://www.mendix.com/agile-framework/> (dostęp: 24.11.2021).

⁶⁴ <https://agilemanifesto.org/history.html> (dostęp: 24.11.2021).

⁶⁵ K. Schwaber, J. Sutherland, *The Scrum Guide*, 2020, <https://scrumguides.org/scrum-guide.html> (dostęp: 9.06.2021).

⁶⁶ Agile Business Consortium, *Zwinne zarządzanie projektami – wersja 2. Wydanie polskie*, Agile Business Consortium, Warszawa 2016, s. 6.

⁶⁷ Z. Pawlak, M. Trocki, *Metody organizatorskie*, Wydawnictwo SGPiS, Warszawa 1986; Trocki M., *Podstawy metodycznego wsparcia zarządzania projektami*, w: *Metodyki i standardy zarządzania projektami*, M. Trocki (red.), Polskie Wydawnictwo Ekonomiczne, Warszawa 2017.

Kategoria ta będzie jednak zbyt ogólna jak na potrzeby niniejszej pracy. Idąc zatem głębiej, zgodnie z taksonomią M. Trockiego bazującego na wcześniejszych pracach J. Zieleniewskiego⁶⁸ i T. Pszczołowskiego⁶⁹, metody można podzielić pod względem kompleksowości oraz szczegółowości zaleceń⁷⁰. Podział taki pozwala wyróżnić cztery kategorie.

- 1) **„Reguły metodyczne (heurystyczne) zarządzania projektami** stanowią najbardziej ogólne metody zarządzania projektami. Mogą się one odnosić zarówno do zakresu problemowego odpowiadającego metodom kompleksowym, jak również do węższych zakresów odpowiadających metodom częściowym. Zalecenia co do sposobu postępowania, stanowiące treść reguł metodycznych nie określają ani kroków postępowania, ani ich treści, lecz jedynie ogólny kierunek postępowania”⁷¹.
- 2) **„Podejścia metodyczne zarządzania projektami** są to metody zarządzania projektami określające ogólne zasady postępowania przy rozwiązywaniu problemów zarządzania projektami zarówno całościowych, jak i częściowych. Podejście metodyczne zarządzania projektami odróżniają się od metodyk i technik zarządzania projektami wyższym poziomem ogólności”⁷².
- 3) **„Techniki zarządzania projektami** są to częściowe (wycinkowe), szczegółowe metody zarządzania projektami. Zawierają one zalecenia odnoszące się do części procesu zarządzania projektami, określając szczegółowo kroki postępowania prowadzące do uzyskania rozwiązania problemów częściowych. Techniki zarządzania projektami różnić mogą się znacznie w zależności od ich zakresu problemowego. Techniki zarządzania projektami odnoszące się do szerszego zakresu problemowego wykorzystywać mogą techniki o węższym zakresie problemowym”⁷³.
- 4) **„Metodyki zarządzania projektami** są to kompleksowe, szczegółowe metody zarządzania projektami. Zawierają one zalecenia odnoszące się do całego procesu zarządzania projektem, określając szczegółowo kroki postępowania prowadzące do uzyskania zamierzonego rezultatu. Metodyki zawierają zazwyczaj również wykaz metod szczegółowych koniecznych do zastosowania przy realizacji poszczególnych problemów częściowych”⁷⁴.

Wobec powyższego można zauważyć, iż zalecenia dotyczące zwinnego zarządzania odnajdziemy w każdej z tych kategorii. Przykładowo Manifest zwinnego programowania wraz z towarzyszącymi mu Dwunastoma zasadami zwinności wpisywać się będzie w re-

⁶⁸ J. Zieleniewski, *Organizacja i zarządzanie*, PWN, Warszawa 1097.

⁶⁹ T. Pszczołowski, *Zasady sprawnego działania*, Wiedza Powszechna, Warszawa 1982.

⁷⁰ M. Trocki, *Podstawy metodyczne zarządzania projektami*, w: *Metodyki zarządzania projektami*, M. Trocki (red.), Bizarre, Warszawa 2011, s. 17.

⁷¹ M. Trocki, *Podstawy metodyczne zarządzania projektami...*, s. 16–17.

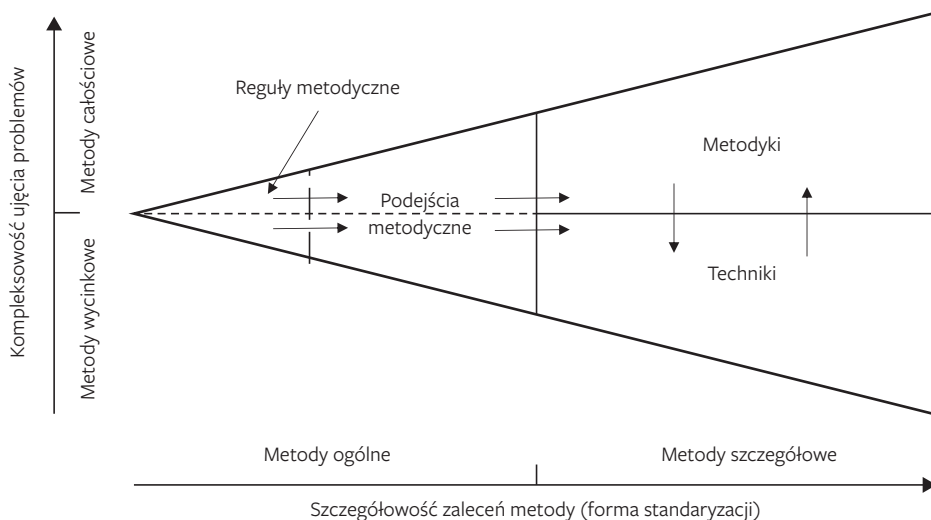
⁷² *Ibid.*

⁷³ *Ibid.*

⁷⁴ *Ibid.*

guły heurystyczne. Z kolei SCRUM, mimo innych intencji autorów, ze względu na bardzo wysoką szczegółowość zaleceń i towarzyszący im rygor stosowania spełnia wymagania pozwalające nazywać go w techniką lub, akceptując pewne obszary wyjęte z zakresu zaleceń, nawet metodyką zarządzania. Rozwiązania takie, jak np. AgilePM, w pełni wyczerpują znamiona rozbudowanej, kompleksowej metodyki⁷⁵. Zaś całościowo o *agile* mówić można również jako podejściu metodycznym.

Rysunek 9. Schemat metod zarządzania projektami



Źródło: M. Trocki, B. Grucza, K. Ogonek, *Zarządzanie projektami*, PWE, Warszawa 2009, s. 128.

W ujęciu metodologicznym w swoim ekosystemie *agile* odnajdziemy także elementy pochodzące z różnych poziomów hierarchii metodyk zarządzania projektami⁷⁶. Będą to:

- 1) **wytyczne metodyczne** (wspomniane wcześniej Manifest i Dwanaście zasad, ale również Wartości i Filary SCRUM⁷⁷),
- 2) **uniwersalne metodyki zarządzania projektami** (np. SCRUM, Kanban, Lean, Design Thinking, AgilePM i inne),
- 3) **metodyki branżowe** o zaleceniach stricte programistycznych (np. Feature Driven Development, Test Driven Development, Extreme Programming),
- 4) **metodyki problemowe** (np. Lean Startup lub Design Thinking),
- 5) **firmowe metodyki** (np. model opracowany przez firmę Spotify),

⁷⁵ Agile Business Consortium, *Zwinne zarządzanie projektami...*

⁷⁶ M. Trocki, *Podstawy metodyczne zarządzania projektami...*, s. 28.

⁷⁷ K. Schwaber, J. Sutherland, *The SCRUM Guide...*, s. 13.

- 6) **metody autorskie** (por. siedemnastu sygnatariuszy Manifestu zwinnego wytwarzania oprogramowania rozwijających opracowane przez siebie indywidualne, stosowane w swoich firmach metody pracy, np. Ken Beck i jego XP czy Alistair Cockburn i Crystal Agile Framework).

Liczba rozwiązań klasyfikowanych jako przynależące do zwinnego podejścia do zarządzania projektami jest znaczna, ale i nie definitywnie zamknięta. E. Alali zwraca na przykład uwagę, iż nawet klasyfikacja praktyk jako zwinnych lub niezwinnych nie jest zero-jedynkowa i bardziej związana z filozofią pracy niż przyklejeniem jednej lub drugiej etykiety⁷⁸. Mimo to jedną z prób zebrania ich razem pojął Christopher Webb, dyrektor Agile Strategy & Delivery w firmie doradczej Deloitte, tworząc tzw. krajobraz zwinności (ang. *agile landscape*).

Powyższe bogactwo i zróżnicowanie proponowanych rozwiązań powoduje konieczność przyjęcia bardziej elastycznego podejścia do kategoryzowania i oceny metod zwinnego zarządzania. W stosunku do omawianych w niniejszej publikacji metod stosowane będą kategorie, które – zdaniem piszącego te słowa – są najbliższe intencjom autorów, merytorycznej zawartości zaleceń oraz dominującym w branży określeniom fachowym. Szczegółowa kategoryzacja wszystkich dostępnych rozwiązań znacznie wykracza poza zakres niniejszej publikacji, ale może stanowić interesujący kierunek dalszych badań.

1.7. Fale rozwoju zwinności

Agile jest żywym i bardzo dynamicznym nurtem w programowaniu, zarządzaniu projektami, jak również w dziedzinie zarządzania w ogóle. Obserwując z perspektywy lat rozwój tej koncepcji, prof. Trocki identyfikuje trzy wymiary jej rozwoju:

- 1) rozwój pionowy (wertykalny),
- 2) rozwój poziomy (horyzontalny) oraz
- 3) rozwój metod wspierających zwinność (instrumentalny)⁷⁹.

Rozwój pionowy dotyczy poziomów hierarchii organizacyjnej, którą obejmują swoim wsparciem koncepcje i metody zwinne. W literaturze obserwacje te przypisuje się Ch. Ruddowi, twórcy i prezesowi Solution IQ (obecnie część Accenture), doświadczonemu „agiliście” z licznymi zwinnymi transformacjami przedsiębiorstw w portfolio⁸⁰. W 2016 r. dostrzegł on i opublikował na swoim blogu artykuł dotyczący tzw. trzech fal

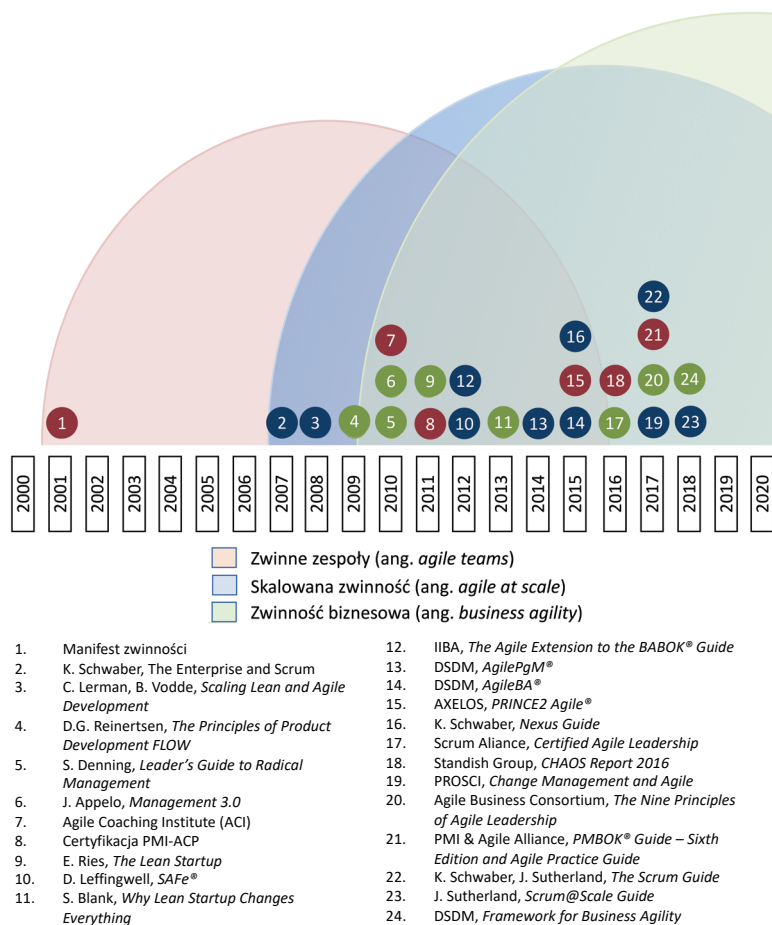
⁷⁸ E. Alali, *7 things wrong with Deloitte's Agile Tube Map*, <https://medium.com/tech-sojourn/7-things-wrong-with-deloittes-agile-tube-map-641192e20068> (dostęp: 17.06.2021).

⁷⁹ M. Trocki, *20 lat zwinnego zarządzania...*

⁸⁰ T. Kosiński, *Trzy fale zwinności w zarządzaniu*, „Studia I Prace Kolegium Zarządzania I Finansów” 2019, nr 172, s. 143–154, DOI: 10.33119/SIP.2019.172.9.

zwinności, oddających jego zdaniem ewolucję dziedziny na przestrzeni lat⁸¹. Koncepcja ta spotkała się z pozytywnym odbiorem środowiska⁸².

Rysunek 11. Trzy fale zwinności wg. Ch. Rudda i T. Kosińskiego



Źródło: T. Kosiński, *Trzy fale zwinności w zarządzaniu*, „Studia i Prace Kolegium Zarządzania i Finansów” 2019, nr 172, s. 143–154, DOI: 10.33119/SIP.2019.172.9, za: Ch. Ruud, *The Third Wave of Agile*, <https://www.solutionsiq.com/resource/white-paper/the-third-wave-of-agile-2/> (dostęp: 17.06.2021).

Pierwszą falę zwinności datuje on od opublikowania Manifestu zwinnego programowania, czyli od 2001 r., choć zdaniem piszącego te słowa można by ten okres rozciągnąć na połowę lat 90. XX w. Hasłem podsumowującym ten etap są „zwinne zespoły”. W tym czasie rozwój zwinności koncentrował się na opracowaniu metod pracy pozwalających

⁸¹ C. Rudd, *The Third Wave of Agile*, <https://www.solutionsiq.com/resource/white-paper/the-third-wave-of-agile-2/> (dostęp: 17.06.2021).

⁸² T. Kosiński, *Trzy fale zwinności w zarządzaniu...*; M. Trocki, *20 lat zwinnego zarządzania...*

relatywnie małym zespołom programistycznych radzić sobie z przytłaczającą złożonością i zmiennością zakresu realizowanych projektów nadzorowanych przez mikrozarządzających przedsięwzięciem kierowników projektów. Odpowiedzią na te problemy miały być samoorganizujące się, elastyczne zespoły, z osłabioną rolą kierownika projektu lub nawet pozbawione tej roli. Odrzuceniu liniowego przebiegu faz towarzyszyło wprowadzenie podejścia iteracyjnego i przyrostowego opartego na zmieniających się priorytetach i wysokich standardach jakościowych pracy. Z pierwszą falą zwinności związane były swoista kreatywność autorów oraz pojawienie się licznych, w różnym stopniu rozbudowanych i kompletnych metodyk zwinnych takich jak: SCRUM, Extreeme Programming, Crystal, Adaptive Software Development, Feature Driven Development, Dynamic System Development Method (DSDM) i innych. Spośród nich, jak już dzisiaj wiemy, tylko niektóre zdobyły szeroką popularność i przetrwały „młodzieńczy” okres rozwoju dziedziny. Według Ch. Rudda za ówczesne kamienie milowe uznać należy początek pierwszej fali z Manifestem z 2001 r. oraz powołanie Agile Coaching Institute (ACI) w 2010 r. i uruchomienie przez Project Management Institute certyfikacji Agile Certified Practitioner w roku kolejnym. T. Kosiński, który twórczo rozwinął oryginalną koncepcję Rudda, dodaje do tego zestawienia jeszcze trzy wydarzenia związane z rozwojem instrumentalnym w tej fazie. Są to: wydanie podręcznika Axelos *Prince 2 Agile*, prezentacja raportu Standish Group *Chaos Report 2016* oraz opublikowanie 6. edycji globalnego standardu PMI PMBoK wzbogaconego o elementy zwinności w 2017 r.⁸³ Wszystkie te zdarzenia świadczą o intensywnym rozwoju metodycznym zwinności prowadzącym do osiągnięcia dojrzałości na poziomie organizacji pracy zespołów w połowie drugiej dekady XX w., kiedy to oficjalnie zwinność została wprowadzona do głównego nurtu zarządzania projektami przez globalne organizacje fachowe. Swego rodzaju ciekawym epilogiem tej fazy jest stosunkowo późna (z 2017 r.), ale ważna publikacja K. Schwabera i J. Sutherlanda – autorów metody SCRUM. Nieporozumienia między nimi spowodowały, że ich ścieżki biznesowe rozeszły się. Założyli osobne przedsiębiorstwa i niezależnie od siebie rozwijali metodę. Skorzystały na tym inne organizacje oferujące w tym czasie zróżnicowane wersje SCRUM, jak np. *A guide to the SCRUM Body of Knowledge (SBOK Guide)*⁸⁴. Upływ czasu i utrata rynku na rzecz innych zabił jednak niezgodę, co doprowadziło to odnowienia współpracy i wspólnego opublikowania w 2017 r. *The SCRUM Guide*, od tej pory oficjalnej, kanonicznej i, można rzec, dość ortodoksyjnej wersji metodyki⁸⁵. W ten sposób symbolicznie pierwsza fala rozwoju zwinności zatoczyła koło.

Sukces zwinności na poziomie zespołów przyczynił się do pojawienia pytań dotyczących ekspansji na inne obszary organizacji oraz przełamania ograniczeń związanych z ory-

⁸³ Ibid.

⁸⁴ SCRUMStudy, *A guide to the Scrum Body of Knowledge (SBOK Guide)*, SCRUM Study, Phoenix 2013.

⁸⁵ K. Schwaber, J. Sutherland, *The SCRUM Guide...*

ginalnymi adresatami zaleceń, czyli małymi (liczącymi 7–10 osób), silnie zintegrowanymi zespołami. Pojawiło się pytanie o „skalowanie” metod zwinnych, czyli wykorzystanie ich zasad w kierowaniu przedsięwzięciami o znacznie liczniejszych zespołach i większym rozmiarze. Ch. Rudd za początek tej fazy uważa książkę K. Schwabera zatytułowaną *Enterprise and SCRUM* z 2007 r.⁸⁶ oraz drugą publikację, z 2008 r., autorstwa C. Larmana i B. Vodde’a, dotyczącą skalowania szczupłego i zwinnego rozwoju produktów⁸⁷. Podczas drugiej fali za przełomową uznaje się publikację D. Leffingwella prezentującą społeczności profesjonalistów *agile* metodę Scaled Agile Framework⁸⁸. Tak zwany SAFe stał się w późniejszym czasie jedną z najpopularniejszych metod przeniesienia koncepcji zwinności z poziomu zespołów wytwórczych i małych projektów na przedsięwzięcia duże, programy, portfele projektów oraz poziom strategiczny – dając przy tym rozbudowane rozwiązania planistyczne i koordynujące obejmujące swoim zasięgiem całą organizację. Rozbudowany charakter modelu (ang. *framework*) spowodował, iż zyskał on grono zarówno swoich zwolenników, jak i przeciwników⁸⁹.

Według T. Kosińskiego innymi ważnymi wydarzeniami tej fazy były:

- 1) wydanie w 2012 r. przez International Institute of Business Analysis *The Agile Extension to the BABOK® Guide* oraz „wprowadzenie w 2014 r. przez APMG we współpracy z Konsorcjum DSDM AgileBA®, nowego systemu wskazówek, szkoleń i certyfikacji stworzonego dla analityków biznesowych pracujących w środowisku zwinnym”⁹⁰,
- 2) wprowadzenie przez APMG standardu i certyfikacji zwinnego zarządzania programami „AgilePgM” w 2014 r.,
- 3) opublikowanie dwóch modeli skalowania zwinności: *Nexus Guide* K. Schwabera oraz *SCRUM@Scale Guide* J. Sutherlanda,
- 4) wpływ podejść zwinnych na obszar zarządzania zmianą i pojawienie się problematyki Agile Organizational Change Management (OCM)⁹¹.

Powodzenia w przeniesieniu *agile* na wyższe poziomy zarządzania zainicjowało trzecią falę, tzw. zwinność biznesową (ang. *business agility*), jako cechę, sposób i kulturę zarządzania całymi organizacjami. Według Ch. Rudda sygnałem rozpoczęcia tej fali była publikacja w 2010 r. książki *Management 3.0* autorstwa J. Appelo⁹², a dotyczącej współpracy między

⁸⁶ K. Schwaber, *The Enterprise and Scrum*, Microsoft Press 2007.

⁸⁷ C. Larman, B. Vodde, *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*, Addison-Wesley Professional, Boston 2008.

⁸⁸ D. Leffingwell, *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*, Addison-Wesley Professional, Boston 2011.

⁸⁹ K. Schwaber, *unSAFe at any speed...*

⁹⁰ T. Kosiński, *Trzy fale zwinności w zarządzaniu...*

⁹¹ T. Kosiński, *Trzy fale zwinności w zarządzaniu...*

⁹² J. Appelo, *Management 3.0: Leading Agile Developers, Developing Agile Leaders*, Addison-Wesley, Boston 2010; wyd. polskie: J. Appelo, *Zarządzanie 3.0, Kierowanie Zespołami z wykorzystaniem metodyk Agile*, Wydawnictwo Helion, Gliwice 2016.

menedżerami i zwinnymi zespołami oraz zmianie ich roli w zwinnych organizacjach. Wśród innych wpływowych osób autor wymienia także S. Denninga i jego poglądy na to, jak zwinne praktyki mogą pomóc w zarządzaniu, wdrażaniu strategii oraz zarządzaniu operacyjnym⁹³ oraz E. Riesa z przeniesieniem metod zwinnych na pole rozwoju działalności typu *start-up*⁹⁴ wraz z S. Blankiem i jego późniejszym artykułem w HBR rozwijającym prace Riesa⁹⁵. Ch. Rudd w swoim artykule odnosi się również do wielu innych autorów, których koncepcje są zgodne lub bezpośrednio wspierają postulaty podejścia zwinnego. W szczególności zajmuje się aspektami przywództwa, kreatywności czy zarządzania wiedzą⁹⁶.

Tabela 4. Obszary oddziaływania poszczególnych fal zwinności

Fale zwinności	Poziom zarządzania	Główni uczestnicy	Dziedzina zastosowania
Pierwsza fala: zespoły zwinne	zarządzanie projektami	członkowie zespołów, kierownicy niższego szczebla	wytwarzanie oprogramowania
Druga fala: skalowana zwinność	analizy biznesowe zarządzanie programami zarządzanie portfelem	kierownicy średniego szczebla	inne obszary działalności organizacji
Trzecia fala: zwinność biznesowa	zarządzanie przedsiębiorstwem	kierownicy wyższego szczebla, naczelne kierownictwo	całość działalności organizacji

Źródło: T. Kosiński, *Trzy fale zwinności w zarządzaniu*, „Studia i Prace Kolegium Zarządzania i Finansów” 2019, nr 172, s. 143–154, DOI: 10.33119/SIP.2019.172.9 (na podst.: Ch. Rudd, *The Third Wave of Agile*, <https://www.solutionsiq.com/resource/white-paper/the-third-wave-of-agile-2/> (dostęp: 17.06.2021)).

Jak wiemy, źródła podejścia zwinnego leżą w bardzo specyficznej kategorii projektów, jakimi są projekty tworzenia oprogramowania (ang. *software development*) w małych zespołach programistycznych. Jednak sukces *agile* oraz liczne korzyści płynące z jej stosowania sprawiły, iż szybko pojawiły się postulaty i zainteresowanie rozwojem podejścia w wymiarze poziomym/horyzontalnym. Na przestrzeni lat poszukiwano możliwości przeniesienia założeń zwinności na przedsięwzięcia inne niż programistyczne.

Wyniki badań pokazują, iż tak też dzieje się w praktyce. Według VersionOne z 2020 r. 56% ankietowanych potwierdziło stosowanie metod zwinnych w obszarach innych niż tworzenie oprogramowania. W jednej czwartej było to nadal IT, ale w innych zadaniach niż programistyczne. Co ósmy respondent potwierdził stosowanie *agile* z operacjach, zaś od 5% do 7% z nich w obszarach marketingu, kadr i sprzedaży⁹⁷.

⁹³ S. Denning, *The Leader's Guide to Radical Management: Reinventing the Workplace for the 21st Century*, Jossey-Bass, New York 2010.

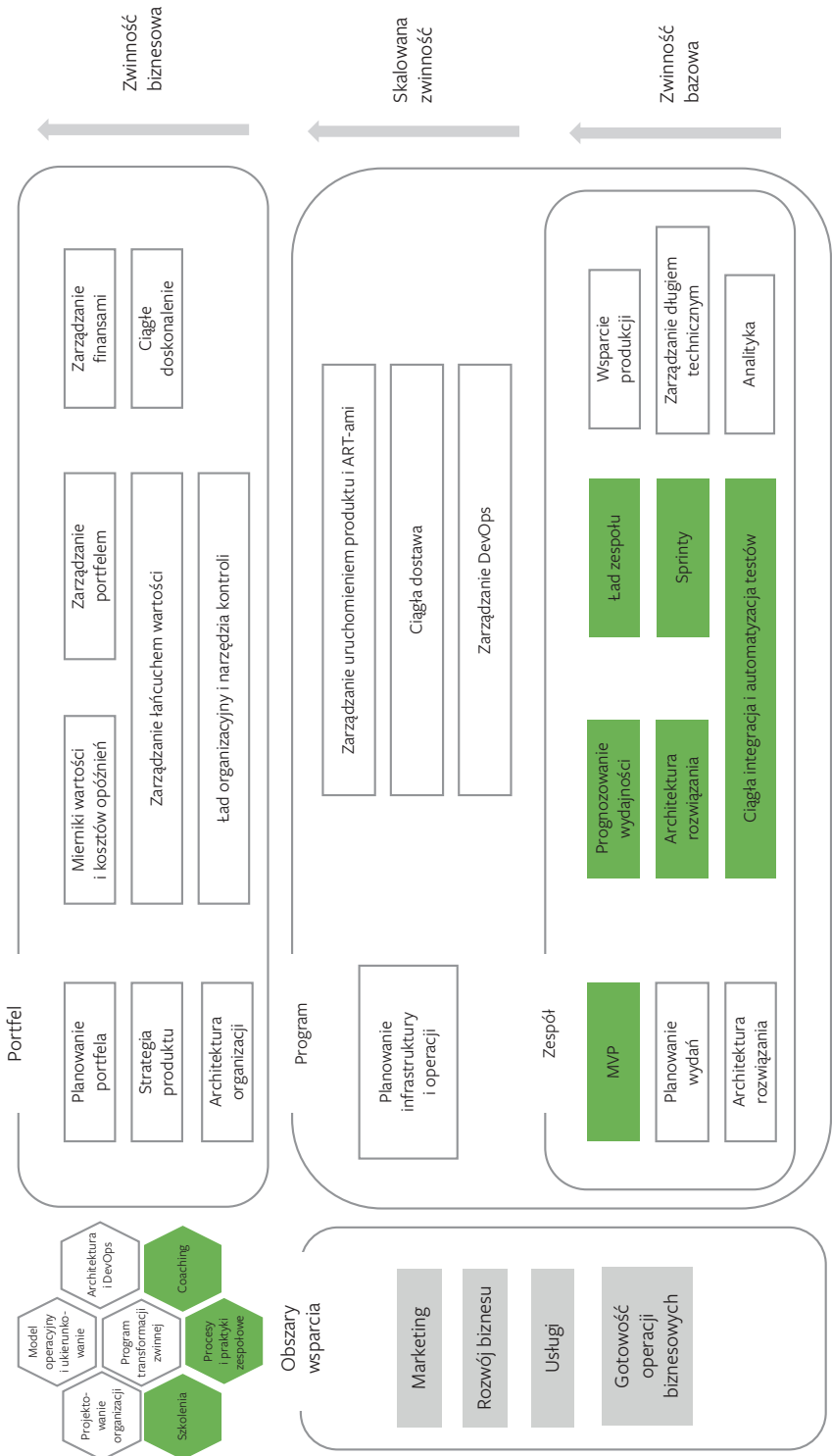
⁹⁴ E. Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Crown Business, New York 2011.

⁹⁵ S. Blank, *Why The Lean Startup Changes Everything*, „Harvard Business Review”, May 2013.

⁹⁶ C. Rudd, *The Third Wave of Agile...*

⁹⁷ VersionOne, *14th Annual State of Agile Report*, <https://www.qagile.pl/wp-content/uploads/2020/06/14th-annual-state-of-agile-report.pdf> (dostęp: 24.11.2021).

Rysunek 12. Trzy poziomy zwinności wg globalnej firmy konsultingowej Deloitte



Źródło: Deloitte, Agile Transformation Approach, <https://www2.deloitte.com/content/dam/Deloitte/global/Documents/About-Deloitte/gx-about-deloitte-agile-deloitte-agile-transformation-approach.pdf> (dostęp: 17.06.2021).

Warto w tym zakresie zwrócić uwagę na obszerne studia literaturowe przeprowadzone przez T. Gustavssona⁹⁸. Wychodząc z obserwacji, iż w popularnych wyszukiwarkach internetowych spotkać można liczne odniesienia do stosowania *agile* w projektowaniu, marketingu, działalności publikacyjnej czy usługach finansowych, postanowił sprawdzić, na ile towarzyszą im formalne opisy przypadków w literaturze naukowej. W tym celu dokonał on szerokiej systematycznej kwerendy w bazach danych w zakresie stosowania metod zwinnych poza oryginalnym obszarem tworzenia oprogramowania. Zidentyfikował w efekcie 21 studiów przypadków, których charakterystyki i źródła pokazano w tabeli 5.

Tabela 5. Studia przypadków stosowania zwinnych metodyk zarządzania projektami w obszarach innych niż IT

Autorzy (rok opublikowania artykułu)	Kontekst (rodzaj projektu, jeśli dotyczy)	Jakie „części” zwinnych sposobów pracy opisali jako wdrożone?	Zgłaszane korzyści
Andersson et al. (2006)	zarządzanie łańcuchem dostaw / produkcją	krótkie sprinty, codzienne spotkania stand-up i zespół PO	wzrost wydajności, szybkości i jakości
Denning (2015)	zarządzanie na najwyższym poziomie. praca strategiczna	koncentracja na wartości klienta, samoorganizujące się zespoły, krótkie sprinty i przejrzystość za pomocą wizualnych artefaktów	przejrzystość i współpraca w zespole, zwiększona interakcja z klientem, wydajność, jakość i szybkość
Edin Grimheden (2013)	edukacja, projekt rozwoju szkoleń	nie opisano	elastyczność, jakość, skupienie, dzielenie się wiedzą i radzenie sobie ze zmianą
Gangjun et al. (2009)	wzornictwo przemysłowe	planowanie iteracyjne, iteracyjna ewaluacja i śledzenie iteracyjne	wydajność, szybkość, elastyczność, dzielenie się wiedzą i radzenie sobie ze zmianą
Gangjun et al. (2010)	wzornictwo przemysłowe (projekty rozwoju produktów)	zarządzanie popytem, planowanie iteracyjne, iteracyjna ewaluacja i iteracyjne śledzenie	wydajność, szybkość i dzielenie się wiedzą
Molhanec (2008)	projektowanie produktów, opakowania i elektronika	iteracje i przeglądy.	nieopisane szczegółowo w artykule
Molhanec (2009)	projektowanie produktów, opakowania i elektronika	iteracje i przeglądy.	nieopisane szczegółowo w artykule
Niemi- Grundstrom (2014)	zarządzanie biblioteką	nie jest szczegółowo opisane w artykule	współpraca, produktywność, szybkość, elastyczność, jakość (zadania/cele/wymagania)

⁹⁸ T. Gustavsson, *Benefits of agile project management in a non-software development context. A literature review*, w: *Project Management Development – Practice and Perspectives: Fifth International Scientific Conference on Project Management in the Baltic Countries April 14–15, University of Latvia, Riga 2016*, s. 67–82.

cd. tabeli 5

Autorzy (rok opublikowania artykułu)	Kontekst (rodzaj projektu, jeśli dotyczy)	Jakie „części” zwinnych sposobów pracy opisali jako wdrożone?	Zgłaszane korzyści
Pope-Ruark (2015)	szkolnictwo wyższe (projekt rozwoju kursu)	krótkie sprinty, tablica scrumowa, codzienne stand-upy, planowanie sprintów, przeglądy i retrospektywy	współpraca w zespole, lepsze zrozumienie celów/zadań/wymagań i wyraźne poczucie postępu
Quaglia <i>et al.</i> (2011)	modelowanie symulacyjne w fabryce elektroniki	zdefiniowany wykaz prac (<i>backlog</i>), planowanie sprintów, 1–2 tygodniowe sprinty i przeglądy klientów.	współpraca w zespole, zrozumienie celów/zadań/wymagań, poprawa interakcji z klientem, priorytet procesu
Sommer <i>et al.</i> (2015)	produkcja, farmacja	tablice scrumowe, wykres spalania, codzienny scrum, wykaz prac produktu i model łańcucha wartości	współpraca w zespole, zrozumienie celów/zadań, poprawa interakcji z klientem, dzielenie się wiedzą i alokacja zasobów
Sommer <i>et al.</i> (2015)	produkcja, zabawki	tablice scrumowe, wykres spalania, codzienny scrum, wykaz prac produktu i pakiety robocze.	współpraca w zespole, zrozumienie celów/zadań/wymagań, priorytet, przejrzystość i autonomia
Sommer <i>et al.</i> (2015)	produkcja, elektronika	tablice scrumowe, wykres spalania, codzienny scrum, wykaz prac produktu	współpraca w zespole, poprawa interakcji z klientem, wydajność, szybkość, elastyczność i motywacja (mniej reklamacji)
Sommer <i>et al.</i> (2015)	produkcja, stolarka okienna	tablice scrumowe, wykres spalania, tygodniowy scrum, wykaz prac produktu i model łańcucha wartości	współpraca w zespole, poprawa interakcji z klientem, priorytetowy proces i współpraca
Sommer <i>et al.</i> (2015)	produkcja, przewody energetyczne	tablice scrumowe, wykres spalania, codzienny scrum, wykaz prac produktu, pakiety robocze	współpraca w zespole. poprawa interakcji z klientem, współpracy i motywacji
Sutherland, Altman (2009)	naczelne kierownictwo, strategia, wewnętrzne doradztwo	zasady scrum i tygodniowe sprinty	interakcja, zrozumienie celów/ zadań/wymagań, przejrzystość, proces usuwania przeszkód
Sutherland, Altman (2009)	zarządzanie, strategia, doradztwo wewnętrzne	zasady scrum i tygodniowe sprinty	wydajność. szybkość, przejrzystość i indywidualną autonomię
Sutherland <i>et al.</i> (2009)	non profit (projekt zmiany wewnętrznej)	zasady scrum	współpraca, przezroczystość, koncentracja i radzenie sobie ze zmianami i przeszkodami
Tolf <i>et al.</i> (2015)	opieka zdrowotna, zarządzanie szpitalem	nie opisane szczegółowo w artykule	elastyczność i motywacja, radzenie sobie ze zmianami i priorytetami
Van Ruler (2014)	public relations	zasady scrum	zwiększenie elastyczności i radzenie sobie ze zmianą
Wainer (2006)	szkolnictwo wyższe (projekt rozwoju kursu)	role scrum, krótkie sprinty, przeglądy i retrospektywy	współpraca i koncentracja w zespole, poprawa interakcji z klientem, produktywność, jakość i szybkość

Źródło: T. Gustavsson, *Benefits of agile project management in a non-software development context. A literature review*, w: *Project Management Development – Practice and Perspectives: Fifth International Scientific Conference on Project Management in the Baltic Countries April 14–15, University of Latvia, Riga 2016*, s. 67–82.

Wśród wymienionych wyżej branż dostrzec można: szkolnictwo wyższe, przemysł farmaceutyczny, konsulting, opiekę zdrowotną, produkcję zróżnicowanych dóbr: od elektroniki, przez okna, do dziecięcych zabawek, jak również logistykę czy zarządzanie na poziomie strategicznym. Zestawienie to dowodzi, iż idee i metody pracy proponowane przez zwinne metody zarządzania mają uniwersalny charakter, pozwalający osiągnąć ponadprzeciętne efekty w różnorodnych branżach i środowiskach zarządzania. Potwierdza również, iż rozwój zwinności nie zatrzymał się na trzech fazach, zaś pryncypia zwinności utrzymywać się będą w zainteresowaniu praktyków gospodarczych i badaczy nauk o zarządzaniu jeszcze przez wiele lat.

Podsumowanie

Zwinne zarządzanie projektami, a właściwie zwinne programowanie zaznaczyło swoją obecność z połowie lat 90. oraz na przełomie XX i XXI wieku. Początkowo traktowane było jako ciekawostka ograniczona tylko do subkategorii projektów i traktowana z dużą dozą nieufności. Z czasem okazało się, iż założenia nowego podejścia, choć rewolucyjne w swoim kształcie, dobrze trafiają w potrzeby nowoczesnych pracowników wiedzy⁹⁹, współczesnego biznesu, przedsiębiorstw i – szerzej – dynamicznie digitalizującej się gospodarki¹⁰⁰. Agile przestała być lokalną modą w zarządzaniu i szybko wyewoluowała, wspinając na wyższe poziomy organizacji, rozlewając się poza IT i rozwijając narzędziowo. We współczesnym biznesie trzeciej dekady XXI wieku mamy już do czynienia z w pełni ukształtowanym, dojrzałym i przenikającym wszystkie obszary działania przedsiębiorstw podejściem.

⁹⁹ P. Wyrozębski, *Zarządzanie wiedzą projektową*, Difin, Warszawa 2014.

¹⁰⁰ M.A. Afonassova et al., *Digitalization in economy and innovation: The effect on social and economic processes*, „Polish Journal of Management Studies” 2019, t. 19, nr 2.

2 PRZEGLĄD NAJWAŻNIEJSZYCH ZWINNYCH METODYK ZARZĄDZANIA PROJEKTAMI

2.1. Evolutionary Project Management (EVO)

2.1.1. Geneza metodyki

Jednym z określeń przypisywanych metodykom zwinnym jest hasło „ewolucyjność”. Projekt oraz tworzony produkt mają „ewoluować”, dopasowując się do doprecyzowywanych i zmiennych warunków oraz oczekiwań otoczenia. Filozofia ta jest w oczywisty sposób w opozycji do tradycyjnego, planistycznego podejścia do realizacji projektów. Ideę ewolucyjności w tworzeniu systemów wprowadził do zarządzania projektami Tom Gilb w 1976 r. w swojej książce pt. *Software Metrics*¹⁰¹. Pomysł ów stał się przyczynkiem do stworzenia przez niego metodyki nazywanej Evolutionary Project Management, w skrócie EVO.

Tom Gilb związany był z branżą tworzenia oprogramowania od końca lat 50. XX w., kiedy to podjął pracę dla IBM. W kolejnych latach T. Gilb współpracował przy wielkich projektach dla przemysłu lotniczego, telekomów oraz branży elektronicznej. Jest autorem książek na temat zagadnień tworzenia oprogramowania oraz licznych artykułów naukowych. Gościł w charakterze wykładowcy na wielu uniwersytetach, takich jak Berkley, Stanford czy London School of Economics. Aktualnie, wraz ze swoim synem – Kaiem Gilbem – pracują jako konsultanci i trenerzy, wdrażając opracowane przez siebie metody w firmach takich jak Sony, IBM, Ericsson, HP, Intel, Citigroup i innych¹⁰².

Evolutionary Project Management należy do rodziny zwinnych metodyk zarządzania projektami. Jak podkreśla autor jest to metodyka uniwersalna, dająca się zastosować w przypadku zarządzania dowolnym typem projektów i rozwoju dowolnych produktów¹⁰³.

¹⁰¹ Gilb T., *Software Metrics*, Little, Brown, and Co., New York 1976.

¹⁰² <http://www.gilb.com/> (dostęp: 24.11.2021).

¹⁰³ *Ibid.*

Jedną z cech przypisywanych metodzie jest jej skalowalność – od małych do wielkich projektów. Według deklaracji autora, spośród mocnych stron metodyki na pierwszym miejscu należy wymienić fakt, iż jest to „jedyna znana metoda, która historycznie ma ciągle pasmo nieprzerwanych sukcesów i brak porażek”¹⁰⁴, a przy tym „EVO czyni porażkę projektu strukturalnie niemożliwą”¹⁰⁵. Jakkolwiek stwierdzenie to może wydawać się na wyrost, to faktem jest, iż od ponad pół wieku ewolucyjne i inkrementalne podejście do wytwarzania produktów z powodzeniem wykorzystywane jest w zarządzaniu projektami i to tam odnaleźć można korzenie ruchu *agile*¹⁰⁶. Szczegółowe informacje dotyczące założeń metodyki EVO zawarte zostały w obszernym podręczniku autorstwa Kaia Gilba pt. *EVO. Evolutionary Project Management and Product Development*, opublikowanym w 2007 r. na stronie internetowej Toma i Kaia Gilbów (www.gilb.com). Metodkę EVO spotkać można także pod innymi nazwami, takimi jak: Evolutionary Delivery, Evolutionary Management, Requirements Driven Project Management oraz Competitive Engineering¹⁰⁷.

2.1.2. Założenia i struktura metodyki

Założenia metodyki EVO koncentrują się na idei ewolucyjności, przy czym autorzy celowo starają się odróżnić rozwój „ewolucyjny” od „inkrementalnego/przyrostowego”. Wskazując na zalety swojego podejścia, podkreślają konieczność ciągłej nauki, wprowadzania zmian oraz dopasowywania się do oczekiwań interesariuszy projektu, którzy odgrywają w projektach EVO znaczącą rolę. Metodyka kładzie nacisk na zdobywanie wiedzy, naukę, poznawanie wymagań i oczekiwań klientów, właściwych sposobów dostarczania rozwiązań, czy np. rzeczywistych kosztów projektów. Jest to możliwe dzięki wczesnemu i częstemu korzystaniu z informacji zwrotnych, które dostarczają wiedzę o stanie projektu i stosunku interesariuszy do jego rozwoju. Podobnie jak inne metodyki zwinne EVO stosuje krótkie i szybkie iteracje – tzw. cykle ewolucyjne (ang. *evolutionary cycles*) trwające np. w HP tydzień, a w Microsoft Corporation sześć tygodni.

Przedstawiając główne założenia EVO, Tom i Kai Gilbowie zdefiniowali dziesięć charakteryzujących metodykę pryncypiów¹⁰⁸:

- 1) Wcześniej i często dostarczane realne rezultaty, tworzące wartość dla realnych interesariuszy.

¹⁰⁴ *Ibid.*

¹⁰⁵ *Ibid.*

¹⁰⁶ C. Larman, V.R. Basili, *Iterative and Incremental Development: A Brief History*, “IEEE Computer”, June 2003, s. 2–11.

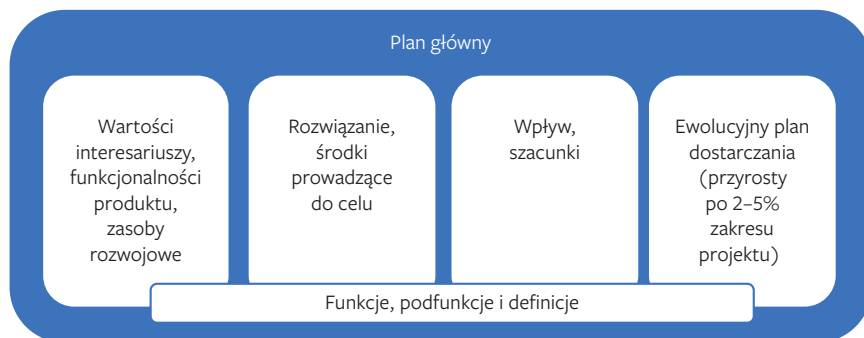
¹⁰⁷ T. Gilb, K. Gilb, *EVO. Evolutionary Project Management and Product Development*, <https://www.gilb.com/evo-book-preview-sign-up> (dostęp: 24.11.2021).

¹⁰⁸ <http://blogs.agilefaqs.com/2009/01/29/10-evolutionary-project-management-evo-principles/>; <http://www.gilb.com/tiki-page.php?pageName=about-gilb> (dostęp: 24.11.2021).

- 2) Kolejny krok EVO dostarcza maksymalną, możliwą w danym czasie wartość dla interesariuszy.
- 3) Kroki EVO pozwalają ewolucyjnie rozpoznawać i realizować definiowane na bieżąco wymagania.
- 4) Nie jesteśmy w stanie z góry przewidzieć właściwych wymagań, ale możemy odkryć je szybciej poprzez próby dostarczenia realnej wartości realnym interesariuszom.
- 5) EVO ma charakter holistycznej inżynierii oprogramowania: wszystkie niezbędne aspekty system muszą być ukończone, poprawne i dostarczone realnym interesariuszom – nie chodzi tylko o programowanie, ale o satysfakcję klienta.
- 6) Projekty EVO pracują na otwartej architekturze – dzieje się tak, ponieważ będziemy zmieniali rozwiązania w projekcie tak często, jak to potrzebne – jest to konieczne aby dostarczyć rzeczywistych wartości naszym interesariuszom;
- 7) Zespół EVO będzie koncentrował swoją kolektywną energię na odniesieniu sukcesu w bieżącym kroku EVO. Porażka lub sukces kroku będą wspólne. Zespół nie marnuje energii na dalsze kroki, do momentu aż nie zakończy z powodzeniem bieżącego.
- 8) W EVO zespoły powinny korzystać z doświadczeń, aby jak najszybciej weryfikować rzeczywistą wartość i realne działanie dostarczanych rozwiązań. EVO ma na celu wczesne skonfrontowanie zespołu z problemami – takie, które pozwoli uzyskać szybkie postępy po znalezieniu rozwiązania.
- 9) EVO zapewnia wczesne i terminowe dostarczanie produktów – jest to możliwe dzięki wczesnemu ustaleniu priorytetów oraz wiedzy, jak prawidłowo i we właściwym czasie wykonywać zadania.
- 10) EVO powinno umożliwić nam rozwój nowych i wczesne pozbycie się złych procesów w wykonywanej pracy.

Struktura metodyki EVO przedstawiona została na poniższym schemacie:

Rysunek 13. Struktura metodyki EVO

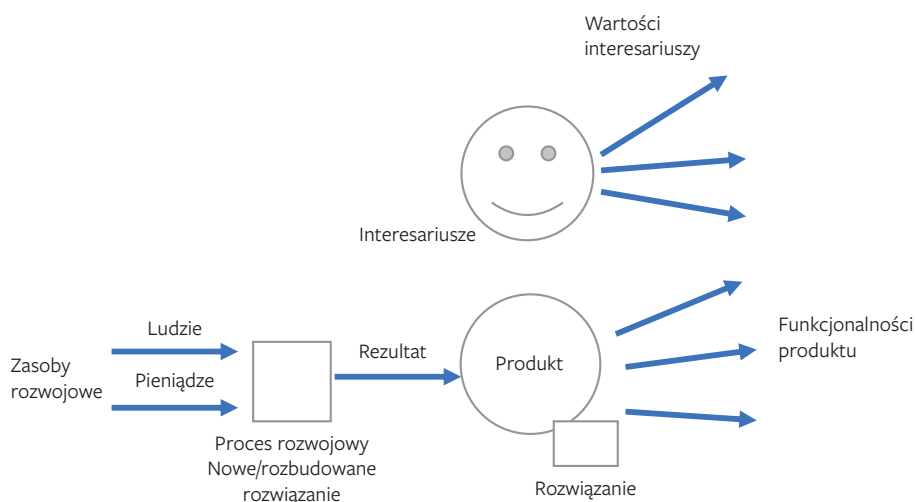


Źródło: T. Gilb, K. Gilb, *EVO. Evolutionary Project Management and Product Development*, <https://www.gilb.com/evo-book-preview-sign-up> (dostęp: 24.11.2021).

Zgodnie z metodyką EVO projekt rozpoczyna się od rozpoznania i zdefiniowania interesariuszy, ich oczekiwań względem udoskonań (ang. *stakeholder values*) oraz określenia poziomu cech jakościowych (własności, funkcjonalności) produktu, który ich usatysfakcjonuje (ang. *product quality requirements*)¹⁰⁹. Na tym etapie określone są także zasoby dostępne dla zespołu projektowego (ang. *project development resources*). Przyznane zasoby zużywane będą stopniowo w kolejnych ewolucyjnych krokach cyklu. Będą stanowić one ograniczenie, którego zespołowi nie wolno przekroczyć. Istotą tego etapu jest rozpoznanie kluczowych parametrów projektu: krytycznych interesariuszy, jakich usprawnień oczekują, ile środków są w stanie na nie przeznaczyć oraz jak będzie się to zmieniało z biegiem czasu. Podkreślenie rangi interesariuszy na etapie definiowania projektu ma za zadanie zapewnić realny, a nie wymaginowany charakter wymagań, założeń i wartości przynoszonych w efekcie projektu.

Na podstawie uzyskanych informacji przygotowywany jest wysokopoziomowy opis docelowego rozwiązania (ang. *solution*). Rozwiązanie (jedno lub wiele) jest sposobem, w jaki zespół projektowy uzyska przyjęte wartości interesariuszy oraz wymagania jakościowe produktu. Odnosząc poszczególne elementy do siebie, autorzy metodyki traktują funkcje jako odpowiedź na pytanie „co?”, cechy jakościowe – „jak dobrze? / w jakim stopniu?” oraz rozwiązania – „jak?”.

Rysunek 14. Relacje między pojęciami stosowanymi w metodyce EVO



Źródło: *ibid.*

¹⁰⁹ Wśród nazw stosowanych do określenia *stakeholder value* & *product quality requirements* można także spotkać określenia takie jak: *objectives, strategic goals, requirements, aims, ends, targets, purposes, ambitions, qualities, non-functional requirements, intentions* czy *performance attributes*.

poziom konsumpcji zasobów, ewentualne współzależności i inne). Tabela pozwala także utrzymać ewolucyjny projekt w ramach istniejących ograniczeń czasu i zasobów.

Zakończenie prac przy tabeli umożliwia przystąpienie do opracowania planu pracy zespołu czyli tzw. ewolucyjnego planu dostarczania (ang. *evolutionary delivery plan*). Jest to iteracyjny plan, który koncentruje się na opisie sposobu dostarczania usprawnień opisanych w wartościach interesariuszy i cechach produktu, nie zaś na poszczególnych rozwiązaniach.

Poprzez dobór pomysłów z listy rozwiązań oraz tworzenie ewolucyjnych cykli dostarczania zespół uszczegóławia, rozwija i steruje cechami produktu tak, aby były one zgodne z oczekiwaniami interesariuszy. W trakcie pracy zespół powinien gromadzić informacje zwrotne i opinie na temat efektów pracy, uzyskując coraz lepsze rozeznanie w projekcie oraz modyfikując odpowiednio jego przebieg i plany.

Proces realizacji projektu przedstawiony został w siedmiu krokach przedstawionych na tablicy 4:

Tablica 4. Siedem kroków do sukcesu w zarządzaniu projektem EVO

1. Interesariusze: na początku identyfikuję naszych wszystkich interesariuszy.
2. Wyzwania: dowiaduję się, co chcę osiągnąć, jakie są ich wartości (ang. *stakeholder values*) oraz w jakich obszarach trudno im je uzyskać.
3. Wartości interesariuszy: skupiam się na pomocy interesariuszom w uzyskaniu niektórych z ich wartości, w szczególności w obszarach trudnych; definiuję wartości interesariuszy i ustalam poziom celów (ang. *goals*), przez co stają się wymaganymi wartościami interesariuszy (ang. *stakeholder value requirements*).
4. Cechy produktu: następnie dowiaduję się, jakie cechy produktów (ang. *product qualities*) i zasoby (ang. *development resources*) konieczne są do osiągnięcia wartości interesariuszy; ustalam wymagane cechy produktu (ang. *product quality requirements*) oraz budżet (ang. *development resource budget*).
5. Rozwiązania: odnajduję i opracowuję rozwiązania (ang. *solutions*), które w ramach dostępnych zasobów utworzą produkt z pożądanymi cechami jakościowymi; posługując się tabelą szacowania wpływu (ang. *impact estimation table*), określam, które rozwiązania są najbardziej efektywne z perspektywy dostępnych zasobów.
6. EVO: przydzielam wybrane rozwiązania do krótkich (1–2 tygodnie) cykli ewolucyjnych (ang. *EVO cycles*); każdy cykl ma usprawnić cechy produktu zgodnie z przyjętymi poziomami celów; w każdym z cykli weryfikujemy, jakie usprawnienia faktycznie zostały osiągnięte.
7. Powrót: wracam do początku, najczęściej jednak przechodząc szybko przez pierwsze cztery kroki bezpośrednio do punktu 5 lub 6. Powtarzam do momentu osiągnięcia sukcesu projektu.

Źródło: *Ibid.*

2.1.3. Podsumowanie

Podsumowując metodykę EVO, należy zdecydowanie podkreślić jej koncentrację na zapewnieniu satysfakcji interesariuszom. Identyfikacja interesariuszy oraz rozpoznanie ich wartości i oczekiwań jest kluczowym elementem metodyki oraz krokiem, którego nie sposób pominąć. Sukces projektu budowany jest poprzez ewolucyjne odkrycie najlepszej konfiguracji dostępnych rozwiązań realizujących ich cele. Mimo nawiązywania do interesariuszy przez wiele, zarówno tradycyjnych, jak i zwinnych metodyk zarządzania projektami (np. PMBoK, SCRUM), w EVO stają się oni kluczowymi postaciami projektu.

EVO wniosło interesujący wkład w rozwój metodyk zwinnych poprzez tabelę szacowania wpływu. Techniki macierzowe, takie jak LogFrame, czy Canvas, funkcjonują szerzej w zarządzaniu projektami, jednak narzędzie rekomendowane przez EVO, wspierając eksperymentowanie i odkrywanie najbardziej wartościowych konfiguracji, jest znacznie elastyczniejsze niż inne rozwiązania. Będzie ono poza tym pomocne nie tylko na etapie definiowania i planowania projektu, ale także podczas jego realizacji. Oceny i wnioski z tabeli umożliwiają rozstrzygnięcie o akceptacji bądź odrzuceniu zmian w projekcie.

Wśród słabszych stron omawianej metodyki należy zwrócić uwagę na brak rozwiązań w zakresie struktury zespołu projektowego oraz zaleceń pracy w ramach iteracji. Przyjmuje ona w tym zakresie generyczne zalecenia, które spotkać można w przypadku rodzinnych metodyk zwinnych.

Główną ideą EVO jest ewolucja. Dostrzec ją można nie tylko w warstwie filozoficznej, ale i sposobie kontroli projektu poprzez tabele wpływu. Definiując cechy jakościowe interesariusze najczęściej określać będą stan obecny i docelowy oraz czas jego osiągnięcia. Poszukiwanie rozwiązań oraz wszelkie zmiany w trakcie projektu mają umożliwić jak najlepsze wyewoluowanie produktu projektu. Metodyka EVO lub jej elementy mogą zatem zostać zarekomendowane jako wzorzec postępowania w przypadku projektów optymalizacyjnych oraz rozwojowych, dotyczących doskonalenia już istniejących produktów.

2.2. Agile Project Management (APM)

2.2.1. Geneza metodyki

Wyrażenie *agile project management*, tłumaczone jako „zwinne zarządzanie projektami”, utożsamiane jest najczęściej, tak jak zostało to przedstawione we wcześniejszej części opracowania, z określoną filozofią i podejściem do realizacji projektów. Podejście to, oparte m.in. na iteracyjnym budowaniu rezultatów projektów, cechuje liczną grupę metod (metodyk) zjednoczonych pod sztandarem Manifestu zwinności. Zgodnie z powyższym APM odnosić się będzie do obszernego nurtu elastycznego zarządzania projektami i będzie miało charakter uniwersalny – oznaczać będzie „rodzinę” metodyk zwinnych będącą przedmiotem niniejszego opracowania. Jednocześnie jednak ta sama nazwa wykorzystana została w 2004 r. do oznaczenia jednego z podejść w ramach tejże grupy. Uczynił to Jim Highsmith, tworząc model Agile Project Management, (APM), który opisany został w publikacji jego autorstwa pod tytułem *Agile Project Management: Creating Innovative Products*¹¹⁰. Publikacja

¹¹⁰ J. Highsmith, *Agile Project Management: Creating Innovative Products*, Pearson Education/Addison-Wesley, New York 2004.

szybko doczekała się polskiego tłumaczenia i wydana została w 2005 r. w wydawnictwie Mikom pod nazwą *APM Agile Project Management. Jak tworzyć innowacyjne produkty*¹¹¹.

Autor metodyki Jim A. Highsmith urodził się w 1945 r. w USA. Mimo iż zdobył tytuł licencjata w dziedzinie inżynierii elektrycznej (ang. *electrical engineering*), a magistra w zarządzaniu, to swoją karierę zawodową związał z branżą inżynierii oprogramowania. Tuż po graduacji brał udział m.in. przy tworzeniu oprogramowania w ramach kosmicznej misji Apollo. W ciągu ponad trzydziestu lat pracy w branży informatycznej pracował jako menedżer działów IT, menedżer projektów, menedżer produktów, konsultant oraz programista w firmach operujących w USA, Europie, Chinach, Kanadzie, RPA, Japonii, Indiach, czy Nowej Zelandii¹¹². Jim Highsmith jest jedną z kluczowych osób w społeczności profesjonalistów zarządzania projektami informatycznymi. Jest jednym z autorów i sygnatariuszy Manifestu zwinności oraz „Deklaracji niepodległości” liderów projektów zwinnych¹¹³. Jest również założycielem i członkiem pierwszego zarządu The Agile Alliance¹¹⁴ oraz Agile Project Leadership Network¹¹⁵ – dwóch globalnych organizacji zrzeszających specjalistów w dziedzinie zwinnego zarządzania projektami. Jego znaczący wkład w postęp wiedzy teoretycznej i praktycznej w zakresie metod inżynierii oprogramowania i systemów uhonorowanych został w 2005 r. Międzynarodową Nagrodą Stevensa¹¹⁶.

2.2.2. Struktura modelu

Przedstawiony przez autora model adaptacyjnego zarządzania projektem ma charakter uniwersalny i znajduje zastosowanie nie tylko w wytwarzaniu oprogramowania, ale także w różnego typu projektach o charakterze innowacyjnym i eksploracyjnym. J. Highsmith zdefiniował pięć kluczowych celów biznesowych przyświecających adaptacyjnemu zarządzaniu projektami.

Po pierwsze jest to „ciągła innowacja” – zespół projektowy powinien zapewnić, iż uzyskany produkt w jak największym stopniu jest zgodny z oczekiwaniami klienta. Zespoły adaptacyjne powinny charakteryzować otwartość, zdolności do samoorganizacji i samodyscypliny. Cechy te wspierają kreatywność i pomysłowość w doskonaleniu jakości i użyteczności tworzonych produktów.

Drugim celem jest „zdolność adaptacyjna produktu”. Model APM ma zapewniać możliwość dostosowania i doskonalenia produktów także w przyszłości. Proces tworze-

¹¹¹ J. Highsmith, *APM Agile Project Management. Jak tworzyć innowacyjne produkty*, Mikom, Warszawa 2005.

¹¹² Highsmith J., biogram, <http://www.dorsethouse.com/authors/highsmith.html> (dostęp: 24.11.2021).

¹¹³ Declaration of Interdependence, <http://pmdoi.org/index.html> (dostęp: 07.02.2020).

¹¹⁴ <http://www.agilealliance.org/> (dostęp: 24.11.2021).

¹¹⁵ <http://www.agileleadershipnetwork.org/> (dostęp: 24.11.2021).

¹¹⁶ Stevens Award, Stevens Lecture on Software Development Methods, <http://reengineer.org/stevens/previous.htm> (dostęp: 2.03.2020).

nia innowacyjnych produktów ma umożliwić antycypację zmian, które mogą pojawić się w przyszłości oraz stosować narzędzia i praktyki obniżające koszt dopasowania się do nich. Zasada ta bliska jest podejściu ewolucyjnemu, które traktuje ciągłą adaptację i dopasowywanie się do zmieniających się oczekiwań za czynnik sukcesu projektów.

Trzecim celem stawianym przez Highsmitha modelowi APM są „skrócone harmonogramy procesu wytwarzania”. Zastosowania APM ma umożliwić szybszą realizację projektów, a przez to skrócenie czasu między zainwestowaniem środków w projekt, a momentem kiedy zaczną one przynosić oczekiwane korzyści i zwrot finansowy. Efekt ten osiągany jest poprzez trzy sposoby:

- 1) dostarczanie priorytetowych elementów funkcjonalności produktu w krótkich, stałych okresach czasu (tzw. iteracjach),
- 2) koncentrację na tworzących wartość działaniach wytwórczych, przy ograniczeniu do niezbędnego minimum prac administracyjnych i dokumentacji,
- 3) nacisk na doskonalenie umiejętności oraz zapewnienie wysokich kompetencji członków zespołu projektowego.

Czwarty cele APM sformułowano jako „adaptacyjność ludzi i procesów”. Zmiana filozofii realizacji projektów z podejścia presytuacyjnego na adaptacyjne pociąga za sobą konieczną zmianę funkcjonowania zespołów oraz organizacji jako takiej. Adaptacyjność oznacza odejście od sztywnego rygoru planów i szczegółowej specyfikacji na rzecz stopniowej eksploracji docelowego kształtu produktu, przy nieraz niepełnych i zmieniających się oczekiwaniach klienta. Wiąże się to z częstą i intensywną komunikacją, współpracą w małych, silnie zintegrowanych, interdyscyplinarnych zespołach projektowych, ale także koniecznością eksperymentowania, a więc i akceptacją, że wybrana ścieżka okaże się czasem błędna. Wdrożenie modelu adaptacyjnego zarządzania projektami powinno być zatem rozpatrywane także z perspektywy jego wpływu na kulturę organizacyjną przedsiębiorstwa.

Ostatnim celem postawionym APM przez jego autora są „rzetelne wyniki”. Ze względu na eksploracyjny charakter projektów zarządzanych adaptacyjnie sposób ich oceny różni się od oceny projektów realizowanych tradycyjnie. Różnicę tę można sprowadzić do sformułowania, iż metodami tradycyjnymi dostarczamy odbiorcom „to, czego chcieli”, a metodami adaptacyjnymi „to, czego potrzebowali”. Spośród trzech parametrów „żelaznego trójkąta” projekty adaptacyjne funkcjonują w warunkach sztywnego ograniczenia czasu i kosztów. Zakres rozwijany jest ewolucyjnie, zgodnie z pierwotną wizją produktu, ale uwzględniając także zmieniające się oczekiwania i wymagania klienta, nową wiedzę na temat funkcjonowania produktu czy bieżące priorytety rozwoju funkcjonalności. Wypracowany zakres nie musi być zawsze zgodny z pierwotnym zamierzeniem (co jest naturalne w stosunku do projektów innowacyjnych), ale będzie realizował zasadę użycia maksymalnej użyteczności dla klienta.

Wśród innych fundamentalnych założeń modelu autorstwa J. Highsmitha należy zwrócić uwagę na jego zgodność z Manifestem zwinności. Mimo iż w pierwotnym znaczeniu adresowany był on do projektantów oprogramowania, J. Highsmith podjął się jego interpretacji, tak aby leżał także u podstaw uniwersalnego modelu APM. Dlatego też przyjął i zaakceptował cały tekst Manifestu, wprowadzając do niego jedną zmianę, tzn. zastępując „oprogramowanie” (w oryg. *software*) słowem „produkt” (ang. *product*). Wartości przyświecające APM stały się więc identyczne z wartościami przyświecającymi całemu ruchowi *agile*.

Budując swój model, J. Highsmith rozwinął dodatkowo wartości zawarte w Manifestie w sześć zasad przewodnich APM. Mają one być przewodnikiem zespołów adaptacyjnych w organizowaniu prac oraz opracowywaniu dopasowanych praktyk realizacji projektów. Praktyki te zostały przez niego podzielone na dwie grupy – związane z produktami i klientem oraz związane z zarządzaniem.

Tablica 5. Zasady przewodnie APM

Wartość dla klienta przez innowacyjne produkty:

- dostarczaj wartości klientowi
- zastosuj wytwarzanie iteracyjne oparte na dostarczaniu elementów funkcjonalności
- bądź orędownikiem doskonałości technicznej

Przywódczo-współpracujący styl zarządzania:

- zachęcaj do eksploracji
- buduj adaptacyjnie (samoorganizujące się, samodyscyplinujące się) zespoły
- upraszczaj

Źródło: J. Highsmith, *APM Agile Project Management. Jak tworzyć innowacyjne produkty*, Mikom, Warszawa 2005, s. 45.

Pierwsza zasada stawia sobie za cel dostarczanie klientowi wartości. Satysfakcja, zadowolenie i zgodność z oczekiwaniami klientów (tj. szeroko rozumianych użytkowników, zamawiających, decydentów itp.) staje się w ten sposób jednym z kryteriów oceny sukcesu projektów. Projekt oraz tworzony w jego efekcie produkt mają zaspokoić pewną potrzebę, umożliwić rozwiązanie problemu. Będzie to możliwe, jeśli produkt i jego funkcjonalności będą klientom przydatne. Aby stało się to możliwe, zespół projektowy powinien koncentrować swoje wysiłki na pracy wytwórczej, generującej faktyczny postęp pracy i rozwój produktu. W ten sposób najlepiej wykorzystają oni swój czas i kompetencje. W opinii autora modelu czynności administrowania, dokumentowania, planowania i kontroli nie mogą odciągać zespołów od prac realizacyjnych. Te pierwsze powinny zostać ograniczone do niezbędnego minimum oraz w miarę możliwości przeniesione na pomocniczy personel projektowy lub kierownictwo. Postulatu J. Highsmitha, powtarzanego także przez innych twórców metodyk zwinnych, nie należy interpretować jako nawoływania do rezygnacji z takich działań, a jedynie jako próbę przywrócenia należytych priorytetów i hierarchii między działaniami wytwórczymi a wspierającymi.

Druga zasada charakteryzuje sposób organizacji pracy zespołu podczas wytwarzania. Zgodnie z podejściem zwinnym produkty innowacyjne buduje się przyrostowo, inkrementalnie. Kolejne przyrosty, odzwierciedlające rozwój produktu, nazywane są iteracjami. W każdej iteracji zespół ma za zadanie rozbudować istniejącą wersję częściową o kolejne elementy budujące całość wizji i zakresu produktu. Opisując zakres poszczególnych iteracji, APM odrzuca formułowanie go w kategorii zadań/czynności. Jest on definiowany poprzez tzw. funkcjonalności produktu – opis sposobu działania językiem korzyści dla klientów. Identyfikacja funkcjonalności produktu znacznie poprawia komunikację między użytkownikami, którzy opisują, jak zamierzają korzystać z produktu, a zespołem wytwórczym, który mając wiedzę fachową, będzie te oczekiwania realizował. Czas poświęcony na każdą iterację jest ograniczony i krótki, tak aby zapobiec rozpraszaniu uwagi zespołu. Wśród licznych korzyści z wdrożenia tej zasady wskazać można m.in.: ograniczenie kosztów zmian, wczesne osiągnięcie korzyści z fragmentów prac oraz częstsze informacje zwrotne ograniczające poziom ryzyka projektów.

Trzecia zasada APM także nawiązuje do warstwy procesów wytwarzania i głosi postulat doskonałości technicznej. Dążenie zespołów *agile* do dostarczenia odbiorcom produktów jak najwyższej jakości wyraża się przede wszystkim poprzez zapewnienie wysokich kompetencji w zespole projektowym, kwalifikacje kierownika projektu w obszarze technologii, w którym operują jego ludzie, oraz zawarcie w poszczególnych iteracjach działań związanych zarówno z planowaniem, projektowaniem, budowaniem, jak i testowaniem elementów funkcjonalności. Nowe elementy funkcjonalności uzyskane na zakończenie iteracji muszą funkcjonować w sposób sprawny, tj. powinny być pozbawione wszelkich zidentyfikowanych w trakcie iteracji błędów i wad.

Druga grupa zasad związana jest ze stylem i kulturą zarządzania. W przypadku projektów realizowanych zgodnie z podejściem APM kultura zarządzania powinna przypominać bardziej styl oparty na przywództwie, współpracy i wyznaczaniu celów niż na mikrozarządzaniu w rękach kierownika projektu.

Zgodnie z zasadami w tym względzie kierownik projektów powinien zachęcać do eksploracji, inspirować członków zespołu do poszukiwań i odkrywania jak najlepszych rozwiązań opracowywanych produktów i sposobów działania. Atmosfera pracy i współpracy powinna sprzyjać eksperymentom, wymianie wiedzy, chęci podejmowania wyzwań oraz swobodnego zgłaszaniu propozycji usprawnień.

Tworzenie innowacyjnych produktów wymaga także innego podejścia do kierowania zespołem. APM promuje tworzenie zespołów adaptacyjnych. Zespołów składających się z ograniczonej liczby członków, reprezentujących różne specjalności, ale będących w nich ekspertami. Zespołów, których członkowie bardziej przystają do Y niż X według teorii motywacji McGregora. Sukces eksploracji w realizacji wizji projektu i produktu wymaga pozostawienia takim zespołom dużej dozy samodzielności, samoorganizacji

i samodyscypliny. W momencie przyjęcia celu odpowiedzialność za jego realizację spoczywa na zespole oraz na jego członkach. Sposób osiągnięcia celu, obciążenie pracą, podejmowanie decyzji itp. powinno pozostać w gestii zespołu. W takich warunkach, każdy z jego członków będzie mógł pracować zgodnie z swoją najlepszą wiedzą i umiejętnościami, w sposób pozwalający osiągnąć maksymalny potencjał. Istotą roli kierownika projektu będzie zatem pozwolić pracować w spokoju, nie przeszkadzać i przejąć na siebie czynności administracyjne wspierające projekt. Wprowadzenie tak daleko idącej swobody wymaga jednocześnie doskonałych kompetencji technicznych, osobowościowych i etycznych wśród pracowników zespołu. Może to stanowić wyzwanie dla organizacji podczas rekrutacji i tworzenia takich teamów, ale także dawać szansę odpowiednim ludziom na rozwój i doskonalenie zawodowe.

Ostatnia zasada przyświecająca podejściu APM to hasło „upraszczaj”. Prostota, w myśl J. Highsmitha, ma umożliwiać szybką i sprawną realizację projektów. Zgodnie z prezentowanym poglądem upraszczanie procedur, struktur i procesów zmuszać będzie ludzi do interakcji i współpracy, zaś proste zasady, tak jak te wyszczególnione w APM, pozwolą organizować prace w sposób nie gorszy niż wielostronicowe regulaminy – wskazując pożądaną cel i kierunek działania, pozostawiając jednocześnie decyzje o sposobie realizacji w rękach kompetentnych osób. Autor APM postuluje przede wszystkim zachowanie zdrowego rozsądku oraz filozofii „niezbędnego minimum”.

Przedstawione powyżej wartości i zasady towarzyszące podejściu APM opisują jego kontekst, filozofię i kulturę pracy, wspierające pozytywne zachowania, sprzyjające efektywnej, zgodnej pracy przy tworzeniu innowacyjnych produktów. Na tym fundamencie J. Highsmith zbudował strukturę procesu i praktyk APM, konkretyzujących zalecenia postępowania dla zespołów adaptacyjnych.

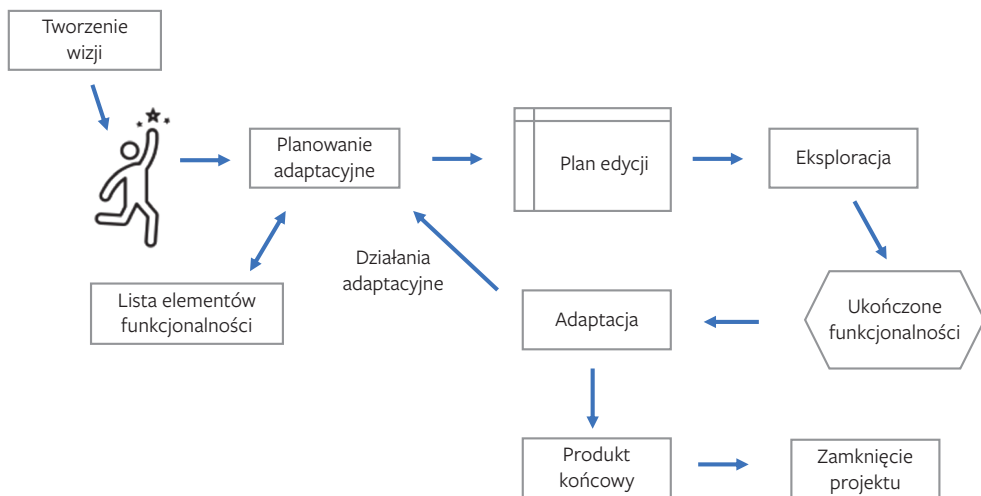
Struktura procesu APM składa się z pięciu faz omówionych poniżej¹¹⁷:

- 1) **Tworzenie wizji:** określenie wizji projektu, jego zakresu oraz zasad współpracy w ramach zespołu projektowego.
- 2) **Planowanie adaptacyjne:** wyspecyfikowanie elementów funkcjonalności dla produktu; stworzenie planu projektu i planów iteracji; zaplanowanie ograniczonych czasowo iteracji; szybki start.
- 3) **Eksploracja:** dostarczanie użytkownikowi elementów funkcjonalności; wdrożenie metod wytwarzania minimalizujących koszt zmian; stworzenie adaptowalnego, ko-laboracyjnego zespołu projektowego.
- 4) **Adaptacja:** ocena produktu, procesu, projektu i zespołu projektowego; korekta dotychczasowych planów i praktyk projektowych.

¹¹⁷ J. Highsmith, *Agile Project Management. Jak tworzyć...*, s. 95.

- 5) **Zamknięcie:** zamknięcie projektu, stworzenie bazy doświadczeń dla kolejnego projektu; celebrowanie.

Rysunek 15. Ramowa struktura zwinnego zarządzania projektem wg APM



Źródło: *ibid.*

Tworzenie wizji projektu ma na celu ukształtowanie ogólnego obrazu projektu oraz pozyskanie oczekiwań klienta. Głównymi pytaniami, przed którymi staje zespół projektowy są: „co?”, „kto?” i „jak?”. Odpowiedź na pierwsze pytanie pozwala określić z grubsza zakres projektu i podstawowe elementy jego funkcjonalności. Druga odpowiedź kształtuje skład zespołu projektowego oraz ma za zadanie zidentyfikować kluczowych interesariuszy projektu. Odpowiedź na pytanie „jak?” daje pojęcie o sposobie funkcjonowania i regułach współpracy w ramach projektu. Przyjęcie założeń dotyczących produktu, jak i zasad pracy zespołu traktowane jest jako jeden z kluczowych czynników sukcesu projektu adaptacyjnego. Według zaleceń autora, w fazie tworzenia wizji zespół projektowy powinien odnaleźć odpowiedzi na cztery pytania¹¹⁸:

- 1) Jaką wizję produktu ma klient?
- 2) Jaki jest zakres projektu i jego ograniczenia (w tym uzasadnienie biznesowe)?
- 3) Kim są uczestnicy odpowiedni do włączenia w wspólnotę projektową?
- 4) Jak zespół wytworzy projekt (podejście)?

W ramach prac przy tworzeniu wizji zespół projektowy może korzystać z siedmiu praktyk przywołanych przez autora metody:

¹¹⁸ *Ibid.*, s. 103.

Tablica 6. Praktyki APM w ramach fazy tworzenia wizji**Wizja produktu:**

- „pudełko produktu” i test windowy
- architektura produktu i zasady przewodnie

Zakres projektu (założone cele i ograniczenia):

- arkusz danych projektu

Wspólnota projektu:

- pozyskanie odpowiednich ludzi
- identyfikacja uczestników
- interfejs pomiędzy klientem a zespołem wytwórczym

Podejście:

- dopasowanie procesu i produktu.

Źródło: *ibid.*

Mimo, iż nie wynika to bezpośrednio ze schematu cyklu życia projektu, to istotnym zaleceniem metody odnośnie do wizji produktu jest powrót do niej na początku każdej iteracji (lub etapu). Ponowny przegląd wizji pozwala ją zmodyfikować, jak również przypomnieć zespołowi jej kształt i w konsekwencji cel, do którego dąży.

Faza planowania adaptacyjnego, nazywana także fazą spekulacji, ma na celu podjęcie próby sprecyzowania zakresu projektu pod kątem elementów funkcjonalności głównego produktu oraz zaplanowanie iteracji w projekcie. Zgodnie z podejściem APM projekt adaptacyjny musi być planowany, jednakże nacisk położony jest bardziej na czynność planowania niż powstające dokumenty. Plan ma wskazywać kierunek i pomagać w pracy zespołu, a nie być sprowadzony do formalnego biurokratycznego wymogu. Ponieważ planowanie projektu adaptacyjnego odbywa się w warunkach niepełnej informacji i niepewności staje się on bardziej przypuszczeniem co do przyszłości, niż jej sztywną projekcją, wymagającą bezwzględnego trzymania się opracowanych założeń. Stąd też słowo „spekulacja”, które zostało użyte celowo, aby podkreślić niepełność informacji i konieczność dochodzenia do pożądanych rozwiązań metodą małych kroków – kolejnych iteracji polegających na ciągłym powtarzaniu faz spekulacji, eksploracji i adaptacji. Zgodnie z zaleceniami metody nie ma konieczności wyczerpującego wczesnego planowania (tzw. *upfront planning*). Antycypacja przyszłości oraz wczesne estymaty obarczone są zbyt dużym ryzykiem, zaś rzetelne informacje pozyskane zostaną dopiero w trakcie trwania projektu. Stąd też duży nacisk na planowanie i replanowanie w trakcie projektu.

W fazie planowania adaptacyjnego zespół projektowy ma za zadanie¹¹⁹:

- 1) zebrać wstępne, szerokie wymagania produktu,
- 2) zdefiniować prace w postaci listy elementów funkcjonalności,

¹¹⁹ *Ibid.*, s. 96.

- 3) stworzyć plan dostarczania (edycja, etapy, iteracje), obejmujący harmonogram i przypisanie zasobów niezbędnych do stworzenia poszczególnych elementów funkcjonalności,
- 4) uwzględnić w planie strategię ograniczania ryzyka,
- 5) oszacować koszty projektu i wytworzenia pozostałych wymaganych informacji administracyjnych i finansowych.

Kluczowym efektem fazy planowania jest plan edycji projektu (ang. *release plan*). W przeciwieństwie do planów tradycyjnych nie przedstawia on zadań do wykonania, ale układ następujących po sobie iteracji wraz z przypisanymi do nich elementami funkcjonalności do dostarczenia. Realizacja projektu oparta na elementach funkcjonalności (ang. *feature-based development*) ma według autora charakter techniki uniwersalnej, stosowanej nie tylko przy projektach informatycznych. Wśród korzyści z nią związanych wskazuje m.in. na poprawę komunikacji między użytkownikami, a personelem wytwórczym, lepsze odwzorowanie postępu oraz możliwość priorytetyzowania wykonywanych prac. Zgodnie z zaleceniami metody możliwe jest przyjęcie różnych wariantów planów, w zależności od stopnia niepewności projektu¹²⁰:

- 1) kompletny plan, w którym elementy funkcjonalności są przypisane do wszystkich przewidzianych iteracji,
- 2) dwuczęściowy plan uwzględniający tylko następną iterację i całą resztę projektu,
- 3) planowanie z iteracji na iterację.

W ramach prac przy planowaniu adaptacyjnym zespół projektowy może korzystać z czterech praktyk przywołanych przez autora metody. Są nimi:

Tablica 7. Praktyki APM w ramach fazy planowania adaptacyjnego

Struktura podziału elementów funkcjonalności:

- lista elementów funkcjonalności,
- karty elementów funkcjonalności,
- karty wymagań wydajnościowych.

Planowanie projektu:

- plany projektu, kamieni milowych, iteracji.

Źródło: *ibid.*

W podsumowaniu etapu planowania warto zwrócić uwagę na fakt, iż mimo że APM koncentruje się na planowaniu zakresu i przebiegu projektu, to nie neguje konieczności opracowania innych planów towarzyszących, takich jak np. budżety, plany marketingowe itp. Są one opracowywane w zależności od potrzeb i wymagań organizacji, ale mają charakter raczej pomocniczy.

¹²⁰ *Ibid.*, s. 163.

Następująca po planowaniu adaptacyjnym faza eksploracji odpowiada tradycyjnym fazom wykonawstwa. Podczas fazy eksploracji tworzone są elementy funkcjonalności produktu, jego podsystemy. W tej fazie odbywa się właściwa praca przy projekcie: kodowanie systemów, opracowywanie nowych rozwiązań konstrukcyjnych, eksperymentalne zastosowania innowacyjnych technologii oraz testowanie gotowych rozwiązań.

Podczas fazy eksploracji zespół projektowy może korzystać z sześciu praktyk przywołanych przez autora metody. Są to:

Tablica 8. Praktyki APM w ramach fazy eksploracji

Prezentacja wizji i celów:

- zarządzanie pracą.

Praktyka techniczna:

- zmiany niskonakładowe.

Środowisko projektowe:

- coachowanie i budowanie zespołu,
- codzienne spotkania integracyjne zespołu,
- wspólne podejmowanie decyzji,
- codzienne interakcje z zespołem klienta.

Źródło: *ibid.*

Istotnym elementem fazy eksploracji jest stworzenie i utrzymanie przez kierownika projektu zespołu pracującego według omówionych wcześniej wartości i zasad adaptacyjnego zarządzania projektami.

Kolejną fazą projektu jest faza adaptacji, czyli dostosowania, modyfikacji. Skoro projekt wg APM rozwijany jest ewolucyjnie i adaptacyjnie, to do jego sukcesu konieczne jest częste uzyskiwanie informacji zwrotnej, co do poczynionych założeń, szacunków, ale także kierunku rozwoju projektu, elementów funkcjonalności, oczekiwań klientów itp. Stąd też tak istotna rola fazy, której celem jest ocena i dostosowanie¹²¹:

- 1) funkcjonalności projektu, głównie pod kątem zespołu klienta,
- 2) jakości produktu, głównie pod kątem zespołu technicznego,
- 3) wydajności i wyników zespołu,
- 4) statutu projektu.

Podobnie jak w przypadku „spekulacji”, tak i w tym przypadku użycie sformułowania „adaptacja” ma swój konkretny wydźwięk. Zastępuje ono tzw. „działania korygujące”, podejmowane w efekcie przeprowadzania kontroli projektu przez kierownictwo. Korekta oznacza zejście z określonej ścieżki postępowania (preskryptywnego planu), zaś adaptacja – odnosi się bardziej do dostosowania i ewolucji, tak aby dostarczać jak największą wartość dla interesariuszy projektu. Podjęcie działań adaptacyjnych pozwala

¹²¹ *Ibid.*, s. 219.

zespołowi na dopracowanie kształtu wytworzonego produktu, przetestowanie i zaproponowanie zmian, jak również uzyskanie od zlecającego informacji zwrotnej na temat opracowanego modułu wraz z zaleceniami do wykorzystania w przyszłych iteracjach. Wyniki otrzymane w tej fazie stanowią punkt wyjścia dla kolejnej pętli faz spekulacji – eksploracji – adaptacji.

W przeciwieństwie do pozostałych faz adaptacji przypisano tylko jedną praktykę, składającą się z kilku składowych.

Tablica 9. Praktyki APM w ramach fazy eksploracji

Przegląd produktu, projektu i zespołu oraz działania adaptacyjne:

- wywiady grupowe z użytkownikiem
- przeglądy techniczne
- oceny wydajności zespołu
- raporty o stanie projektu
- działania adaptacyjne

Źródło: *ibid.*

Ostatnia faza projektu to jego zamknięcie. Niestety, w praktyce jest często lekceważone. Wynika to najczęściej z chęci szybkiego uwolnienia i przerzucenia zasobów do innych działań organizacji. Zgodnie z modelem APM faza ta musi jednak zostać wykonana. J. Highsmith argumentuje to kilkoma punktami. Po pierwsze, jest to moment zakończenia prac wytwórczych w projekcie. Produkt osiągnął zamierzony kształt i realizuje określoną wizję – może zostać przekazany w całości użytkownikom, dla których jest to też istotny sygnał o jego przejściu w fazę eksploatacyjną. Po drugie, jest to okazja do zaznaczenia końca wysiłków zespołu projektowego oraz celebracji – co buduje tożsamość i morale zespołów, nie tylko adaptacyjnych. Po trzecie, odrębna faza zakończenia umożliwia podsumowanie i całościową ocenę projektu, uporządkowanie spraw administracyjnych, przygotowanie stosownych raportów oraz dokumentacji eksploatacyjnej. Po czwarte, zamknięcie projektu jest etapem, w którym należy w myśl APM poświęcić czas na tzw. retrospektywę projektu. Retrospektywa jest formą ewaluacji i oceny, ale z naciskiem na wyciągnięcie wniosków i doświadczeń projektowych na przyszłość. Służy ona zarówno przekazywaniu wiedzy w ramach zespołu, jak i na linii zespół – organizacja, czy zespół – inne zespoły.

W idealnej sytuacji do zamknięcia projektu można przejść, gdy w wyniku kolejnych pętli iteracyjnych klient uzna, iż komplet otrzymanych podsystemów funkcjonalnych spełnia jego oczekiwania i zgłoszone potrzeby. Oczywiście, istnieją również inne ewentualności, takie jak np. zaniechanie prac nad projektem w wyniku zaniku jego uzasadnienia biznesowego lub innej zmiany w otoczeniu projektu. W takim przypadku warto podkreślić, że ze względu na iteracyjny charakter pracy i angażowanie zespołu do realizacji pojedynczych,

kolejnych składowych przekazywanych bezzwłocznie klientowi koszt zaniechania projektu będzie relatywnie mniejszy niż przy wykorzystaniu metodyk tradycyjnych.

Ostatnia faza projektu APM jest jednocześnie jedyną praktyką wykonywaną na jego zakończenie.

2.2.3. Podsumowanie

W podsumowaniu i całościowej ocenie metodyki APM należy przede wszystkim zwrócić uwagę, iż jest to metodyka o charakterze ramowym. Jej autor pozostaje wierny przekonaniu, że zbyt szczegółowy rygor procesów utrudnia i obniża wydajność pracy. Model APM charakteryzuje się zatem wysokim poziomem ogólności i uniwersalności. Autor zawarł w metodyce fazy iteracyjnego wytwarzania produktów oraz opis wybranych praktyk. Zaproponowane rozwiązania stanowią jeden z przypadków w ramach rodziny zwinnych metodyk zarządzania projektami i są zbliżone do zaleceń innych popularnych metodyk.

Zdecydowanie jednak wartość modelu APM leży w jego uniwersalności. J. Highsmith zaproponował model zgodny z założeniami metodyk zwinnych, ale wykraczający znacznie poza projekty informatyczne, a przez to dający się zastosować w wielu innych obszarach działalności organizacji.

2.3. Programowanie ekstremalne – Extreme Programming (XP)

2.3.1. Geneza metodyki

Okoliczności powstania metodyki Extreme Programming (XP), zwanej po polsku programowaniem ekstremalnym, związane są z osobą urodzonego w 1961 r. w USA Kenta Becka. W 1996 r. został on zaproszony w charakterze konsultanta i lidera projektu do współpracy w mocno zagrożonym projekcie systemu płacowego budowanego w koncernie motoryzacyjnym Chrysler. Mimo, iż autorstwo XP przypisuje się właśnie Beckowi, to w ramach tegoż projektu zaangażowano także inne osoby, które wniosły swój znaczący wkład w stworzenie metodyki. Byli to: Ron Jeffries, Martin Fowler, Chet Hendrickson i Don Wells¹²². Beck korzystał także z wcześniejszych prac Warda Cunninghama. Większość z tych osób stało się później sygnatariuszami Manifestu zwinności. Na podstawie dramatycznej oceny sytuacji projektu zdecydowano się wstrzymać jego realizację i zacząć od nowa zgodnie z nowym, zaproponowanym przez współpracowników podejściem. Mimo że sam projekt Chyslera został wkrótce definitywnie zamknięty, to opracowana

¹²² <http://www.extremeprogramming.org> (dostęp: 24.11.2021).

metoda zarządzania projektem utrzymała się i była dalej rozwijana przez jej twórców. W 1999 r. Beck opublikował całościową metodykę w książce pt. *Extreme Programming Explained: Embrace Change*¹²³. Informację o XP znaleźć można także na takich portalach specjalistycznych, jak choćby extremeprogramming.org.

Metodyka XP ma przede wszystkim charakter metodyki wytwórczej. Zakres jej zaleceń skupia się na praktykach i działaniach związanych z programowaniem, stąd też jej zastosowanie przede wszystkim w projektach związanych z kodowaniem.¹²⁴ Istota metodyki związana jest z wartościami takimi jak: komunikacja, prostota, informacja zwrotna oraz odwaga. Podstawową zasadą jest wykonywanie wyłącznie takiej pracy, która jest niezbędna i konieczna w danym momencie oraz w sposób najprostszy z możliwych. Podejście to doprowadza programowanie do „ekstremum”, od czego wzięła nazwę omawiana metoda.

2.3.2. Struktura metodyki

Struktura metodyki obejmuje 12 kluczowych praktyk, opis ról w zespole projektowym, jak również iteracyjny model działań (procesów) ekstremalnego zarządzania projektem.

Do grona praktyk XP należą:

1) Zabawa w planowanie (ang. *planing games*)

Nazwa techniki podkreśla adaptacyjny charakter realizacji projektu, w którym plany mają charakter przewidywania i prognozy, a nie idealnego wzorca przebiegu niedopuszczającego odchyłeń i błędów. Punktem wyjścia w planowaniu w XP są historie użytkownika (ang. *user stories*) pochodzące od klienta, a następnie oszacowane pod kątem pracochłonności przez zespół. Na bazie tej listy planuje się liczbę, terminy i zakres poszczególnych wydań projektu (w pełni działających, kolejnych wersji rozbudowywanego produktu). Każde z wydań dzielone jest na krótkie, nie dłuższe niż dwutygodniowe, iteracje, podczas których trwają prace programistyczne i testowe. Same iteracje także na początku podlegają planowaniu przez zespół. Pod koniec iteracji zespół musi dostarczyć działający i w pełni przetestowany przyrost produktu. Praca w iteracjach umożliwia uczenie się i wyciąganie wniosków, a więc także modyfikację obu planów zgodnie ze zmianami pojawiającymi się w ramach oraz poza projektem.

2) Małe wydania

Praktyka małych wydań pozwala na częstą aktualizację tworzonego oprogramowania i rozbudowywanie go o nowe historie użytkowników. Małe wydania umożliwiają utrzymanie zaangażowania użytkowników, demonstracje postępów i widocznych

¹²³ K. Beck, *Extreme Programming Explained: Embrace Change*, Addison-Wesley Professional, Boston 1999; K. Beck, *Wydajne programowanie eXtreme Programming*, MIKOM, Warszawa 2001.

¹²⁴ K. Kaczor, *Scrum i nie tylko. Teoria i praktyka w metodach agile*, Wydawnictwo Naukowe PWN, Warszawa 2014, s. 63.

rezultatów prac oraz przekazanie zespołowi informacji zwrotnej od klienta wraz z akceptacją wykonanych prac.

3) **Metafora systemu**

Praktyka metafory systemu ma za zadanie stworzyć i utrzymać w zespole i u klienta spójną wizję budowanego produktu oraz usprawnić komunikację na tej linii. Metafora służy znalezieniu wspólnego słownictwa między klientem, a zespołem. Język ten będzie pochodną wiedzy i kompetencji technicznych po stronie klienta oraz rozumienia przez zespół sytuacji użytkowników. Posługiwanie się obrazami, symbolami i hasłami sprawi, iż obie strony będą wspólnie patrzeć na te same problemy i tak samo je rozumieć mimo różnych kompetencji fachowych.

4) **Prosty projekt**

Prostota jest jedną z kluczowych wartości XP, dlatego też cecha ta musi towarzyszyć także budowanym produktom. Metodyka zaleca, aby tworzone rozwiązania były tak proste, jak to tylko możliwe oraz dostarczone w jak najszybszym czasie. Sprzyja to postępowi projektu, ogranicza ryzyko błędów, obniża koszty utrzymania i późniejszego rozwoju oprogramowania.

5) **Ciągłe testowanie**

Ciągłe testowanie jest praktyką mającą zapewnić najwyższy poziom jakości budowanego rozwiązania, poprawność działania każdej budowanej funkcjonalności i szybkie informacje zwrotne ich dotyczące. Każdej historii użytkownika towarzyszyć powinien opis automatycznych testów akceptacyjnych. Dodatkowo, przed przystąpieniem do prac programiści przygotowują scenariusze testowe, które mają zostać ukończone przez tworzony przez nich kod. Dzięki tej praktyce, oprócz oceny kodu, programiści mogą skupić się na wykonaniu niezbędnej pracy, co ponownie oszczędza czas i wspiera prostotę rozwiązania.

6) **Ciągłe przerabianie kodu (ang. *refactoring*)**

Refaktoryzacja jest praktyką polegającą na takiej modyfikacji kodu, która, z jednej strony, znacznie upraszcza jego strukturę i treść, a z drugiej, zapewnia poprawność jego działania. Może ona polegać np. na usunięciu duplikowanych linii kodu, poprawianiu jego spójności, skracaniu, ograniczaniu zmiennych itp. Podobnie jak inne praktyki refaktoryzacja ma na celu zapewnienie prostoty i jak najlepszej jakości budowanego rozwiązania.

7) **Programowanie w parach**

Programowanie w parach jest chyba najbardziej charakterystyczną i rozpoznawalną praktyką XP. Polega ona na wspólnym pisaniu kodu przez dwóch współpracujących ze sobą programistów. W czasie gdy jedna osoba z pary tworzy kod, druga nie pozostaje bezczynna, lecz na bieżąco sprawdza go, wnosi swoje uwagi oraz wychwytuje błędy lub możliwości optymalizacji. Mimo pozornego marnowania połowy poten-

cjału zespołu, uzyskiwane wyniki są lepsze niż w przypadku odrębnej pracy obu osób. Zastosowanie programowania w parach podnosi jakość kodu, ogranicza ryzyka techniczne, a także zapewnia dzielenie się wiedzą oraz zespołowe uczenie się członków zespołu. Istnieją różne warianty tej techniki (np. programowanie „ramię w ramię”), jak również jej odmiany względem aktywności drugiej z osób (np. praca przy drugim monitorze przy pisaniu scenariuszy testowych).

8) **Standardy kodowania**

Podobnie jak w przypadku pracy przy edytorze tekstów, tak i w przypadku tworzenia oprogramowania istnieje poważne ryzyko, iż każda z osób piszących tekst, który ma zostać złożony w wspólną całość, będzie stosowała własne przyzwyczajenia, sposoby formatowania, wcięcia itp. Praktyka standardów kodowania ma zapewnić, iż cały zespół posługuje się jednymi, wspólnie ustalonymi zasadami dotyczącymi wyglądu kodu, opisu klas, komentarzy i innych jego elementów. Regulacje te powinny zostać spisane i utrwalone na papierze.

9) **Wspólna własność kodu**

W przeciwieństwie do metodyki FDD, metodyka XP przyjmuje za naczelną zasadę wspólną własność kodu. Każdy programista, w razie potrzeby, może wprowadzić zmiany w każdej części tworzonego oprogramowania. Dzięki temu jest ono poddane ocenie większej liczbie osób, które mogą zaproponować usprawnienia, poprawki i usunięcie błędów. Kolektywna własność zdejmuje z programistów dyskomfort związany z modyfikacjami efektów pracy swoich kolegów. Metodyka XP zaleca wręcz lokalizację całego zespołu blisko siebie, w jednym pomieszczeniu.

10) **Ciągła integracja**

Praktyka ciągłej integracji polega na ciągłym, automatycznym integrowaniu wprowadzanych zmian i nowych historii użytkownika do rozbudowywanego oprogramowania. Integracja ma na celu uzyskanie szybkiej informacji zwrotnej odnośnie do poprawności działania systemu po dodaniu nowych fragmentów kodu. Pozwala ona wychwycić i szybko rozwiązać konflikty wynikające z niezależnej modyfikacji tych samych fragmentów kodu przez różne osoby. Obniża ona także ryzyko techniczne, gdyż integracja dotyczy w efekcie małego przyrostu, a nie np. efektów dwutygodniowej pracy.

11) **40-godzinny tydzień pracy**

Ustalenie z góry i przestrzeganie czasu pracy zespołów programistycznych stanowi jeden z istotnych elementów metodyki XP. W celu utrzymania jakości pracy, poziomu intelektualnego, wydajności i morale zespołu praca w projekcie powinna być zorganizowana w sposób umożliwiający wywiązanie się z niej w 40-godzinny tydzień pracy. Praktyka ta ma z jednej strony dbać o zespół, z drugiej przyzwyczaić go do pracy w regularnym tempie i pod stałym obciążeniem. Wydajność zespołu ma

być utrzymywana na wysokim poziomie przez długi czas. Aby zapobiec wyczerpaniu, nadgodziny powinny być dopuszczane tylko w wyjątkowych przypadkach.

12) Ciągły kontakt z klientem

Ze względu na ewolucyjny i adaptacyjny charakter projektu pracy zespołu muszą towarzyszyć informacje zwrotne od klienta, która umożliwia korektę aktualnych planów i zamierzeń. Klient powinien być na równi z zespołem zaangażowany w projekt i mieć kompetencje do podejmowania decyzji odnośnie do jego zakresu i jakości. Często przedstawiciel klienta zostaje na co dzień oddelegowany do zespołu XP. Praktyka ta nie tylko usprawnia problemy operacyjne, ale także pozwala utrzymać relacje z klientem i unaocznic mu postępy i poziom skomplikowania realizowanego projektu.

Metodyka XP kładzie duży nacisk na współpracę i jakość pracy zespołu projektowego. Aby możliwe było stosowanie przedstawionych powyżej praktyk, każdy członek zespołu powinien posiadać doskonałe umiejętności fachowe oraz znać swoją rolę w większej całości. Zgodnie z zaleceniami metodyki w zespole XP można wyróżnić sześć ról.

Po pierwsze, są to programiści. Ich głównym zadaniem jest tworzenie oprogramowania, należy jednak pamiętać, iż stosowanie praktyk XP wymaga od nich także działań takich jak refaktoryzacja kodu, testowanie, udział w planowaniu i projektowaniu systemu. Praktyki takie jak wspólna własność kodu oraz ciągły kontakt z klientem powodują, iż umiejętnościom fachowym towarzyszyć powinny także umiejętności „miękkie”, a przede wszystkim dobra komunikacja i zdolność do pracy w grupie.

Po drugie, jest to śledzący (zarządca, ang. *tracker*). Jego rola polega na gromadzeniu danych i informacji powstających w trakcie pracy zespołu, dotyczących np. wydajności członków zespołu, liczby błędów, planowanych i rzeczywistych szacunków odnośnie do pracochłonności zadań itd. Jest to rola wspierająca, o charakterze administracyjnym. Efekty jego pracy mają umożliwić rejestrowanie postępów pracy oraz wyciąganie wniosków na bazie informacji historycznych.

Po trzecie: trener (ang. *coach*). Jest on odpowiedzialny za całość procesu budowania oprogramowania i zachowanie wytycznych metodyki XP. Ma on służyć zespołowi swoją wiedzą i doświadczeniem, tak aby mógł on sprawnie i bez przeszkód pracować nad projektem. Zadaniem trenera jest także rozwój zespołu i budowanie jego dojrzałości w pracy projektowej.

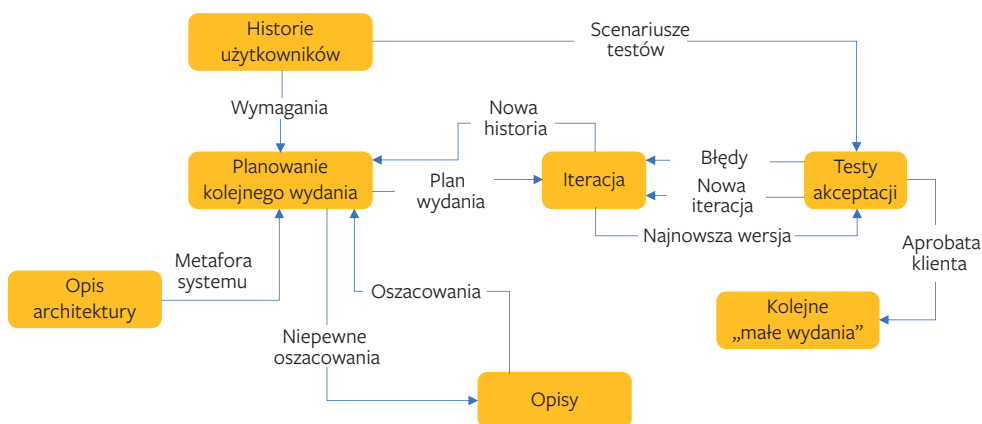
Czwartą rolę jest konsultant, który pomaga zespołowi przezwyciężyć problem, z którym jego członkowie nie są w stanie sobie poradzić. Konsultant przywoływany jest do zespołu w miarę potrzeb i nie musi być jego stałym członkiem.

Piątą rolę jest klient. Zgodnie z zaleceniami metodyki zespół utrzymuje stały kontakt z klientem. Zasilą on zespół opracowanymi historyjkami użytkownika, przygotowuje testy akceptacyjne, bierze udział w planowaniu wydań i iteracji, informuje o priorytetach i wartości dla użytkowników. Jest to rola o charakterze biznesowym.

Szóstą rolą jest szef (wielki szef, ang. *big boss*). Nadzoruje on pracę zespołu projektowego. Rola ta jest połączeniem sponsora/właściciela projektu oraz jego kierownika. Szef przede wszystkim rozlicza zespół z postawionych mu celów oraz uzyskiwanych efektów. Do jego obowiązków należy także zapewnienie zespołowi adekwatnych zasobów ludzkich oraz warunków pracy. Wśród cech oczekiwanych od szefa metodyka XP wyszczególnia: odwagę, pewność siebie, zdolność wpływania na zespół oraz uczciwość i szczerść w komunikacji.

Wartości, role i praktyki XP łączą się w procesie wytwarzania oprogramowania rekomendowanym przez twórcę metodyki.

Rysunek 16. Proces zarządzania projektami w metodyce XP



Źródło: K. Kaczmarek, *Extreme Programming – zapewnienie skutecznej i wydajnej pracy programistów*, <https://docplayer.pl/55642961-Extreme-programming-zapewnienie-skutecznej-i-wydajnej-pracy-programistow.html> (dostęp: 24.11.2021).

Proces realizacji projektu według XP rozpoczyna się od stworzenia przez klienta opisów historii użytkownika, które przedstawiają sposób działania budowanego systemu. Na ich podstawie tworzone są także automatyczne testy akceptacyjne, które zweryfikują poprawność ich implementacji w systemie. Historie użytkownika będą następnie szacowane pod kątem czasu ich realizacji oraz stanowić będą zasilenie do spotkania planistycznego poświęconego planowaniu wydania.

Podczas planowania wydania tworzony jest plan wydań funkcjonujący jako całościowy plan projektu. Lista historii użytkownika z informacjami dotyczącymi ich pracochłonności stanowi podstawę do określenia liczby, czasu i zakresu wydań oraz wchodzących w ich zakres iteracji. Miejsce poszczególnych historii użytkownika w planie zasadniczo zależy od ich wartości dla klienta. Jeśli w trakcie trwania projektu zajdą zasadnicze zmiany, planowanie wydania może zostać powtórzone, a sam plan zaktualizowany. W szczególności dotyczy to wydajności zespołu, która dyktuje tempo pracy i szybkość z jaką realizowane będą kolejne historie.

Następnie planuje się pierwszą iterację. Kolejne planowane będą w miarę postępów pracy, gdy nadejdzie czas na ich realizację. W trakcie planowania iteracji historie użytkowników rozbijane są na poszczególne zadania, szacowany jest czas potrzebny na ich wykonanie oraz tworzona odpowiednia sekwencja czynności. Przyjęcie planu iteracji umożliwia zespołowi przystąpienie do jego realizacji i wykonania. Podczas pracy w ramach iteracji zespół zobowiązany jest stosować przedstawione wcześniej praktyki XP, w tym programowanie w parach, refaktoryzację kodu oraz ciągłą integrację. Każdy dzień rozpoczynany jest wspólnym spotkaniem roboczym mającym na celu synchronizację pracy, przedstawienie planów na najbliższy dzień oraz podniesienie istotnych problemów. Wykonanie i pozytywne przetestowanie zakresu przewidzianego w danej iteracji pozwala rozpocząć planowanie kolejnej. Zakończenie tej fazy umożliwia wypuszczenie pierwszego i kolejnych wydań, aż do pomyślnego zakończenia całego projektu.

2.3.3. Podsumowanie

Metodyka Extreme Programming jest jedną z ciekawszych metodyk należących do grona zwinnych. Głównym obszarem jej zaleceń, zgodnie z nazwą, jest poziom wytwórczy. Jest to zatem metodyka developerska, którą zainteresowani będą przede wszystkim członkowie zespołów programistycznych. Do takich wniosków dojść można po zapoznaniu się z praktykami XP, spośród których wspólna własność kodu, programowanie w parach oraz 40-godzinny tydzień pracy mogą wzbudzać opór wśród kierownictwa organizacji. Wdrożenie zasad metodyki zdecydowanie zmienia układ sił wokół projektu na rzecz oddania programistom władzy i kompetencji do sterowania nim. Działanie takie, zdaniem autora XP, przynosi pozytywne efekty. Przestrzeganie zaleceń pozwala tworzyć produkty najwyższej jakości: o prostym kodzie, stale integrowane i w pełni działające. Zespół, równomiernie zaangażowany w pracę, przez cały czas koncentruje się na zadaniach, które tworzą wartość dla klienta i są dla niego istotne, a sam klient stale obserwuje rozwój produktu i przekazuje informacje zwrotne ukierunkowujące pracę.

XP zdaje się być wyśmienitym rozwiązaniem dla zespołów programistycznych, jednakże jego wartość bywa kwestionowana. W źródłach fachowych spotkać można także i negatywne głosy dotyczące XP¹²⁵. Wśród argumentów krytycznych zdarzają się zarzuty: począwszy od fiaska projektu Chryslera, od którego XP wzięło swój początek, poprzez krytykę programowania w parach, ograniczonej dokumentacji projektu, znacznego, a często przesadnego wysiłku wkładanego w refaktoryzację, trudności w praktycznym pozyskaniu i zaangażowaniu właściwej osoby jako przedstawiciela klienta, braku ska-

¹²⁵ Liczny zbiór artykułów krytycznych wobec XP można znaleźć m.in. na stronie: www.softwarereality.com/lifecycle/xp/links.jsp (dostęp: 24.11.2021).

lowalności, zbytniego skupienia się na kodzie kosztem działań towarzyszących i wymagań niefunkcjonalnych, aż po marginalizację działań związanych z zarządzaniem projektem biznesowym.

2.4. Crystal Light

2.4.1. Geneza metodyki

Crystal Light jest nie jedną, ale grupą metodyk zaproponowanych przez Alistaira Cockburna. Urodził się on w 1953 r., 22 lata później uzyskał magistra w zakresie nauk informatycznych na Uniwersytecie w Cleveland. W 2003 r. otrzymał tytuł doktora na Uniwersytecie w Oslo. Swoją karierę w branży ICT rozpoczynał w drugiej połowie lat 80. XX w. w koncernie IBM. Jest jednym z aktywnych członków ruchu zwinnego wytwarzania oprogramowania oraz współautorem Manifestu zwinności, jak również PM Declaration of Independence.

Podobnie jak światło, przechodząc przez pryzmat, rozszczepia się na paletę kolorów, tak i rodzina Crystal Light składa się z całej gamy metodyk oznaczanych różnymi kolorami. Ich dostępność ma umożliwić zarządzającym dobór rozwiązania najlepiej oddającego specyfikę realizowanego projektu. Autor uzależnił powyższy wybór od czterech parametrów opisujących potencjalne skutki porażki projektu lub budowanego systemu. Są to: utrata komfortu (ang. *comfort* – *C*), strata bieżących środków pieniężnych (ang. *discretionary money* – *D*), utrata aktywów (ang. *essential moneys* – *E*) oraz utrata życia (ang. *life* – *L*)¹²⁶.

Stopień intensywności koloru odpowiada „ciężkości” metodyki. W palecie metodyk Crystal Light znaleźć można wersje metodyki takie jak: Clear, Yellow, Orange oraz Red¹²⁷. Ich liczba odpowiadała zamierzeniom zbudowania szerszego systemu, jednak w praktyce Cockburn przedstawił w szczegółach jedynie Crystal Clear, zaś Crystal Orange doczekało się jedynie zgrubnego opisu. Informacje dotyczące pierwszej z nich autor zawarł w książce swojego autorstwa pt. *Crystal Clear. A Human-Powered Methodology For Small Teams* wydanej w 2004 r. przez wydawnictwo Addison-Wesley, zaś druga opisana została w jednym z rozdziałów jego książki pt. *Surviving Object-Oriented Projects* opublikowanej siedem lat wcześniej. Wśród innych źródeł wiedzy o metodyce warto wskazać stronę internetową autora (<http://alistair.cockburn.us/Crystal+light+methods>).

¹²⁶ A. Cockburn: *Crystal Clear: A Human-Powered Methodology for Small Teams*, Addison-Wesley, London 2004, s. 240.

¹²⁷ W niektórych źródłach spotkać można także warianty takie jak: *Crystal Orange We*, *Crystal Maroon*, *Crystal Diamond*, *Crystal Sapphire*, *Crystal Blue*, por. <http://www.scrumstudy.com/blog/what-is-crystal/> (dostęp: 24.11.2021).

Rysunek 17. Dobór metodyk z rodziny CL względem typu projektu

	L6	L20	L40	L80
E6	E6	E20	E40	E80
D6	D6	D20	D40	D80
C6	C6	C20	C40	C80
Czysta	Czysta	Żółty	Pomarańczowy	Czerwony

Źródło: A. Cockburn: *Crystal Clear: A Human-Powered Methodology for Small Teams*, Addison-Wesley, London 2004, s. 240.

2.4.2. Założenia metodyki

W intencji autora metodyka Crystal Clear (CC) jest metodyką lekką, opracowaną na potrzeby małych liczących od 6 do 8 osób, zlokalizowanych w jednym miejscu zespołów pracujących nad systemami, których ewentualne błędy nie zagrażają ludzkiemu życiu. Zespołów, które „dokonują priorytetyzacji dla bezpieczeństwa dostarczenia satysfakcjonującego rezultatu, wydajności w budowaniu rozwiązania oraz dogodności konwencji pracy”¹²⁸. W przeciwieństwie do innych metodyk zwinnych (jak np. Extreme Programming) metodyka CC koncentruje się przede wszystkim na ludziach i zespole, a nie dyscyplinie w praktykach i procesach realizacji projektu. Sam autor, podsumowując założenia swojej metody opisał ją słowami:

Projektant wiodący, wraz z dwoma – ośmioma programistami,
w dużym pokoju lub w przylegających do siebie pokojach,
stosując tablice, ekrany i arkusze papierów
mając łatwy dostęp do ekspertów,
nie będąc narażonym na zakłócenia pracy,
regularnie, co miesiąc lub dwa (najpóźniej co kwartał)
dostarczają użytkownikom działający, przetestowany, użyteczny kod,
okresowo dokonując oceny i refleksji oraz modyfikując swoje praktyki pracy¹²⁹.

¹²⁸ A. Cockburn, *Crystal Clear...*, s. 312.

¹²⁹ *Ibid.*

Metodyka CC koncentruje się na elementach takich jak: stosowanie przyrostowych, iteracyjnych modeli cyklu życia, komunikacja i współpraca między członkami zespołu, swoboda w zakresie przyjmowanych przez zespół praktyk realizacji projektu. W przeciwieństwie do metodyki XP, Crystal Clear sukces projektu opiera nie na przestrzeganiu praktyk, ale na maksymalizacji indywidualnego potencjału członków zespołu.

2.4.3. Struktura metodyki

Podstawowe zalecenia metodyki Crystal Clear opierają się na opracowanej przez autora liście siedmiu wymaganych priorytetów. Pierwsze trzy mają charakter obligatoryjny, kolejne wspierają poziom bezpieczeństwa zespołu. Są to:

- 1) częste dostawy (ang. *frequent delivery*),
- 2) refleksyjne usprawnienia (ang. *reflective improvement*),
- 3) osmotyczna komunikacja (ang. *osmotic communication*),
- 4) bezpieczeństwo osobiste (ang. *personal safety*),
- 5) skupienie (ang. *focus*),
- 6) wczesny dostęp do ekspertów użytkownika,
- 7) środowisko techniczne z zautomatyzowanymi testami, zarządzaniem konfiguracją i częstą integracją.

Zespoły posługujące się metodyką CC mają swobodę określenia praktyk, którymi będą się posługiwać. CC nie wymaga stosowania żadnej techniki lub narzędzi, jednakże dla wsparcia zespołów rozpoczynających swoją pracę autor przytacza w podręczniku zestaw pięciu strategii oraz dziewięciu technik. Mimo deklaracji nienarzucania ich stosowania, są one w praktyce ważnym elementem CC. Są to:¹³⁰

Strategie

- 1) Rozpoznanie 360 stopni – wraz z rozpoczęciem projektu zespół powinien kompleksowo rozpoznać elementy takie jak: wartość biznesowa, wymagania, model domowy, plany technologii, plan projektu, tworzenie zespołu, proces realizacji projektu.
- 2) Wczesne zwycięstwa – poszukiwanie możliwości szybkiego dostarczenia wartości użytkownikom, budujące siłę, pewność i poczucie wartości zespołu.
- 3) Chodzący szkielet (ang. *walking skeleton*) – małe wdrożenie systemu wykonujące niewielki, ale w całości opracowany zakres funkcji, obrazujące główne elementy architektury.

¹³⁰ Ibid., s. 63–129.

- 4) Przyrostowa reakchitektura – modyfikowanie przez zespół infrastruktury lub architektury systemu wraz z jego rozbudową o nowe funkcjonalności.
- 5) Tablice informacyjne – dobrze widoczne i czytelne plansze lub ekrany obrazujące postęp projektu i jego najważniejsze parametry; tablice informacyjne powinny być aktualne.

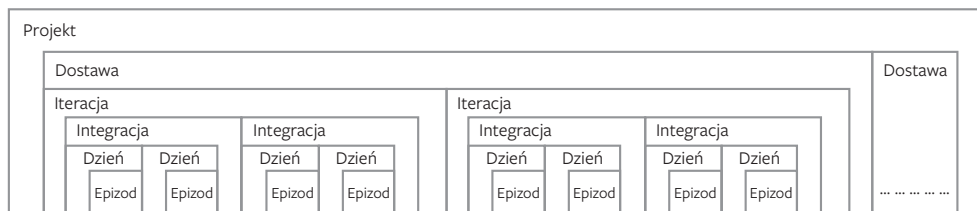
Techniki

- 1) Kształtowanie metodyki – gromadzenie informacji o wcześniejszych doświadczeniach oraz kształtowanie procesu i praktyk realizacji projektu (wywiady i warsztat metodyczny).
- 2) Warsztaty refleksyjne – warsztat organizowany po każdej dostawie, wspierający priorytet „refleksyjne usprawnienia”; jego celem jest zidentyfikowanie przez zespół problemów, jak również praktyk, które powinny zostać zachowane oraz wprowadzone jako nowe do pracy zespołu.
- 3) Błyskawiczne planowanie (ang. *blitz planning*) – technika dla stworzenia planu rozwoju projektu składająca się z dziesięciu kroków postępowania, tj.: zgromadzenia uczestników (sponsor, przedstawiciel użytkownika, programiści), burzy mózgów na temat zadań, strukturalizacji zadań, przeglądu zadań, szacowania zadań, ustalenia kolejności zadań, wskazania chodzącego szkieletu, identyfikacji kolejnych wydań, optymalizacji planu, dokumentacji planu.
- 4) Szacowanie delfickie z rankingami ekspertów – technika szacowania czasu i nakładów pracy zalecana podczas planowania projektu.
- 5) Codzienne spotkania „na stojąco” – krótkie spotkania robocze projektu.
- 6) Projektowanie kluczowych interakcji (ang. *essential interaction design*) – technika warsztatowa pozwalająca w szybki sposób odkryć wymagania poprzez identyfikację ról użytkowników, ich celów oraz wzajemnych interakcji zachodzących w systemie.
- 7) Miniatura procesu (ang. *process miniature*) – szybkie przeszkolenie pracowników w zakresie procesów poprzez warsztaty mające na celu przyśpieszoną symulację procesu (prześledzenie przebiegu, uczestniczenie w nim, odegranie kroków postępowania).
- 8) Programowanie „ramię w ramię” – modyfikacja techniki programowania w parach, polegająca na pracy dwóch programistów obok siebie, przy osobnych stanowiskach, ale wspólnie doglądających swojej pracy.
- 9) Wykresy spalania – sposób ilustracji postępów projektu oraz tempa pracy; wykres, gdzie na osi odciętych mierzony jest czas iteracji, zaś na osi rzędnych – zakres prac do wykonania; informacje na wykresach mogą obrazować planowane postępy pracy oraz faktycznie osiągnięte, a także ułatwiają prognozowanie przy ekstrapolacji trendów historycznej wydajności zespołów.

Cykl życia projektu

Cykl życia projektu rekomendowany w CC opiera się *de facto* na kilku poziomach hierarchii. Pierwszy poziom to poziom projektu. Projekt rozpoczyna się od ustanowienia projektu (ang. *charter*), poprzez dwa lub więcej cykle wytwórcze, aż do jego zakończenia (ang. *wrapup*). Schematyczny układ cyklu życia projektu w CC przedstawiono na poniższym schemacie:

Rysunek 18. Schemat cyklu życia projektu w metodyce CC



Źródło: *ibid.*

Ustanawiając projekt należy zbudować zespół projektu, wykonać rozpoznanie 360 stopni, dopasować metodykę oraz opracować wstępny plan projektu.

W skład poszczególnych cykli wytwórczych wchodzi zadania takie jak: rekalkulacja planu wydań, seria iteracji dostarczających działający i przetestowany kod, przekazanie rezultatu użytkownikom oraz podsumowanie cyklu z refleksją odnośnie do efektów i sposobu pracy.

W ramach poszczególnych iteracji zespoły powinny wykonać przewidziane prace programistyczne wraz z integracją oraz podsumować iterację poprzez warsztat refleksyjny oraz celebrację.

Najniższym poziomem modelu są poszczególne tygodnie oraz dni robocze. W ciągu dnia członkowie zespołu spotykają się na „na stojąco”, by następnie przejść do pracy przy tzw. epizodach, czyli podstawowych jednostkach pracy programistycznej wg CC.

Role

Ostatnim elementem rekomendacji metodyki Crystal Clear są role przypisane poszczególnym członkom zespołu. Alister Cockburn zdefiniował ich osiem. Są to:

- 1) sponsor wykonawczy (ang. *executive sponsor*) – odpowiedzialny za stronę biznesową przedsięwzięcia, zapewnienie zasobów do jego realizacji, podejmowanie strategicznych decyzji dotyczących relacji wymiennych między parametrami projektu,

- 2) ambasador użytkownika (ang. *ambassador user*) – osoba reprezentująca grono interesariuszy po stronie klienta/użytkowników budowanego systemu; ma za zadanie przekazywać zespołowi informacje dotyczące ich wymagań, priorytetów, sposobu korzystania z systemu itp.; jest to osoba doskonale orientująca się w oczekiwaniach grupy użytkowników i występująca w ich imieniu,
- 3) projektant wiodący (ang. *lead designer*) – doświadczony ekspert od strony technicznej budowanego rozwiązania odgrywający kluczową rolę podczas określania metodyki pracy zespołu oraz projektowania i programowania rozwiązania,
- 4) projektant-programista (ang. *designer-programmer*) – członkowie zespołu zajmujący się projektowaniem i programowaniem budowanego rozwiązania,
- 5) ekspert biznesowy (ang. *business expert*) – osoba dysponująca wiedzą i doświadczeniem odnośnie do sposobu funkcjonowania budowanego rozwiązania; wspierająca pozostałe role w zespole,
- 6) koordynator (ang. *coordinator*) – odpowiednik kierownika projektu, osoba dokumentująca plany projektu, rejestrująca jego postępy i koordynująca pracę; często wspierająca sponsora i wiodącego projektanta,
- 7) tester i pisarz (ang. *writer*) – często role rotacyjne i tymczasowe służące zapewnieniu adekwatnego poziomu testowania i dokumentowania budowanego systemu.

2.5. Feature Driven Development (FDD)

Podobnie jak inne metodyki z rodziny zwinnych także Feature Driven Development (FDD) swoje powstanie i popularyzację zawdzięcza ponadprzeciętnym jednostkom i ich wizji nowego sposobu realizacji projektów informatycznych. W przypadku FDD był to Jeff De Luca. Urodził się on w 1964 r. w Australii gdzie, między 1981, a 1993 r. pracował w oddziale IBM w Melbourne. W tym czasie przeszedł kolejne szczeble kariery od pracownika kancelarii poczty wewnętrznej do *senior system strategist* zaangażowanego w kluczowe projekty przedsiębiorstwa. Po odejściu z IBM De Luca założył własną firmę informatyczną – Nebulon Pty Ltd – zajmującą się realizacją złożonych systemów informatycznych.

W 1997 r., podczas realizacji znacznego projektu rozwoju oprogramowania dla jednego z banków w Singapurze De Luca wykorzystał swoje prawie 30-letnie doświadczenie w zarządzaniu projektami do stworzenia innowacyjnej metody, którą nazwał Feature Driven Development, czyli „rozwojem opartym na funkcjonalnościach”. Tworząc nową metodykę, zaimplementował w niej rozwiązania charakterystyczne dla podejść zwinnych, w tym opracowaną w 1997 r. przez Petera Coda koncepcję funkcjonalności. Metoda opisana została po raz pierwszy dwa lata później, w jednym z rozdziałów książki

pt. *Java Modeling in Color with UML*. Aktualnie, najnowsza wersja metody prezentowana jest m.in. na stronie internetowej autora, na portalu społeczności FDD (www.featuredrivendevelopment.com), publikacji poświęconej FDD pt. *A Practical Guide to Feature-Driven Development* autorstwa Stephena R. Palmera i Johna Mac Felsinga¹³¹ oraz w licznych innych źródłach internetowych.

Istota podejścia FDD opiera się na przedstawieniu zakresu projektu w rozbiciu na drobne elementy, wartościowych dla klienta funkcji, nazywanych przez De Lucę i Coadą funkcjonalnościami. Identyfikacja funkcjonalności pozwala następnie realizować projekt iteracyjnie, demonstrując w krótkich odstępach czasu materialne, działające wyniki prac. FDD umożliwia ponadto szczegółowe planowanie, dokładny pomiar postępów realizacji projektów oraz szybką prognozę szans powodzenia projektu nawet już po 10% zaawansowania prac¹³². Co ważne, FDD odróżnia się od innych zwinnych metodyk zarządzania projektami możliwością skalowania do dużych, zespołów przez co stanowi wartościowe rozwiązanie w przypadku obszernych projektów informatycznych.

2.5.1. Struktura i zastosowanie FDD

Struktura modelu zaproponowanego w FDD oparta jest na pięciu procesach, z których pierwsze trzy przynależą do fazy rozpoczęcia prac, zaś dwa kolejne do etapu budowania rozwiązań. Poza tym w realizację przedsięwzięcia zaangażowanych jest sześć głównych ról. Metodyka FDD rekomenduje także wykorzystanie serii najlepszych praktyk tworzenia systemów informatycznych.

Role w FDD¹³³:

- 1) *project manager* (kierownik projektu) – osoba odpowiedzialna za administrowanie projektem czyli czynności związane z raportowaniem postępów, kontrolą budżetu, organizacją pracy zespołu, zapewnieniem mu adekwatnych warunków do funkcjonowania,
- 2) *chief architect* (główny architekt) – w jego kompetencjach leży całościowy projekt systemu oraz wszelkie techniczne aspekty projektu; jest on odpowiedzialny m.in. za prowadzenie warsztatów projektowania (ang. *workshop design sessions*) podczas których zespół wspólnymi siłami projektuje budowany system; rola ta jest kluczowa dla

¹³¹ S.R. Palmer, J.M. Felsing, *A Practical Guide to Feature-Driven Development*, Prentice Hall, Hoboken 2002.

¹³² <http://www.nebulon.com/fdd/index.html> (dostęp: 24.11.2021).

¹³³ Oprac. na podst.: S. Goyal, *Major Seminar On Feature Driven Development. Agile Techniques for Project Management and Software Engineering*, https://www.academia.edu/31885026/Major_Seminar_On_Feature_Driven_Development_Agile_Techniques_for_Project_Management_and_Software_Engineering (dostęp: 24.11.2021) oraz <http://www.step-10.com/SoftwareProcess/FeatureDrivenDevelopment/FDDPeople.html> (dostęp: 12.03.2020).

powodzenia projektu i wymaga zarówno umiejętności i doświadczenia specjalistycznego, jak i zdolności miękkich,

- 3) *development manager* (kierownik rozwoju) – odpowiedzialny za prowadzenie bieżących, codziennych prac rozwojowych oraz wsparcie zespołu, w tym np. rozstrzyganie konfliktów między głównymi programistami,
- 4) *chief programmers* (główni programiści) – doświadczeni programiści uczestniczący w wysokopoziomowej analizie wymagań oraz planowaniu działań w projekcie; kierują pracą podległych im małych zespołów programistycznych dostarczających nowe funkcjonalności,
- 5) *class owners* (właściciele klas) – programiści, członkowie zespołów kierowanych przez głównych programistów, zajmujący się projektowaniem, kodowaniem, testowaniem i dokumentowaniem funkcjonalności wchodzących w skład tworzonego oprogramowania,
- 6) *domain experts* (ekspersi obszarowi) – pozostali, najczęściej nietechniczni, interesariusze, którzy współpracują z zespołem przekazując informacje konieczne do zapewnienia sukcesu projektu oraz jego zgodności z oczekiwaniami; najczęściej do tego grona wchodzi użytkownicy, sponsorzy, analitycy biznesowi i inni.

Metodyka FDD rekomenduje także dodatkowe role wspierające, takie jak: kierownik wydań (*release manager*), językowy guru (ang. *language guru*, *language lawyer*), inżynier budowy (ang. *build engineer*), ekspert ds. narzędzi (ang. *toolsmith*), administrator systemu (ang. *system administrator*) oraz oczywiście role takie jak testerzy, wdrożeniowcy oraz twórcy dokumentacji (ang. *technical writers*)¹³⁴.

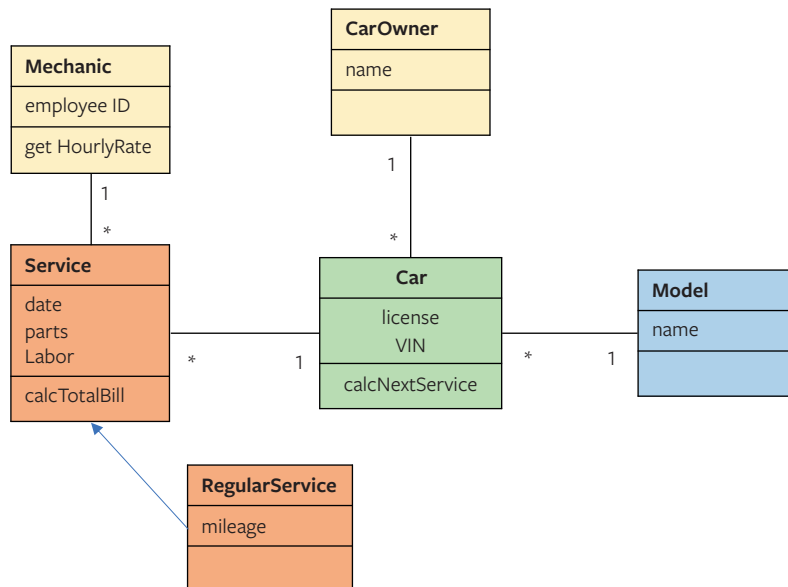
Praktyki FDD¹³⁵

Wśród zaleceń metodyki FDD znaleźć można osiem najlepszych praktyk stosowanych w trakcie planowania i realizacji projektu. Są to:

- 1) Modelowanie obiektów domeny (ang. *domain object modeling*) – praktyka polegająca na dokładnym rozpoznaniu i opisaniu rozpatrywanego problemu poprzez tworzenie diagramów klas opartych na istotnych obiektach i ich relacjach w ramach rozpatrywanego problemu (domeny); dzięki tej praktyce problem rozbijany jest na części składowe, które następnie rozwiązywane są już jako mniejsze zagadnienia, których rozwiązania tworzą wspólnie oczekiwany rezultat; w FDD zaleca się stosowanie „modelowania w kolorze” (ang. *modeling in color*), czyli porządkowania klas za pomocą różnokolorowych kategorii.

¹³⁴ <http://www.step-10.com/SoftwareProcess/FeatureDrivenDevelopment/FDDPeople.html> (dostęp: 24.11.2021).

¹³⁵ Oprac. na podst.: S. Goyal, *Major Seminar...* oraz <http://www.step-10.com/SoftwareProcess/FeatureDrivenDevelopment/FDDPeople.html> (dostęp: 12.03.2020).

Rysunek 19. Modelowanie obiektów domeny

Źródło: S. Goyal, *Major Seminar On Feature Driven Development. Agile Techniques for Project Management and Software Engineering*, https://www.academia.edu/31885026/Major_Seminar_On_Feature_Driven_Development_Agile_Techniques_for_Project_Management_and_Software_Engineering (dostęp: 24.11.2021).

- 2) Rozwój oparty na funkcjonalnościach (ang. *developing by feature*) – praktyka stanowiąca istotę metodyki FDD polegająca na kierowaniu i śledzeniu rozwoju produktu poprzez strukturę rozbitych, małych funkcji tworzących wartość dla klienta (tzw. funkcjonalności); identyfikacja funkcjonalności z perspektywy użytkowników i klientów pozwala programistom budować system zgodnie z oczekiwaniami zamawiającego, jak również prezentować postęp projektu w zrozumiałej dla niego formie; ponieważ system informatyczny zwykle ma za zadanie wspierać procesy biznesowe to funkcjonalności mogą być kojarzone z poszczególnymi działaniami w takich procesach; funkcjonalności w FDD opisywane są w formie <działanie> <wynik> <obiekt>, np. „uzyskaj saldo konta klienta” albo „autoryzuj transakcję kartą kredytową jej właściciela”; zgodnie z zaleceniami metodyki poszczególne funkcjonalności powinny być małe, tak aby ich realizacja zajęła nie więcej niż dwa tygodnie, wymaga to czasem dalszego rozbicia większych funkcjonalności na elementy niższego rzędu.
- 3) Indywidualna własność klas (kodu) (ang. *class/code ownership*) – w przeciwieństwie do niektórych metodyk zwinnych FDD stosuje zasadę indywidualnej własności kodu; podczas rozwoju produktu programistom przypisuje się odpowiedzialność za zestaw klas w ramach rozpatrywanego modelu domeny; osoby te stają się w całości odpowiedzialne za elementy kodu związane z ich obszarem, stają się ekspertami mającymi

zapewnić spójność, poprawność działania i wydajność każdej z klas oraz służyć innym pomocą i swoją wiedzą w tym zakresie.

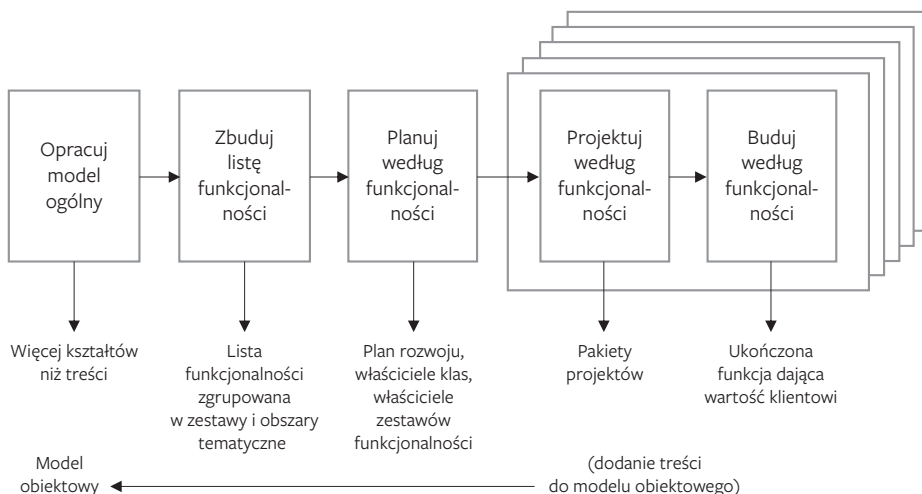
- 4) Zespoły funkcjonalnościowe (ang. *feature teams*) – struktura podzespołów w FDD ma charakter dynamiczny i zorientowany na dostarczanie funkcjonalności; odpowiedzialność za dostarczenie funkcjonalności ponoszą liderzy podzespołów, którzy zgodnie z potrzebami angażują poszczególnych programistów (właścicieli klas) do pracy nad nimi; w efekcie zespoły funkcjonalnościowe są małe (3–6 osób) i składają się z wszystkich programistów, których dotyczyć będą zmiany i praca konieczna do wykonania w celu wprowadzenia nowych funkcjonalności; zespoły te mają charakter macierzowy, zaś poszczególni programiści często przynależą do kilku z nich.
- 5) Inspekcje (ang. *inspection*) – według zaleceń metodyki FDD tworzone oprogramowanie oraz poszczególne funkcjonalności powinny przechodzić inspekcje; ich celem jest weryfikacja poprawności, wychwycenie błędów, jak również zespołowe uczenie się i transfer wiedzy, jak też ustalenie oraz zapewnienie przestrzegania zasad i standardów tworzenia oprogramowania; ponieważ inspekcje dotyczą efektów pracy zespołów funkcjonalnościowych ocena nie dotyczy jednostek, ale rezultatów pracy zespołowej; zgodnie z deklaracjami autora FDD sprzyja to atmosferze inspekcji oraz obniża stres z tym związany.
- 6) Regularne wydania (ang. *regular bid schedule*) – istotą regularnych wydań jest całościowe wykonywanie powierzanych celów w regularnych odstępach czasu, tj. ukończenie kodu, bibliotek i komponentów dla wszystkich przewidzianych funkcjonalności w ramach bieżącego przyrostu; metodyka nie reguluje tempa regularnych wydań pozostawiając to do decyzji zespołu; regularne wydania nie tylko umożliwiają przetestowanie spójności i integralności budowanych rozwiązań częściowych, ale także pozwalają wcześniej demonstrować klientowi częściowo gotowy produkt oraz używać z niego chociaż częściowe korzyści.
- 7) Zarządzanie konfiguracją (ang. *configuration management*) – zarządzanie konfiguracją jest praktyką mającą na celu kontrolowanie wersji, czyli identyfikowanie i śledzenie stanów rezultatów prac zespołów (kolejnych wydań, rozbudowywanych funkcjonalności, wprowadzanych zmian); o ile zarządzanie konfiguracją dotyczy przede wszystkim tworzonego kodu, to należy także pamiętać, iż zmianom i wersjonowaniu podlega będą także inne składowe projektu, takie jak: dokumentacja techniczna, dokumentacja zarządcza, środowisko testowe, wdrożenia u klienta oraz inne; zespół w każdym momencie powinien być w stanie wskazać najnowszą wersję rezultatów oraz zakres wprowadzonych historycznych zmian.
- 8) Raportowanie postępów (ang. *progress reporting*) – praktyka zapewniająca częste, dokładne i adekwatne raportowanie postępów projektu na wszystkich poziomach, zarówno na potrzeby interesariuszy wewnętrznych, jak i zewnętrznych; dostęp do

bieżących i aktualnych danych o stanie projektu jest konieczny szefom i liderom zespołów w celu podejmowania decyzji i kierowania rozwojem projektu; w zakresie śledzenia postępów technika FDD posługuje się szczegółowymi zaleceniami dotyczącymi pomiaru postępów oraz sposobu ich prezentacji – zostaną one przedstawione w dalszej części opracowania.

Proces FDD

Role i praktyki metodyki FDD stanowią jej istotny zakres, jednak główną osią realizacji projektu są oczywiście procesy. Działania związane z tworzeniem oprogramowania przypisane zostały do pięciu procesów według poniższego schematu:

Rysunek 20. Proces FDD



Źródło: <http://www.step-10.com/SoftwareProcess/FeatureDrivenDevelopment/index.html> (dostęp: 24.11.2021).

Przystępując do projektu, zespół w pierwszym kroku ma za zadanie stworzyć całościowy model (ang. *develop an overall model*), na podstawie którego tworzona jest lista funkcjonalności oraz zgrubny plan działania. Następnie zespoły przystępują do pracy, budując system poprzez projektowanie i dostarczanie jego funkcjonalności.

Proces 1. Opracowanie całościowego modelu

Proces ten ma charakter inicjujący. Warunkiem jego rozpoczęcia jest wskazanie ekspertów domenowych, głównego architekta oraz głównych programistów. Przewodnią rolę w procesie odgrywa główny architekt, którego zadaniem jest pokierować zespołem

w celu dokładnego omówienia zakresu tworzonego systemu, wskazania jego domen (obszarów) oraz ich zamodelowania. Tworzenie modeli domen odbywa się warsztatowo, przy współpracy ekspertów z programistami. Stworzone modele są następnie prezentowane i dyskutowane na forum całego zespołu. Zatwierdzenie ostatecznego kształtu modeli domen umożliwia opracowanie na tej podstawie modelu całościowego, będącego podstawą do dalszej pracy.

Pierwszy proces i jednocześnie etap projektu gwarantuje, że przed przystąpieniem do pracy zespół podejmie adekwatny wysiłek rozpoznania struktury problemu i stworzy jego model koncepcyjny. Poziom szczegółowości powinien być właściwy, tzn. odchodzący od podejścia BDUF (ang. *big design up-front*) na rzecz JEDI (ang. *just enough design initially*).

Proces 2. Opracowanie listy funkcjonalności

Kolejnym krokiem projektu jest, zgodnie z FDD, przystąpienie zespołu (zazwyczaj składającego się na tym etapie tylko z głównych programistów) do funkcjonalnego podziału domen na obszary tematyczne, aktywności biznesowe (a w ramach nich na poszczególne kroki), dochodząc finalnie do listy funkcjonalności. Tak więc, w przeciwieństwie do innych metodyk zwinnych opierających się na funkcjonalnościach (np. SCRUM), metodyka FDD buduje ich trzypoziomową strukturę (zamiast prostej listy). Etap uznaje się za zakończony, gdy lista ta spełnia oczekiwania kierownika projektu oraz kierownika rozwoju.

Proces 3. Planowanie oparte na funkcjonalnościach

Głównym celem tego procesu jest przygotowanie zespołu do realizacji działań wytwórczych poprzez opracowanie planu rozwoju produktu. Zespół planistyczny składać się będzie z kierownika projektu, kierownika rozwoju oraz głównych programistów. Podczas planowania mają oni za zadanie określić kolejność realizacji funkcjonalności, pogrupować je w zestawy oraz przedstawić terminy ich dostarczenia (miesiące i lata). Decyzje te powinny brać pod uwagę m.in. zidentyfikowane zależności, obciążenia właścicieli klas, stopień złożoności oraz poziom ryzyka funkcjonalności. Następnie zestawy funkcjonalności przydzielane są głównym programistom, a klasy – pozostałym programistom.

Proces 4. Planowanie oparte na funkcjonalnościach

Procesy czwarty i piąty są iteracyjnymi procesami wytwórczymi realizowanymi zgodnie z opracowanym wcześniej planem. Główny programista na bazie ustalonego harmonogramu otrzymuje do realizacji zestaw funkcjonalności, z tego zbioru wybiera on zakres

do wykonania w najbliższej iteracji. Następnie, identyfikując właścicieli klas, tworzy on zespół funkcjonalnościowy, który odpowiedzialny będzie najpierw za przygotowanie diagramu następstw dla funkcji, a w kolejnym procesie także za jej dostarczenie. Na bazie efektów prac planistycznych zespołu główny programista może zmodyfikować całościowy model tak, aby oddawał on zmiany wniesione przez opracowywane funkcjonalności. Etap kończy się opracowaniem planu inspekcji dla przygotowywanych funkcjonalności.

Proces 5. Budowanie oparte na funkcjonalnościach

Proces piąty odpowiada za fizyczne tworzenie zaprojektowanych wcześniej funkcjonalności przez zespoły funkcjonalnościowe oraz wchodzących w ich skład właścicieli klas (programistów). Stworzony kod jest następnie testowany jednostkowo oraz podlega inspekcji. Po jej odbyciu może zostać zbudowany. Zakończenie procesu ma miejsce, gdy główny programista uzna, iż efekty spełniają oczekiwania. Wtedy to nowe funkcjonalności integrowane są z budowanym, całościowym rozwiązaniem, a sam zespół rozwiązywany, by umożliwić rozpoczęcie pracy nad nowym zakresem funkcjonalności.

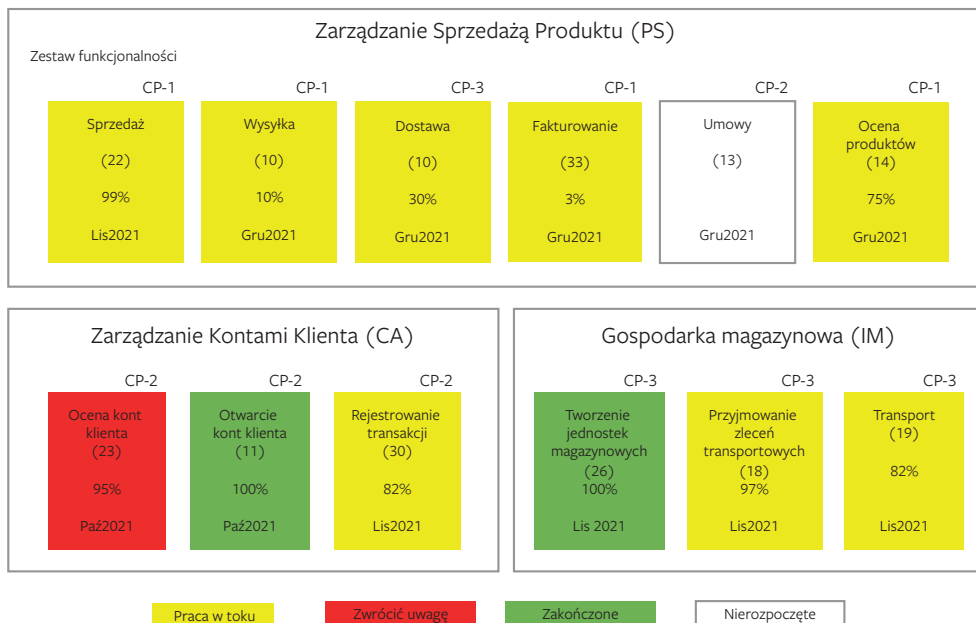
Procesy 4. i 5. wykonywane są iteracyjnie, generując kolejne przyrosty produktu. Mogą być także wykonywane równolegle przez różne zespoły pracujące przy jednym projekcie. Rozwiązanie takie zapewnia skalowalność metodyki także do dużych i rozległych projektów.

Pomiar postępów w FDD

Jedną z cech charakterystycznych FDD jest podejście do pomiaru i raportowania postępów. W tym celu autor metodyki zaproponował sześć „kamieni milowych” pozwalających odzwierciedlić postęp prac przy planowaniu i budowaniu funkcjonalności:

- 1) kamień milowy szczegółowego przeglądu domeny (ang. *domain walkthrough milestone*, 1%),
- 2) kamień milowy projektowania (ang. *design milestone*, 40%),
- 3) kamień milowy projektowania inspekcji (ang. *design inspection milestone*, 3%),
- 4) kamień milowy kodowania (ang. *code milestone*, 45%),
- 5) kamień milowy inspekcji kodu (ang. *code inspection milestone*, 10%),
- 6) kamień milowy zwolnienia przyrostu (ang. *promote to build milestone*, 1%).

Tak jak zaznaczono wyżej, każdemu z powyższych kamieni milowych przypisano miarę w procentach. Osiągnięcie kolejnych kamieni milowych pozwala wskazać procentowe zaawansowanie pracy nad funkcjonalnością oraz dokonywać przeliczeń, agregując wartości na wyższe poziomy. Informacje zbiorcze prezentowane są następnie w formie tabelarycznej, na wykresach oraz specjalnych raportach postępów, tzw. *parking lot*.

Rysunek 21. Przykład raportu statusu projektu, tzw. *parking lot*

Źródło: S.R. Palmer, J.M. Felsing, *A Practical Guide to Feature-Driven Development*, Prentice Hall, Hoboken 2002.

Dokładny pomiar postępów prac nad funkcjonalnościami umożliwia otrzymanie pełnej wiedzy dotyczącej dotychczasowego tempa pracy zespołu, faktycznych postępów, jak również prognoz odnośnie do dalszego jego przebiegu. Przyjazna i czytelna forma raportów sprzyja komunikacji z kluczowymi interesariuszami.

2.5.2. Podsumowanie

Metodyka FDD stanowi bardzo ciekawe rozwiązanie w ramach rodziny metodyk zwinnych. W stosunku do pozostałych przedstawicieli tej grupy do jej niewątpliwych zalet zaliczyć można relatywną prostotę zaleceń, jasną strukturę oraz wzorzec postępowania. Co ważne, metodyka daje zespołom solidne narzędzia i procesy umożliwiające przygotowanie się i zaplanowanie pracy wytwórczej oraz śledzenie jej postępów. Jest także skalowalna, a więc adekwatna także do dużych projektów informatycznych.

2.6. SCRUM

2.6.1. Geneza metodyki

Bezpośrednimi twórcami metodyki młyna – SCRUM – byli Jeff Sutherland z firmy Easel Corporation oraz Ken Schwaber, dyrektor zarządzający firmy Advanced Development Methods. Jeff Sutherland był głównym inżynierem w zespole Object Studio, który zainspirowany wcześniejszymi badaniami nad modelami tworzenia oprogramowania zdefiniował role, zatrudnił pierwszego właściciela produktu i szefa SCRUM, opracował pierwszy wykaz prac produktu (ang. *product backlog*), wykaz prac sprintu (ang. *sprint backlog*) i zbudował pierwszy portfel produktów stworzonych z wykorzystaniem metodyki SCRUM¹³⁶.

Pierwszy szczegółowy opis metody, datowany na rok 1993, Sutherland przedstawił w artykule *Agile Development: Lessons Learned from the First SCRUM*¹³⁷, zaś w 1995 r., wspólnie z K. Schwaberm, na konferencji Object-Oriented Programming, Systems, Languages & Applications (OOPSLA) w Stanach Zjednoczonych po raz pierwszy zaprezentował metodykę szerokiej publiczności. W latach 2002–2004 ciągle doskonalona przez autorów metodyka SCRUM została opublikowana w książce *Agile Software Development with SCRUM*¹³⁸.

2.6.2. Główne założenia metodyki

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

Metodyka, zgodnie z wartościami i zasadami *agile* zaprojektowana jest iteracyjne i elastycznie, czyli tak, aby w jak najlepszy sposób adaptować projekt do zmieniających się oczekiwań interesariuszy i umożliwić w ten sposób dostarczenie produktu na czas oraz przy jak najmniejszym nakładzie (zwłaszcza niepotrzebnej) pracy. W tym celu SCRUM posługuje się krótkimi, regularnymi etapami realizacji projektu tzw. sprintami (ang. *sprints*),

¹³⁶ J. Sutherland, K. Schwaber, *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*, <https://www.qagile.pl/wp-content/uploads/2018/11/scrum-papers.pdf> (dostęp: 24.11.2021).

¹³⁷ J. Sutherland, *Agile Development: Lessons Learned from the First SCRUM*, <https://www.torak.com/files/Lessons%20Learned%20From%20The%20First%20Scrum%20by%20Dr.%20Jeff%20Sutherland.pdf> (dostęp: 22.10.2021).

¹³⁸ K. Schwaber, M. Beedle, *Agile Software Development with SCRUM*, Prentice Hall, Hoboken 2002; K. Schwaber, *Agile Software Development with SCRUM*, Microsoft Press, Redmont 2004.

kładzie bardzo silny nacisk na jasną i czytelną priorytetyzację wymagań klienta oraz szybkie dostosowywanie rezultatów pracy do jego rzeczywistych potrzeb¹³⁹.

W przeciwieństwie do niektórych metodyk zwinnych, takich jak np. Extreme Programming (XP), SCRUM w większym stopniu koncentruje się na sferze zarządzania projektem (ról, procesów, podejmowania decyzji itp.), podczas gdy kwestie techniczne pozostają w gestii samego zespołu. XP podchodzi raczej do projektu od strony technicznej i inżynierskiej dostarczając technik i narzędzi wykorzystywanych bezpośrednio przez wykonawców produktów¹⁴⁰. Jednocześnie, warto podkreślić, iż SCRUM może być w łatwy sposób połączony i zintegrowany z innymi iteracyjnymi metodykami zarządzania, np. z Unified Process lub Extreme Programming¹⁴¹.

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

Struktura metodyki SCRUM oparta jest na trzech grupach elementów: trzech rolach, trzech ceremoniach i trzech artefaktach¹⁴²:

1) role:

- właściciel produktu (ang. *product owner*),
- szef SCRUM-a (ang. *scrum master*, *scrummaster*),
- zespół (ang. *scrum team*),

2) ceremonie:

- spotkanie planowania sprintu (ang. *sprint planning meeting*),
- spotkanie przeglądu sprintu (ang. *sprint review meeting*),
- codzienne zebrania (ang. *daily scrum meeting*),

3) artefakty:

- wykaz prac produktu (ang. *product backlog*),
- wykaz prac sprintu (ang. *sprint backlog*),
- wykres malejący (ang. *burndown chart*).

Role

Pierwszą rolą zgodnie z metodyką SCRUM jest **właściciel produktu** (ang. *product owner*). Jest to rola, która odpowiada za produkt finalny oraz wszystkie funkcjonalności, które będą dostarczały wartość klientom i użytkownikom. Rola ta zarządza także priorytetami ich implementacji. Głównym zadaniem właściciela produktu jest zapewnienie, że zespół robi „właściwe, z biznesowego punktu widzenia rzeczy”.

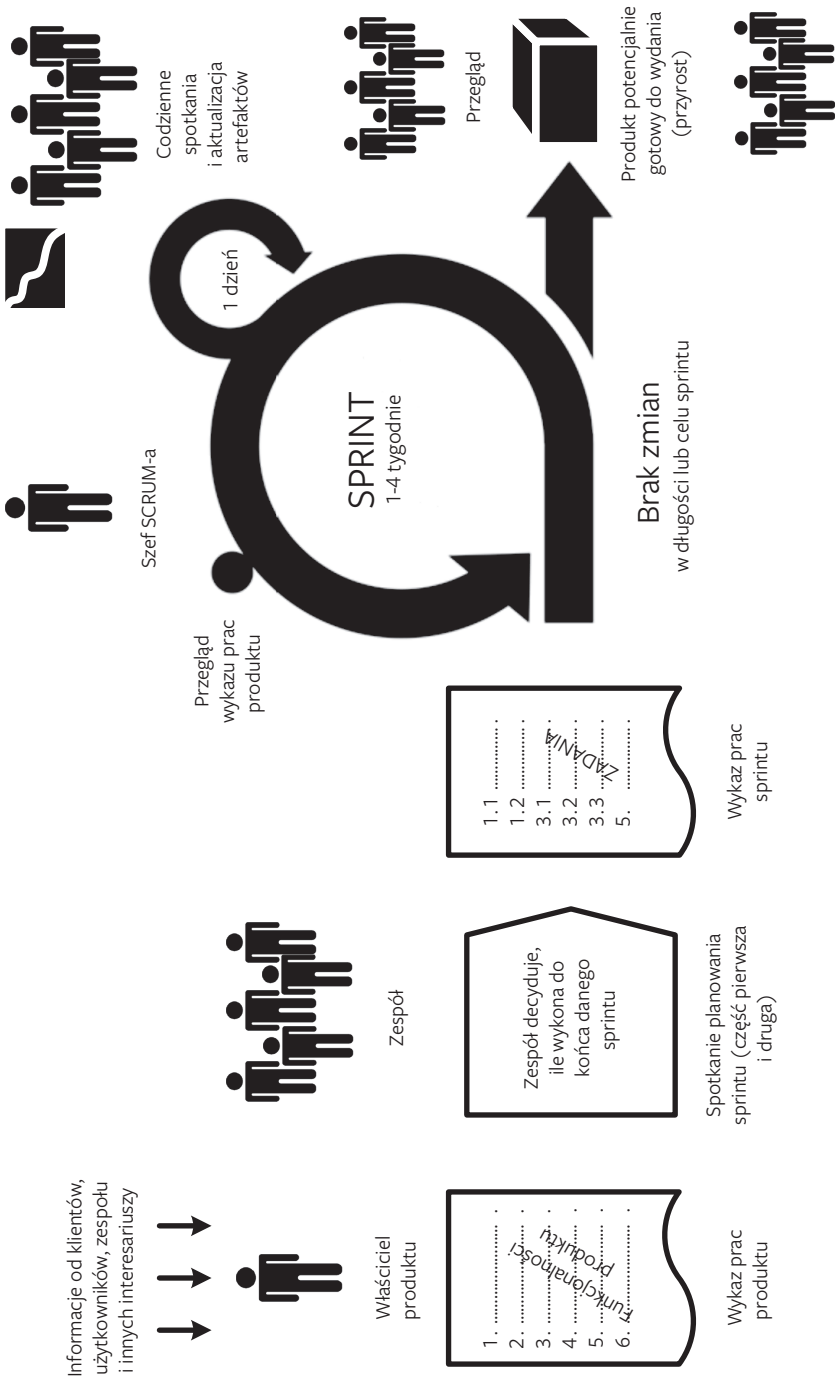
¹³⁹ J. Sutherland, K. Schwaber, *The SCRUM Papers...*

¹⁴⁰ G. Asproni, *Wstęp do SCRUM*, https://www.asprotunity.com/resources/articles/AnIntroductionToScrum_SDJ_06-2006_PL.pdf (dostęp: 24.11.2021).

¹⁴¹ J. Sutherland, *A Brief Introduction to SCRUM*, SCRUM Alliance, Westminster 2006.

¹⁴² G. Asproni, *Wstęp do SCRUM...*

Rysunek 22. Model metodyki SCRUM



Źródło: P. Deemer et al., SCRUM Primer: An Introduction to Agile Project Management with Scrum, ver. 1.2, <https://www.coursehero.com/file/49551637/scrumprimerpdf/> (dostęp: 10.08.2021).

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

Właściciel produktu odpowiada za¹⁴³:

- 1) dostarczenie poprzez produkt wartości klientom,
- 2) identyfikację i definiowanie funkcjonalności produktu,
- 3) nadanie priorytetów funkcjonalnościom zgodnie z ich wartością dla klientów,
- 4) podejmowanie decyzji o datach i zawartości kolejnych wydań produktu,
- 5) aktualizowanie zestawu funkcjonalności i ich priorytetów w kolejnych sprintach,
- 6) akceptację lub odrzucenie wykonanej pracy.

Rola właściciela produktu podobna jest do roli menedżera produktu lub menedżera marketingu produktu. Warto jednocześnie podkreślić, iż osoba zajmująca to stanowisko musi pozostać aktywna i ściśle współpracować z pozostałymi rolami. Rola właściciela produktu jest jednoosobowa i nie może być łączona z funkcją szefa SCRUM-a.

Drugą rolę w metodyce SCRUM jest rola **zespołu developerskiego**. Jego zadaniem jest wykonanie pracy potrzebnej do dostarczenia działających funkcjonalności produktu, wskazanych przez właściciela produktu. W skład zespołu wchodzi zazwyczaj od 5 do 9 osób posiadających umiejętności specjalistyczne konieczne do pracy w projekcie i osiągnięcia celów sprintu. W skład zespołu wchodzi zazwyczaj: analitycy, projektanci, programiści, testerzy, administratorzy i inni.

Zespół w SCRUM-ie charakteryzuje się zupełnie płaską strukturą. Każdy członek występuje na równych zasadach i może zaangażować się w dowolne zadanie. Zespół ma charakter samoorganizujący się, czyli członkowie we własnym zakresie decydują o zobowiązaniach zespołu i sposobie ich wypełnienia, odpowiadają za organizację swojej pracy i wykonują ją bez ingerencji z zewnątrz. Mogą wybrać dowolną ścieżkę działania pod warunkiem, że prowadzi ona do osiągnięcia celu sprintu i na jego koniec będą mogli zaprezentować działającą funkcjonalność oprogramowania. Zarówno szef, jak i właściciel produktu muszą pozostawić zespołowi swobodę poruszania się w ramach pracy przydzielonej do wykonania w ramach danej iteracji; nie powinni ingerować w zatwierdzony plan.

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

Ostatnią rolę wyróżnioną w metodyce jest rola **szefa SCRUM-a** (ang. *scrum master*). Wbrew swojej nazwie nie posiada on uprawnień rozkazodawczych, nie pełni

¹⁴³ J. Sutherland, K. Schwaber, *The SCRUM Papers...*

roli kierownika, a tym bardziej kierownika projektu. Szef SCRUM-a jest rolą wspierającą zespół i właściciela produktu w realizacji projektu z sukcesem, odpowiada za zrozumienie i stosowanie zasad metodyki w zespole, wspiera procesy uczenia się. Działa na rzecz zespołu i jednym z jego kluczowych zadań jest bycie buforem między zespołem a jego bliższym i dalszym otoczeniem, osobą odsuwającą przeciwności, blokującą interwencje z zewnątrz, rozwiązującą problemy mogące utrudnić zespołowi realizację zadeklarowanych celów.

Szef SCRUM-a¹⁴⁴:

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

Trzeba w tym miejscu silnie podkreślić, że metodyka SCRUM nie przewiduje roli *stricte* kierownika projektu, takiej jaka jest znana z innych metodyk zarządzania projektami. Jego zadania zostały rozdzielone na trzy inne. Zgodnie z ideą SCRUM właściciel produktu odpowiada za wyznaczenie i priorytetyzację celów, szef SCRUM-a ma pełnić rolę guru, opiekuna zespołu, zaś sam zespół – złożony przecież z najlepszych specjalistów – ma samodzielnie decydować o sposobach realizacji zadań i samoorganizować swoją pracę. Taki podział odpowiedzialności w zespole stanowi znaczną zmianę i wyzwanie dla dotychczasowych stylów zarządzania w wielu organizacjach.

Proces SCRUM-a

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

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

¹⁴⁴ Ibid.

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

Wykaz prac produktu

Pierwszy krok realizacji projektu zgodnego z metodyką SCRUM należy do właściciela produktu. Jego zadaniem jest stworzenie wizji produktu, która następnie przybiera formę listy, składającej się z uporządkowanych zgodnie z priorytetami elementów funkcjonalności tworzonego oprogramowania. Lista ta nosi nazwę wykazu prac produktu (ang. *product backlog*). Wykaz prac produktu jest jednym dokumentem przedstawiającym jeden, definitywny opis „wszystkiego co może być kiedykolwiek zrobione przez zespół, w kolejności ważności”¹⁴⁵.

Elementami umieszczanymi w wykazie będą przede wszystkim nowe funkcjonalności oprogramowania, cele związane z poprawą wydajności systemu, prace rozwojowe i pomysły na innowacje, jak również prace nad rozwiązaniem znanych usterek. Metodyka nie zakłada jednego sposobu opisu funkcjonalności systemu. Mogą być one opisywane w dowolny sposób przyjęty w projekcie: w postaci równoważników zmian, tzw. historyjek, życzeń użytkownika itp. Autorzy metodyki sugerują, aby pracochłonność jednej pozycji w wykazie nie przekraczała 10 dni pracy programistów¹⁴⁶. Przykład wykazu prac produktu przedstawiony jest w tabeli 7:

Tabela 7. Wykaz prac produktu

Pozycja wykazu	Specyfikacja (link wiki)	Priorytet	Szacowana wartość	Wstępny szacunek pracochłonności	Aktualne szacunki pracy pozostającej do wykonania					
					1	2	3	4	5	6
Jako kupujący chcę móc umieścić książkę w koszyku zakupów	...	1	7	5						
Jako kupujący chcę móc usunąć książkę z koszyka zakupów	...	2	6	2						
Podnieść wskaźniki wydajności przetwarzania transakcji	...	3	6	13						
Rozpoznać możliwości przyspieszenia walidacji kart płatniczych	...	4	6	20						

¹⁴⁵ P. Deemer et al., *SCRUM Primer: An Introduction to Agile Project Management with Scrum*, ver. 1.2, <https://www.coursehero.com/file/49551637/scrumprimerpdf/> (dostęp: 10.08.2021).

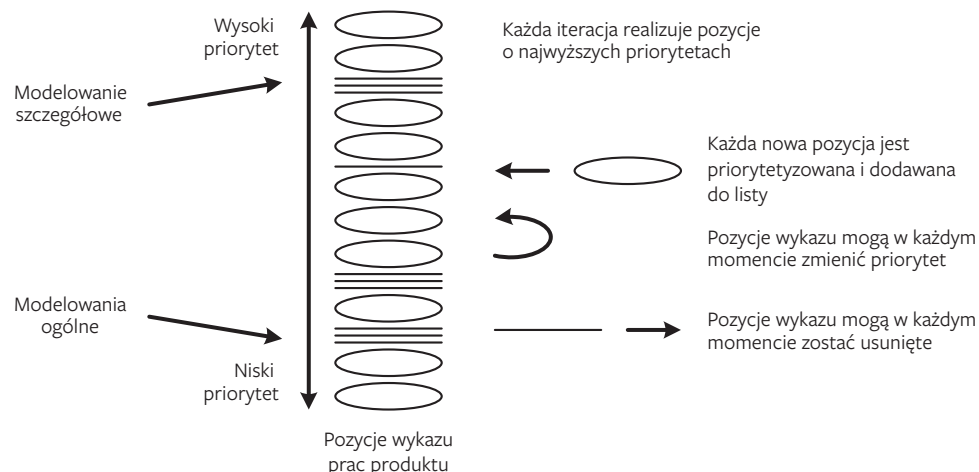
¹⁴⁶ J. Sutherland, *A Brief Introduction...*

Pozycja wykazu	Specyfikacja (link wiki)	Priorytet	Szacowana wartość	Wstępny szacunek pracochłonności	Aktualne szacunki pracy pozostającej do wykonania					
					1	2	3	4	5	6
Zaktualizować wszystkie serwery do Apache 2.2.3	...	5	5	13						
Zdiagnozować i naprawić błędy skryptu realizacji zamówienia	...	6	2	3						
Jako kupujący chcę stworzyć i zapisać listę zakupów	...	7	7	40						
Jako kupujący chcę móc dodawać i odejmować pozycje z listy zakupów	...	8	4	20						

Źródło: *ibid.*

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

Rysunek 23. Dynamika wykazu prac produktu



Źródło: S. Ambler, *Rethinking the Role of Business Analysts: Towards Agile Business Analysts?*, <http://agilemodeling.com/essays/businessAnalysts.htm> (dostęp: 04.12.2021).

Jak podkreślają sami autorzy, nieprawdą jest jakoby SCRUM odrzucał szczegółową pisemną specyfikację funkcjonalności oprogramowania. W rzeczywistości decyza

o stopniu sformalizowania i szczegółowości opisu należy do właściciela produktu i zespołu. Najczęściej elementy o wysokim priorytecie, bardzo dobrze znane i wdrażane w niedługiej przyszłości będą opisane w dokładniejszy sposób niż funkcjonalności o marginalnym znaczeniu.

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

Spotkanie planowania sprintu

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

Na początku pierwszego oraz każdego kolejnego sprintu odbywa się spotkanie planowania sprintu (ang. *sprint planning meeting*). Jego celem jest opracowanie szczegółowego planu dla bieżącej iteracji. Spotkanie planowania sprintu trwa zazwyczaj jeden pełny dzień i składa się z dwóch części po 4 godziny każda. W trakcie pierwszej części właściciel produktu wraz z zespołem i przy wsparciu ze strony szefa SCRUM-a dokonuje przeglądu wizji, mapy drogowej, planu wydań oraz wykazu prac produktu. Zespół powinien zapoznać się z opisami pozycji wykazu, którymi w pierwszej kolejności zainteresowany jest właściciel produktu oraz wyrazić swoją opinię na temat dokładności szacunków ich pracochłonności.

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

Zjawisko odpowiedzialnego „zasysania” pracy, a nie „wpychania” jej do zespołu przez kierownictwo – co podkreślają autorzy metodyki – działa zdecydowanie sprzyjająco produktywności i jest jednym z kluczowych komponentów szczupłego rozwoju produktów (ang. *lean product development*)¹⁴⁷.

Po podjęciu decyzji o elementach wchodzących w skład sprintu rozpoczyna się część druga spotkania, polegająca na szczegółowym zaplanowaniu zadań i czasochłonności pracy do wykonania. Dokonuje się tego za pomocą dekompozycji funkcjonalności wyszczególnionych w wykazie prac produktu na zadania cząstkowe (ang. *sprint tasks*) konieczne do

¹⁴⁷ Ibid.

ich wdrożenia. Podział ten powinien doprowadzić do sytuacji, w której zidentyfikowane zadania nie będą trwały więcej niż 2 dni (16 godzin roboczych). Opracowana lista zadań na najbliższą iterację nosi nazwę wykazu prac sprintu (ang. *sprint backlog*). W momencie spisania wykaz prac sprintu niekoniecznie musi być kompletny, gdyż plan konstruowany jest przy danym (niepełnym) stanie wiedzy zespołu. Lista ta z pewnością będzie aktualizowana w miarę upływu czasu trwania danej iteracji.

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

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

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

SCRUM codzienny

Od momentu rozpoczęcia sprintu zespół pracuje na bieżąco nad realizacją przyjętych zobowiązań. Zgodnie z metodyką SCRUM w trakcie sprintu odbywają się dodatkowo codzienne zebrania (ang. *daily scrum meetings*) inaczej zwane również „spotkaniami na dzień dobry” lub „SCRUM-em codziennym”. Zebrania te odbywają się każdego dnia o tej samej porze, najlepiej przed rozpoczęciem dnia pracy, prowadzone są przez szefa SCRUM-a i mają charakter „synchronizacji” oraz bieżącej wymiany informacji na temat codziennych zajęć, wykonanych postępów oraz zauważonych trudności. Spotkania te są z założenia krótkie i nie powinny trwać więcej niż kwadrans, dlatego najczęściej przeprowadza się je na stojąco, stąd również ich inna nazwa: *stand-up meetings*. Dla członków zespołu, którzy jako jedyni mogą zabierać głos, obecność jest obowiązkowa. Inni zainteresowani mogą być obecni, ale bez możliwości wypowiedziania się.

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

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

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

Wykres spalania

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

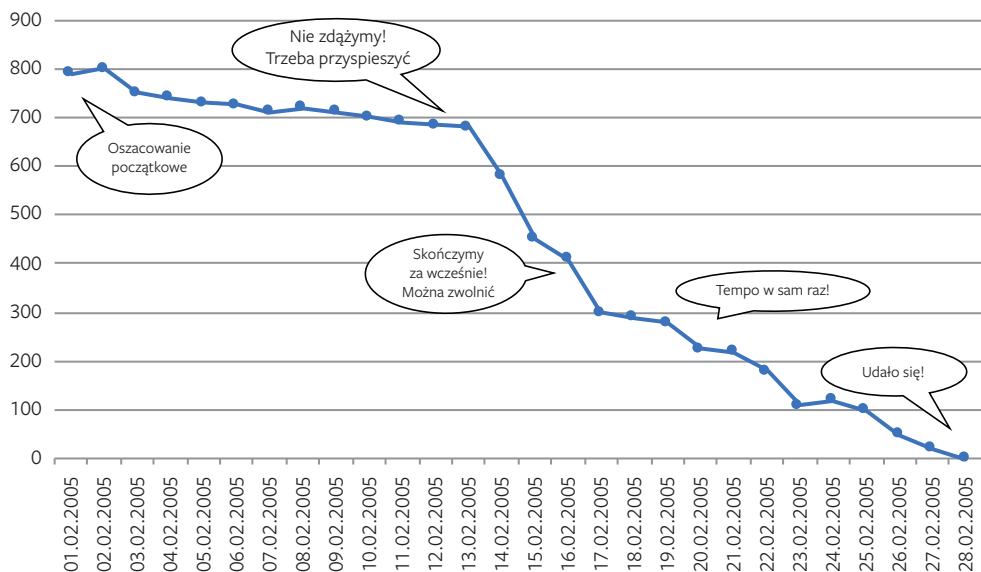
Zadaniem wykresu jest skupienie uwagi zespołu na celu – zrealizowaniu wszystkich zadań sprintu w zamierzonym czasie – oraz pokazanie postępu w realizacji tego celu. Postęp ten nie jest jednak mierzony w odniesieniu do czasu, jaki upłynął, ale w stosunku do czasu, jaki pozostał.

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

W idealnej sytuacji linia wykresu skierowana jest do dołu, jednakże w praktyce mogą zdarzyć się przypadki, gdy pomimo pracy wykonanej ilość pracy pozostałej jeszcze wzrośnie. Może wystąpić to m.in. w przypadku gdy¹⁴⁸:

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

¹⁴⁸ *Ibid.*

Rysunek 24. Monitorowanie projektu z wykorzystaniem wykresu spalania

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

Spotkanie przeglądu sprintu i spotkanie retrospektywne sprintu

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

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

W trakcie spotkania przeglądu sprintu zespół – wraz z właścicielem produktu oraz przy wsparciu szefa SCRUM-a – dokonują przeglądu wszystkiego tego, co wydarzyło się w trakcie ostatniej iteracji i dotyczyło opracowywanego produktu. Zespół przedstawia wyniki swojej pracy oraz demonstruje działające funkcjonalności produktu, zaś właściciel produktu ma za zadanie zapoznać się z rozwiniętym produktem i przekazać zespołowi aktualne informacje na temat wizji produktu i stanu rynku. Istotnym przesłaniem spotkania jest zapewnienie wymiany informacji i konwersacji pomiędzy stronami, tak

aby uczestnicy spotkania nauczyli się od siebie jak najwięcej i mogli lepiej zrozumieć aktualny stan i perspektywy realizowanego projektu.

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

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

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

Kolejna iteracja – nowy sprint

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

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

Zgodnie z metodyką SCRUM nowy sprint powinien rozpocząć się bezzwłocznie po zakończeniu ostatniego. Najlepiej aby spotkanie planujące sprint zostało przeprowadzone już w kolejnym dniu roboczym po spotkaniu przeglądu sprintu. Silnie podkreślaną cechą metodyki SCRUM jest jej szybkie tempo oraz realizacja kolejnych cykli iteracji zgodnie ze stałą częstotliwością i bez zbędnych opóźnień.

2.6.3. Podsumowanie

Źródłem metodyki SCRUM były bez wątpienia projekty opracowania nowych produktów w branży informatycznej, jednakże jej zastosowanie nie ogranicza się wyłącznie do oprogramowania i systemów informatycznych. Idea koncentracji uwagi na produkcie i bliskiej, dynamicznej współpracy w ramach zespołu, bieżącej wymiany informacji oraz rozwoju produktu opartego na podejściu przyrostowym znajduje szerokie zastosowanie w wielu innych branżach i projektach, szczególnie tych o wysokim stopniu złożoności oraz innowacji. SCRUM może znaleźć zastosowanie w zarządzaniu oprogramowaniem i rozwojem sprzętu, w dostarczaniu wsparcia, działalności reklamowej, marketingu i wielu innych dziedzinach¹⁴⁹. Na stronie SCRUM Alliance przeczytać można opisy zastosowania tej metodyki m.in. w operacjach logistycznych marynarki brytyjskiej oraz zarządzania portfelem działalności dużego banku¹⁵⁰.

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

Doświadczenia Yahoo! były przedmiotem badania przeprowadzonego w latach 2005–2007 przez Gabrielle Benefield (*Senior Director of Agile Development*)¹⁵¹. Wśród zaobserwowanych zmian dostrzegła ona m.in. wzrost produktywności zespołów średnio o 37%, przy chęci dobrowolnego kontynuowania stosowania metody przez 86% ankietowanych. W innym artykule przedstawiono wyniki firmy informatycznej znajdującej się na piątym poziomie CMMI, w której po zastosowaniu metodyki SCRUM koszt projektów informatycznych spadł o połowę, a poziom błędów o 40%, przy zachowaniu poziomemu CMMI¹⁵².

Wśród innych zalet metodyki SCRUM wskazać można m.in.¹⁵³:

- 1) wysoki poziom kontroli nad przebiegiem projektu za pomocą krótkich iteracji,
- 2) wczesne uzyskanie istotnych elementów projektu dających wymierne korzyści biznesowe,
- 3) przejrzystość planowania i wdrożenia oprogramowania,

¹⁴⁹ P. Hundermark, *Do Better Scrum*, https://moam.info/do-better-scrum-scrum-sense_59c59c471723dde092c9f7da.html (dostęp: 24.11.2021).

¹⁵⁰ <http://SCRUM-alliance.com> (dostęp: 24.11.2021).

¹⁵¹ G. Benefield, *Rolling out Agile at large Enterprise*, w: J. Sutherland, K. Schwaber (eds.), *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*, https://www.academia.edu/43872765/The_Scrum_Papers_Nut_Bolts_and_Origins_of_an_Agile_Framework (dostęp: 20.10.2021).

¹⁵² J. Sutherland, C. Jacobson, K. Johnson, *Scrum and CMMI Level 5: A Magic Potion for Code Warriors*, <https://www.scruminc.com/wp-content/uploads/2014/05/Scrum-and-CMMI-Level-5-A-Magic-Potion-for-Code-Warriors.pdf> (dostęp: 24.11.2021).

¹⁵³ J. Sutherland, K. Schwaber, *The SCRUM Papers...*

- 4) wzrost szybkości opracowywania nowych produktów,
- 5) powiązanie celów indywidualnych z celami organizacji,
- 6) tworzenie kultury opartej na rezultatach,
- 7) wspieranie tworzenia wartości dla interesariuszy,
- 8) zapewnienie stabilnych i trwałych kanałów komunikacyjnych na wszystkich poziomach,
- 9) umożliwienie rozwoju osobistego i poniesienie jakości życia.

SCRUM posiada niewątpliwie również pewne wady i ograniczenia. Przede wszystkim, jak podkreślają to sami autorzy, SCRUM jest metodyką ramową i nie oferuje zespołom szczegółowego zestawu praktyk do zarządzania projektem. SCRUM może ujawnić problemy i nieprawidłowości w sposobie prowadzenia projektu, ale nie dostarczy ich jednoznacznych rozwiązań – wymaga w tym zakresie dojrzałości i wiedzy od uczestników projektów¹⁵⁴. Jak stwierdza G. Asproni „należy uwzględnić fakt, iż nie każdy się do SCRUM-a nadaje (...) u najbardziej introwertycznych programistów konieczność tak ścisłej współpracy może wywołać poczucie dyskomfortu. Niektórzy programiści czują się też jedynymi właścicielami kodu i nie lubią, gdy ktoś im ten kod modyfikuje lub choćby ogląda”¹⁵⁵.

Często poruszany jest również temat skalowalności metody oraz stosowania jej w dużych, rozproszonych geograficznie zespołach jak również w tych, które są w słabo zintegrowane. Zdecydowanie utrudnia to częsty bezpośredni kontakt, który jest jednym z kluczowych czynników sukcesu tej metody¹⁵⁶.

Wątpliwości budzi również zastosowanie metodyki SCRUM w projektach o znaczeniu krytycznym, jednakże dotychczasowe doświadczenia wskazują, iż w takim przypadku nic nie stoi na przeszkodzie, aby proces SCRUM uzupełnić o bardziej szczegółowe i rozbudowane procedury kontroli jakości i zawansowane mechanizmy testowania¹⁵⁷.

2.7. Adaptive Software Development (ASD)

Metodyka Adaptive Software Development (ASD) opracowana została przez dwóch autorów: Jima Highsmitha oraz Sama Bayera w 1999 r. Oparli ją oni na swoich doświadczeniach w pracy z metodykami tradycyjnymi, zwłaszcza na wcześniejszej teorii złożonych systemów adaptacyjnych (ang. *complex adaptive systems*, CAS)¹⁵⁸. ASD czerpie także z dorobku modelu RAD (ang. *rapid application development*) – szybkiego tworzenia aplikacji.

¹⁵⁴ P. Deemer et al., *SCRUM Primer...*

¹⁵⁵ G. Asproni., *Wstęp do SCRUM...*

¹⁵⁶ J. Milewski, *Projektowy młyn*, „Computerworld” 2005, nr 29.

¹⁵⁷ *Ibid.*

¹⁵⁸ P. Meso, R. Jain, *Agile Software Development: Adaptive Systems Principle and Best Practices*, „Information Systems Management”, Summer 2006, s. 19.

Zaproponowane przez tych autorów podejście do realizacji projektów informatycznych kładzie nacisk przede wszystkim na dostarczanie wartościowych z punktu widzenia klienta rezultatów. Wysoka wartość tychże wyników ma być efektem szybkiej adaptacji zespołu w stosunku do wydarzeń zachodzących zarówno w środowisku wewnętrznym, jak i zewnętrznym projektu. Metodyka bazuje na filozofii podobnej, jak w podejściu ewolucyjnym. Adaptacja ma dawać rezultaty lepszej jakości niż stosowanie formalnych technik optymalizacji i doskonalenia procesów.

Szczegółowe omówienie metody zostało zaprezentowane przez Jima Highsmitha w książce pt. *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*, wydanej w 1999 r. przez Dorset House Publishing, oraz w artykule z 1997 r. pt. *Messy, exciting, and anxiety-ridden: adaptive software development* opublikowanym w czasopiśmie fachowym "American Programmer"¹⁵⁹.

2.7.1. Struktura

Metodyka ASD zawiera zalecenia dotyczące procesu inżynierii oprogramowania w zakresie tworzenia systemów informatycznych. Autorzy odrzucają w niej tradycyjne, liniowe podejście kaskadowe (ang. *waterfall*) i rekomendują budowanie dużych, złożonych systemów z wykorzystaniem modelu iteracyjnego i inkrementalnego ze znacznym użyciem prototypowania.

Zgodnie z wytycznymi autora metodyki ASD cykl życia projektu cechuje się sześcioma charakterystykami. Jest on¹⁶⁰:

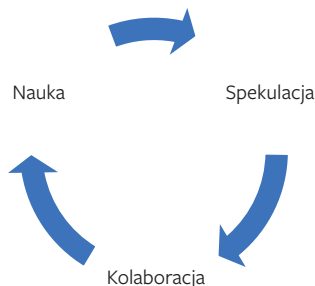
- 1) skupiony na misji,
- 2) oparty na funkcjonalnościach,
- 3) iteracyjny,
- 4) ograniczony w czasie (ang. *time-boxed*),
- 5) napędzany ryzykiem (ang. *risk driven*),
- 6) akceptujący zmiany (ang. *change tolerant*).

Model cyklu życia projektu zgodnego z ASD składa się z powtarzających się trzech faz: spekulacji, współpracy oraz nauki (ang. *speculate, collaborate, learn*), które zastępują dotychczasowe: planowania, projektowania i budowania.

Celem poszczególnych faz cyklu jest ukierunkowanie zespołu na ciągłe uczenie się, akceptację i podążanie za zmianami, ocenę dotychczasowych działań, spekulowanie odnośnie do przyszłości, jak również wsparcie intensywnej współpracy między uczestnikami projektu.

¹⁵⁹ J. Highsmith, *Messy, exciting, and anxiety-ridden: adaptive software development*, "American Programmer" 1997, vol. 10, no. 1.

¹⁶⁰ J. Highsmith, *Agile Software Development Ecosystems*, Addison-Wesley Professional, New York 2002, s. 312.

Rysunek 25. Fazy metodyki ASD

Źródło: J. Highsmith, *Messy, exciting, and anxiety-ridden: adaptive software development*, "American Programmer" 1997, vol. 10, no. 1.

Faza spekulacji odpowiada tradycyjnym fazom inicjowania i planowania, jednak w odmianie Highsmitha nie ma ona za zadanie stworzenia idealnego planu przebiegu projektu i dokładnej specyfikacji, a raczej stanowi próbę przewidzenia przez zespół możliwego przebiegu prac przy świadomości, iż rzeczywistość jest niepewna i trudna do określenia. Spekulacja uznaje planowanie w sytuacji niepewności i złożoności problemu, wspierając eksplorację oraz eksperymentowanie.

Zadania podczas fazy spekulacji:

- 1) określić misję i cele projektu, jego ograniczenia, ustanowić organizację projektu, zidentyfikować wymagania, oszacować wstępnie zakres i rozmiar projektu, zidentyfikować kluczowe ryzyka,
- 2) określić czas realizacji całego projektu uwzględniając rozpoznany zakres, zestawy funkcjonalności i dostępność zasobów,
- 3) określić liczbę i czas iteracji przewidzianych do realizacji w projekcie,
- 4) ustalić cele i temat przewodni dla każdej iteracji oraz przypisać funkcjonalności do każdej z nich.

Faza współpracy jest odpowiednikiem fazy budowania rozwiązań, lecz i w tym przypadku autor podkreśla, iż rozwiązania nie są budowane, lecz ewoluują.¹⁶¹ Nazwa fazy ma podkreślać konieczność intensywnego wspólnego wysiłku, komunikacji i samoorganizacji w pracy zespołu projektowego.

Zadania podczas fazy współpracy:

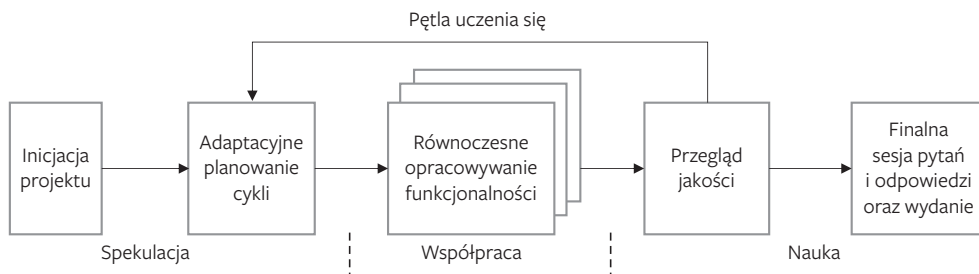
- 1) zespół techniczny dostarcza opisane w zakresie iteracji funkcjonalności,
- 2) kierownik projektu koordynuje równoległe prace i wspiera współpracę w zespole.

Ostatnia faza nauki odpowiada dotychczasowej fazie testowania i przeglądu jakości. Na zakończenie każdej iteracji zespół dokonuje przeglądu uzyskanych efektów i sposobu pracy w celu adaptacji i modyfikacji praktyk, tak aby kolejna iteracja przebiegła sprawniej

¹⁶¹ *Ibid.*, s. 174.

i lepiej. Według autora metodyki zespół powinien w szczególności zwrócić uwagę na cztery kategorie doświadczeń i nowej wiedzy: jakość rezultatów z perspektywy użytkowników, jakość rezultatów z perspektywy technicznej, sposób funkcjonowania zespołu projektowego, status i stan projektu. Zespół może w tym zakresie wykorzystać m.in. spotkania retrospektywne oraz grupy fokusowe z udziałem użytkowników.

Rysunek 26. Cykl życia projektu adaptacyjnego



Źródło: James A. Highsmith, *Agile Software Development Ecosystems*, Addison-Wesley Professional, New York 2002, s. 176.

2.7.2. Podsumowanie

Jim Highsmith jest autorem dwóch podejść do zarządzania projektami: Adaptive Systems Development oraz Agile Project Management (opisana została w osobnym rozdziale). Obie te metodyki mają swoje wspólne korzenie i przyświecające im podobne wartości: adaptacja, zmiana, iteracyjność. O ile APM ma charakter metodyki uniwersalnej, o tyle ASD dedykowana jest *stricte* projektom tworzenia oprogramowania. Ich wartość związana jest z możliwością ustanowienia na ich podstawie ramowych wytycznych i zasad kultury pracy zespołów projektowych. Swoboda w ustaleniu bardziej szczegółowych regulacji oraz wzbogacenia metodyki o techniki i narzędzia może stanowić jednocześnie jej zaletę i wadę.

2.8. Ekstremalne zarządzanie projektami – Extreme Project Management (XPM)

Na fali popularności zwinnych metodyk zarządzania wytwarzaniem oprogramowania powstawać zaczęły także koncepcje czerpiące z filozofii *agile*, ale mające bardziej uniwersalny charakter. Jedną z takich metodyk jest omówiona wcześniej metodyka APM autorstwa Jima Highsmitha, drugą – Extreme Project Management (XPM)¹⁶².

¹⁶² Extreme Project Management® jest zastrzeżonym znakiem handlowym należącym do Doug DeCarlo Group.

Założenia XPM opracowane zostały przez Douga DeCarlo (ur. 1942), eksperta z zakresu zarządzania projektami, konsultanta, trenera, wykładowcę, członka PMI oraz starszego konsultanta ds. zwinnego zarządzania projektami w Cutter Consortium¹⁶³. Kompleksowy opis zaproponowanego przez niego podejścia odnaleźć można w książce jego autorstwa pt. *eXtreme Project Management. Using Leadership, Principles, and Tools to Deliver Value in the Face of Volatility* wydanej przez Jossey-Bass w 2004 r. Popularyzację XPM DeCarlo zawdzięcza także osobom takim, jak James P. Lewis oraz Rober K. Wysocki, którzy promują metodykę w książkach swojego autorstwa¹⁶⁴.

Extreme Project Management jest, w zamyśle autora, metodyką zalecaną do zarządzania projektami cechującymi się wysokim poziomem złożoności i niepewności, w przypadkach, gdy zarówno określenie celu, jak i opracowywanego rozwiązania jest mgliste i niejasne. Przedsięwzięcia te są projektami wysokiego ryzyka, o ogromnej liczbie zmian i szybkim tempie realizacji. Zjawiska towarzyszące projektom ekstremalnym i wymagające takiego podejścia to np.¹⁶⁵:

- 1) raptownie zmieniające się wymagania,
- 2) eksperymentalne, innowacyjne technologie i metody, których wcześniej nie stosowano,
- 3) bardzo krótkie terminy wykonania,
- 4) wysokie wymagania dyspozycyjności zasobów w trakcie projektu,
- 5) zmiany oczekiwań klienta w trakcie trwania projektu,
- 6) chaotyczne i nieprzewidywalne, często zmieniające się otoczenie projektu.

Należą do nich także¹⁶⁶:

- 1) sceptycyzm w zespole odnośnie do szans powodzenia,
- 2) niechęć do zarządzania projektami i procesów towarzyszących,
- 3) bierność sponsora projektu,
- 4) brak władzy kierownika nad członkami zespołu,
- 5) wysoka rotacja wśród członków zespołu,
- 6) wysoki poziom stresu i obciążenia psychicznego wynikających z długiego czasu pracy i nieprzewidywalnego środowiska projektu.

Sytuacje takie wymagają podejścia ekstremalnego i to właśnie od niego, podobnie jak w przypadku ekstremalnego programowania, wzięła nazwę metodyka DeCarla. Podejście ekstremalne cechuje otwartość, elastyczność i brak determinizmu, a przede wszystkim skupienie na ludzkiej stronie projektu: zespole, kliencie i interesariuszach oraz zapew-

¹⁶³ <http://www.dougdecarlo.com/> (dostęp: 24.11.2021).

¹⁶⁴ Zob. R.K. Wysocki, J.P. Lewis, D. DeCarlo, *The World Class Project Manager: A Professional Development Guide*, Basic Books 2001; R.K. Wysocki, *Efektywne zarządzanie projektami. Tradycyjne, zwinne, ekstremalne*, Helion, Gliwice 2013.

¹⁶⁵ D. DeCarlo, *eXtreme Project Management. Using Leadership, Principles, and Tools to Deliver Value in the Face of Volatility*, Jossey-Bass, New York 2004, s. 3.

¹⁶⁶ *Ibid.*, s. 144.

nieniu efektywnej współpracy między nimi. Jak twierdzi autor: „projekty ekstremalne są złożonymi, bardzo szybkimi samokorygującymi się przedsięwzięciami, podczas których ludzie wchodzą między sobą w interakcje poszukując pożądanego rezultatu w warunkach wysokiej niepewności, zmian i napięcia”¹⁶⁷. Towarzysząca mu wizja projektów ekstremalnych opiera się na wartościach takich jak: zarządzanie niepewnością w turbulentnych środowiskach, chaos i nieporządek, szybkość i innowacja, planowanie *just-in-time*. DeCarlo twierdzi, iż dotychczasowe podejście – *Ready, Aim, Fire* – powinno zostać zastąpione przez – *Ready, Fire, Aim*¹⁶⁸.

Autor metodyki przeciwstawia ją tradycyjnemu zarządzaniu projektami, zarzucając dotychczasowym praktykom powolność, przesadną stabilizację i planowanie:

Tabela 8. Porównanie tradycyjnego i ekstremalnego zarządzania projektami

Tradycyjne zarządzanie projektami	Ekstremalne zarządzanie projektami
Zorientowane na przeszłość	Zorientowane na przyszłość
Ludzie w służbie procesów	Procesy w służbie ludzi
Centralizacja kontroli nad ludźmi, procesami i narzędziami	Rozproszenie kontroli nad ludźmi, procesami i narzędziami
Przyjmowanie odpowiedzialności za rzeczywistość	Przyjmowanie odpowiedzialności za siebie i swoje podejście do rzeczywistości
Koncentracja na zarządzaniu	Koncentracja na przewodzeniu

Źródło: D. DeCarlo, *eXtreme Project Management. Using Leadership, Principles, and Tools to Deliver Value in the Face of Volatility*, Jossey-Bass, New York 2004, s. 3.

Podsumowując główne założenia XPM, warto przytoczyć definicję projektu oraz ekstremalnego zarządzania nim przygotowane przez ich autora. Projekt to „zlokalizowane pole energii składające się z emocji i interakcji stale wyrażających siebie w fizycznej formie”¹⁶⁹, zaś ekstremalne zarządzanie projektami to: „sztuka i nauka wspierania i zarządzania przepływem myśli, emocji i interakcji w sposób, który dostarcza wartościowych rezultatów w turbulentnych i złożonych środowiskach, tj. takich, które cechują się dużą szybkością, częstymi zmianami, wysoką niepewnością i wysokim poziomem napięcia”¹⁷⁰.

¹⁶⁷ *Ibid.*, s. 7.

¹⁶⁸ *Ibid.*

¹⁶⁹ *Ibid.*, s. 30.

¹⁷⁰ *Ibid.*, s. 34.

2.8.1. Struktura metodyki

Struktura metodyki XPM oparta została na czterech elementach. Są to:¹⁷¹

- 1) 4 akceleratorzy (ang. *accelerators*) – zasady i pryncypia wyzwalające motywację oraz innowacje,
- 2) 10 wspólnych wartości – budujących wśród interesariuszy zaufanie i pewność siebie,
- 3) 4 pytania biznesowe – których zadawanie sprawia, iż klienci uzyskują odpowiednio wcześniej i często wartość z projektu,
- 4) 5 krytycznych czynników sukcesu – umiejętności, narzędzia oraz wsparcie ze strony organizacji konieczne do powodzenia projektów.

Akceleratorzy

Akceleratorzy mają charakter wartości, poglądów i przekonań, które pomagają interesariuszom współpracować w nurcie adaptacyjności i budować zwarty zespół. DeCarlo porównuje je do „żywej ziemi, na której wzrastać będą innowacyjne produkty”¹⁷². Są to:

1. Uczyń zmianę swoim przyjacielem – zmiana jest szansą na innowację i doskonalenie, akceptuj ją spraw aby pomagała w realizacji projektu; nie sprowadzaj jej do biurokratycznej kontroli zmian;
2. Buduj na ludzkich pragnieniach by coś zmienić – ludzie potrzebują wizji i misji, która będzie określała ich znaczenie i wpływ na większą całość;
3. Podkreślaj wkład w powstawanie rezultatów – uwierz w doświadczenie i zaangażowanie ludzi, pozwól im wpływać na kształt całego projektu;
4. Utrzymuj prostotę – stosuj zasadę „KISS” (ang. *keep it simple, stupid*) w zarządzaniu projektem aby skupić się na istotnych jego elementach

Wspólne wartości

Wspólne wartości stanowią jednakowe dla interesariuszy przekonania, które wspierają prace na rzecz powodzenia projektu. Budują one kulturę zespołu, który osiąga sukces nawet mimo przeciwności losu. Wśród wspólnych wartości w metodyce XPM znajdziemy¹⁷³:

Wartości dotyczące ludzi:

- 1) Ludzie przede wszystkim – eliminujmy przeszkody w pracy zespołu.
- 2) Uczciwa komunikacja – przekazywanie sobie prawdziwych informacji.

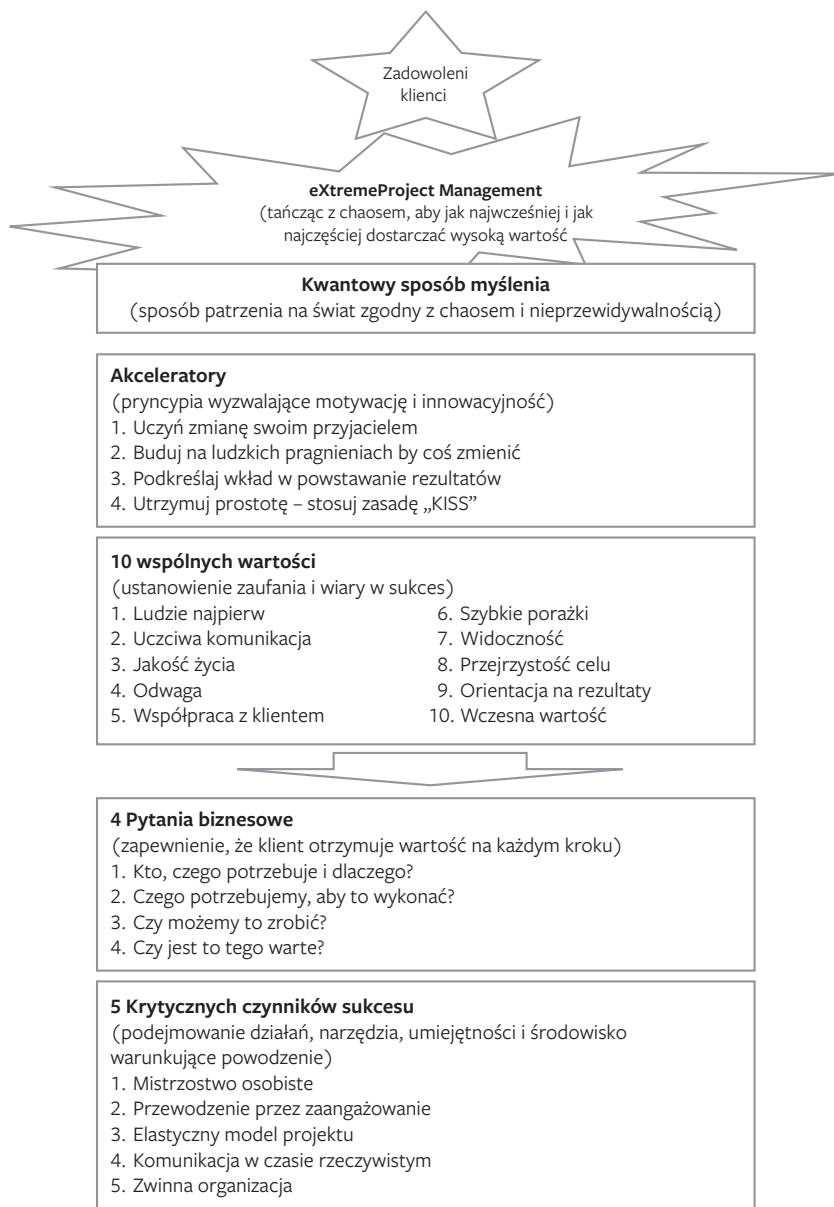
¹⁷¹ *Ibid.*, s.11.

¹⁷² *Ibid.*, s. 41.

¹⁷³ *Ibid.*, s. 42.

- 3) Jakość życia – zachowanie w projekcie równowagi między życiem zawodowym i prywatnym.
- 4) Odwaga – praca mimo strachu, niepewności i trudności w trakcie trwania projektu.

Rysunek 27. Struktura podejścia xPM



Wartości dotyczące procesów:

- 5) Współpraca z klientem – ciągła interakcja i informacje zwrotne uzyskiwane od aktywnego, współuczestniczącego w projekcie klienta.
- 6) Szybkie porażki – najlepszym sposobem na odkrycie błędów jest zmierzenie się już na początku z najtrudniejszym i najbardziej ryzykownym zakresem prac.
- 7) Widoczność – utrzymywanie otwartej i przejrzystej komunikacji o wszystkich aspektach projektu.

Wartości dotyczące biznesu:

- 8) Przejrzystość celu – zrozumienie celu projektu, jak również jego szerszego kontekstu.
- 9) Orientacja na rezultaty – skupienie na dostarczaniu efektów pracy, a nie mierzeniu jej postępów.
- 10) Wczesna wartość – dostarczanie klientowi wartościowych dla niego rezultatów tak szybko, jak to możliwe.

Pytania biznesowe

Pytania biznesowe służą interesariuszom jako rodzaj latarni morskiej, ciągle przypominając o celu i kierunku projektu. Są one związane z jego uzasadnieniem biznesowym, które okresowo powinno być przeglądane i aktualizowane. Pytania te sformułowano następująco¹⁷⁴:

- 1) Kto, czego potrzebuje i dlaczego?
- 2) Czego potrzebujemy, aby to wykonać?
- 3) Czy możemy to zrobić?
- 4) Czy jest to tego warte?

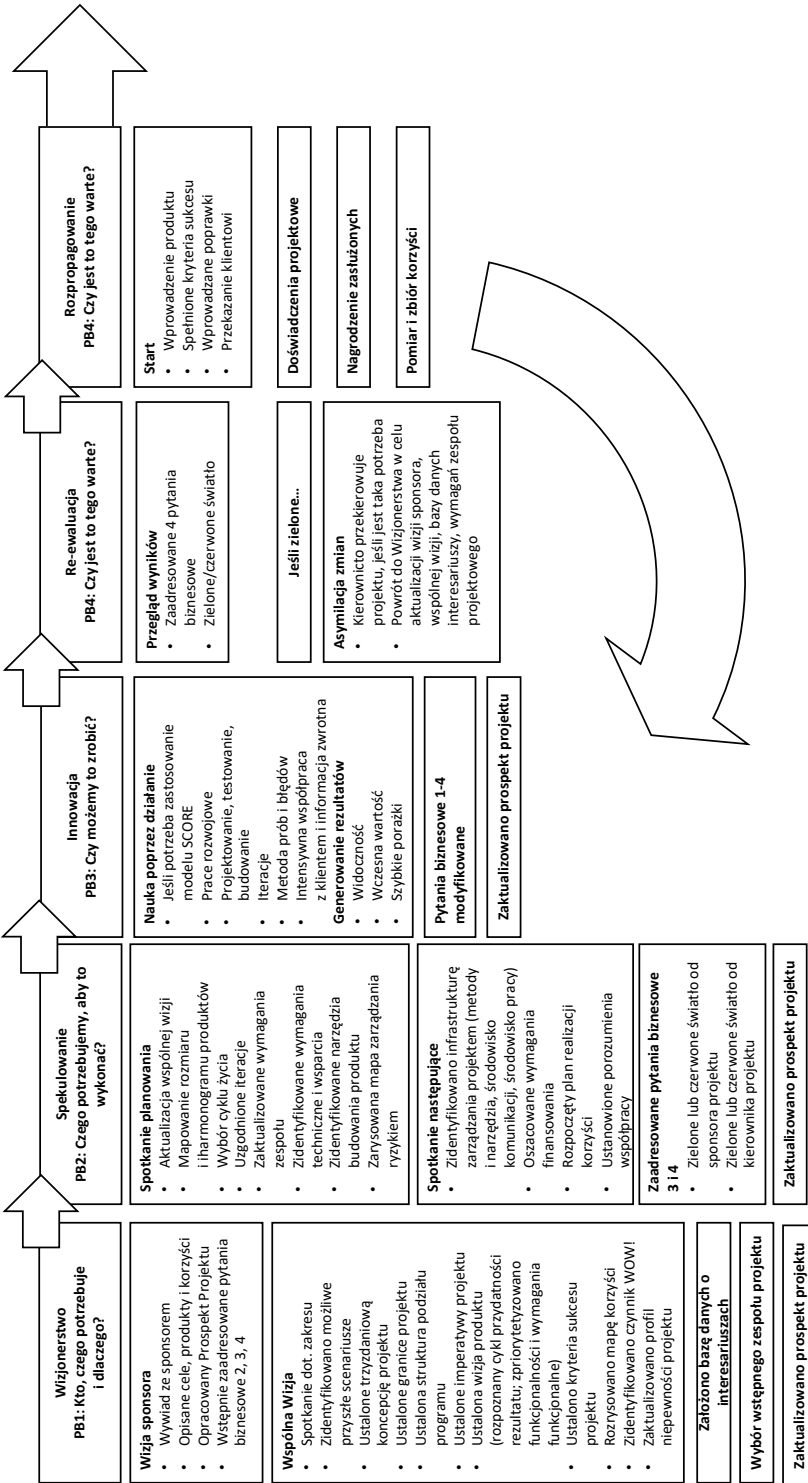
Krytyczne czynniki sukcesu

Krytyczne czynniki sukcesu mówią o działaniach materializujących się podczas ekstremalnego zarządzania projektem, począwszy od jego rozpoczęcia, aż do zakończenia. Wśród nich znalazły się:

- 1) Mistrzostwo osobiste (and. *self-mastery*) – zdolność do zachowania spokoju, opanowania i kontroli nad sobą, projektem i innymi aspektami życia, nawet w sytuacji projektu realizowanego w ekstremalnych warunkach.
- 2) Przewodzenie przez zaangażowanie – czynnik związany z uzyskaniem i utrzymaniem zaangażowania w projekt ze strony innych osób i uzależniający sukces od umiejętności przewodzenia ze strony kierownika projektu.

¹⁷⁴ Ibid., s. 43.

Rysunek 28. Model cyklu życia XPM



Źródło: ibid., s.225

- 3) Elastyczny model projektu – iteracyjny cykl życia projektu ekstremalnego składający się z czterech cyklicznych faz: wizji (ang. *visionate*), spekulacji (ang. *speculate*), innowacji (ang. *innovate*) i oceny (ang. *reevaluate*) oraz fazy kończącej – rozpowszechniania (ang. *disseminate*).
- 4) Komunikacja w czasie rzeczywistym – szybkie i ekstremalne projekty wymagają błyskawicznego przekazywania informacji, współdzielenia dokumentów oraz bezwzględnego podejmowania decyzji; przyjęte metody i narzędzia komunikacji powinny zapewnić wszystkim interesariuszom dostęp do niezbędnych podczas podejmowania decyzji informacji.
- 5) Zwinna organizacja – ostatni czynnik sukcesu projektu dotyczy środowiska organizacji, w którym realizowany jest projekt – powinno ono być równie zwinne, co ekstremalne; wśród pożądanych cech organizacji DeCarlo wyszczególnia: tolerancyjność na zmiany oraz kulturę sprzyjającą specyficznym wymaganiom zarządzania projektami ekstremalnymi.

Jak można zauważyć, większość elementów metodyki XPM związana jest z miękkimi aspektami zarządzania projektem obejmującymi pryncypia, wartości, przekonania i zalecenia dotyczące postaw i zachowań interesariuszy projektów. Sposób realizacji projektu, w rozumieniu etapów postępowania oraz technik i narzędzi zawarty jest w czynniku sukcesu określonym jako „elastyczny model projektu”. Zakres poszczególnych iteracji oraz ich układ przedstawiono na poniższym schemacie.

2.8.2. Podsumowanie

Metodyka ekstremalnego zarządzania projektami jest ważnym głosem w dyskusji na temat problemów realizacji projektów w turbulentnym otoczeniu. W naturalny sposób, poprzez zbieżność nazw, można ją odnieść do metodyki XP. Obie przypisują sobie ekstremalność, jednak jej źródła są zgoła różne. O ile w metodyce XP polega ona na doprowadzeniu praktyk developerskich do ekstremalnego stanu (poprzez programowanie w parach, ciągłą refaktoryzację itp.), o tyle XPM wiąże ekstremalność z heroizmem kierownika projektu stojącego naprzeciw trudnego i złożonego projektu realizowanego w skrajnie nieprzyjawnym środowisku niedojrzałej projektowo organizacji. W strukturze XPM widać wyraźnie dominację zaleceń dotyczących miękkich aspektów realizacji projektów: motywacji do działania, wiary w sukces, przywództwa, relacji z interesariuszami. Mniej tam elementów zwinnego zarządzania projektami, a więcej odpowiedzi na frustrację, stres i wyczerpanie kierowników projektów „ekstremalnych”.

Z formalnej perspektywy XPM bardziej przystaje do poradnika motywacyjnego (szkoły przetrwania) dla kierowników projektów, poruszającego jakże ważne problemy codziennej pracy z interesariuszami niż do określonej metody postępowania w organizacjach.

DeCarlo zawarł w XPM silnie idealistyczną, humanistyczną i osobistą diagnozę skrajnie trudnych warunków realizacji projektów, jednakże proponowane zalecenia mają charakter ogólny, a co za tym idzie: uniwersalistyczny i zawsze aktualny.

2.9. AgilePM (Dynamic System Development Method)

Wśród zwinnych metod zarządzania projektami z pewnością warto zwrócić także uwagę na metodykę AgilePM¹⁷⁵. Cechą ją wyróżniającą jest dużo szerszy niż w przypadku np. SCRUM czy FDD kompleksowy charakter zaleceń uwzględniający nie tylko iteracyjno-przyrostowy model wytwarzania produktu, ale również jego kontekst biznesowy i integrację w środowisku organizacji.

Swoimi korzeniami AgilePM sięga połowy lat 90. XX w., kiedy to powstawała metodyka Dynamic System Development Method (DSDM). Nad rozwiązaniem tym pracował zespół licznych praktyków reprezentujących zaangażowane firmy. Wspólnie stworzyły one związek pod nazwą DSDM Consortium (Agile Business Consortium, ABC), zajmujący się popularyzacją i rozwojem podejścia – początkowo na własne, a potem także na potrzeby społeczności korzystającej od 2007 r. z upublicznionego standardu¹⁷⁶. W kolejnych latach na podstawie DSDM opracowano uniwersalną metodykę nazwaną AgilePM, pozwalającą w lepszy sposób wykorzystać potencjał zaleceń DSDM w zarządzaniu w organizacyjnym środowisku przedsiębiorstw. Należy wspomnieć, iż w ramach pakietu rozwiązań ABC obok AgilePM funkcjonuje również AgilePgM – czyli zwinne zarządzanie programami oraz AgilePMO – model działania zwinnych biur zarządzania projektami¹⁷⁷.

2.9.1. Zalecenia metody

Struktura metodyki AgilePM oparta jest na modelu ateńskiej świątyni: fundamencie, filarów, stropu i dachu. Idąc od dołu, fundamentem metodyki jest „zdrowy rozsądek i pragmatyzm”. Autorzy metodyki podkreślają, że dzięki nim projekt nie traci kontaktu z rzeczywistością, opiera się nie na dogmatach i teorii, ale rozwiązywaniu praktycznych problemów, a przede wszystkim jest prowadzony w sposób zgodny z oryginalnymi założeniami Manifestu zwinności. Na takim fundamencie osadzono cztery filary DSDM-u, którymi są: procesy, ludzie, produkty oraz praktyki. Nad nimi osadzono tzw. pryncypia oraz zwieńczającą je filozofię metodyki.

¹⁷⁵ Znaki handlowe DSDM, Athern, AgilePM są zarejestrowanymi znakami handlowymi Agile Business Consortium Limited.

¹⁷⁶ Agile Business Consortium, *Agile PM. Agile Project Management Handbook v2*. Wydanie polskie, Agile Business Consortium, Ashford 2019, s. 10.

¹⁷⁷ DSDM Consortium, *The Agile PMO*, Agile Business Consortium, Kent 2016.

Filozofia DSDM sformułowana została w formie właściwej wizji zarządczej. Mówi ona, iż „najlepsza wartość biznesowa wyłania się, gdy projekty są powiązane z jasnymi celami biznesowymi, dostarczają często, a także angażują do współpracy ludzi zmotywowanych i odpowiednio umocowanych”¹⁷⁸. Takie hasło i swoisty „manifest” przyświeca autorom metodyki i powinien towarzyszyć także zespołom pracującym zgodnie z zaleceniami metody.

Nakreślonej wizji towarzyszą określone pryncypia uszczegóławiające ją do postaci ośmiu kierunkowych zaleceń, będących *de facto* regułami heurystycznymi:

- 1) Koncentruj się na potrzebie biznesowej.
- 2) Dostarczaj na czas.
- 3) Współpracuj.
- 4) Nigdy nie idź na kompromis w kwestii jakości.
- 5) Buduj przyrostowo od solidnych podstaw.
- 6) Rozwijaj iteracyjnie.
- 7) Komunikuj się ciągle i jasno.
- 8) Demonstrować kontrolę¹⁷⁹.

Podobnie jak w przypadku np. pryncypiów PRINCE2 zalecenia te mają sugerować postawy, kulturę, zachowania zespołu. Mimo ogólnego nakreślenia stanowią one istotny i niepomijalny element metodyki.

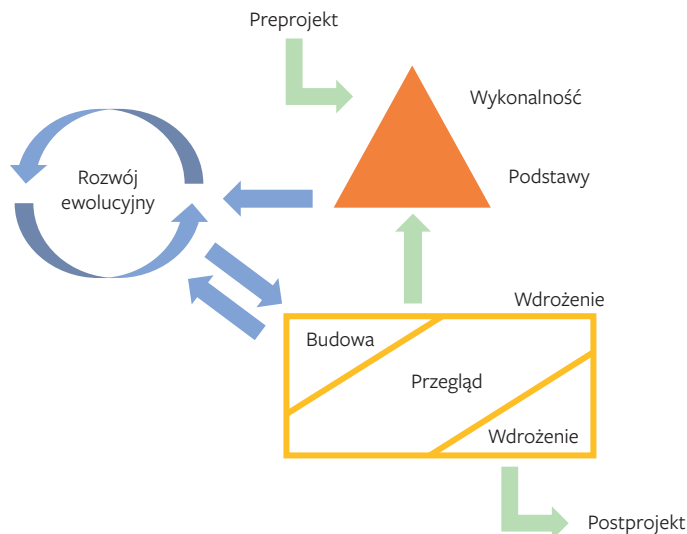
Z perspektywy „mechaniki” prezentowanego rozwiązania, kluczowe znaczenie mają w nim zalecenia dotyczące procesu DSDM. W tym zakresie metodyka proponuje szersze podejście niż większość innych metodyk zwinnych, gdyż obejmuje ona swoim zasięgiem cały cykl życia projektu: począwszy od jego zainicjowania, przez planowanie i organizację, budowanie rozwiązania, po wdrożenie oraz fazę po projektową. Metodyka zachowuje przy tym elastyczność i swój iteracyjny charakter oraz może być dość swobodnie konfigurowana w zależności od rozmiaru, zmienności rozwiązania oraz wymagań względem częstotliwości i zakresu kolejnych wydań.

Faza „przed-projektem” (ang. *pre-project*) związana jest przede wszystkim z podejmowaniem decyzji związanych z uruchomieniem projektu, odniesieniem go do strategii oraz identyfikacją celów i problemów, które ma rozwiązać. W tej fazie powołuje się również osoby pełniące role sponsora biznesowego oraz wizjonera biznesowego. Faza ta ma zapewnić, iż realizujemy właściwe i dobrze zorganizowane projektu. Na tym etapie tworzone są również tzw. warunki odniesienia (ang. *terms of reference*) określające kontekst biznesowy, cele projektu oraz zakres i uzasadnienie kolejnej fazy wykonalności¹⁸⁰. Zgodnie z zaleceniami autorów metodyki „przed-projektem” powinno trwać relatywnie krótko.

¹⁷⁸ Agile Business Consortium, *Agile PM...*, s. 16.

¹⁷⁹ *Ibid.*, s. 20–21.

¹⁸⁰ *Ibid.*, s. 39.

Rysunek 29. Model procesu DSDM

Źródło: opracowanie własne na podst. Agile Business Consortium, *Agile PM. Agile Project Management Handbook v2*. Wydanie polskie, Agile Business Consortium, Ashford 2019, s. 10.

Faza „wykonalność” (ang. *feasibility*) określona została w celu zweryfikowania zasadności i szans zbudowania pożądanego rozwiązania (wykonalność techniczna oraz uzasadnienie biznesowe). W tej fazie opracowuje się także zarys podejścia do realizacji projektu, jak również jego organizację, zarządzanie oraz wstępne szacunki projektu.

Faza „podstawy” (ang. *foundations*) służy dogłębnemu zrozumieniu czekającej zespół pracy. Zrozumienie to powinno uwzględniać zarówno wymiar techniczny budowanego rozwiązania (definicja architektury rozwiązania, definicja podejścia do rozwoju), jak również zarządczy (plan dostarczania oraz definicja podejścia do zarządzania) oraz biznesowy (szczegółowe uzasadnienie biznesowe, lista wymagań z priorytetami). Celem tej fazy jest zbudowanie wysokopoziomowych, ale solidnych podstaw jego realizacji.

Faza „rozwój ewolucyjny” (ang. *evolutionary development*) polega na bieżącej pracy zespołu nad budowaniem przyrostów rozwiązania w modelu ewolucyjnym i przyrostowym. Metodyka rekomenduje w tym zakresie liczne techniki pomocne w pracy zespołu, m.in.: MoSCoW, timeboxy, warsztaty facylitowane, modelowanie i rozwój iteracyjny.

Faza „wdrożenie” (ang. *deployment*) zakłada, że budowane rozwiązanie powinno zostać przekazane do tzw. biznesu, wprowadzone do użytku i uruchomione operacyjnie. Metodyka zaleca stosowanie kolejnych wydań rozwiązania wdrażanych, aż do osiągnięcia pożądanego kształtu końcowego.

Ostatnią fazą projektu zgodnie z DSDM jest „po projekcie” (ang. *post-project*). Rozpoczyna się po ostatnim wydaniu rozwiązania. Jej celem jest ocena przedsięwzięcia

z perspektywy dostarczonej wartości biznesowej, a w szczególności założeń opisanych w uzasadnieniu biznesowym. Według autorów metody zwykle ma to miejsce między 3. a 6. miesiącem od zakończenia projektu¹⁸¹.

Istotą proponowanego przez DSDM modelu jest układ powyższych faz oparty na iteracyjności. O ile uruchomienie projektu i przejście od „przed-projektem”, przez „wykonalność” do „podstaw” i „rozwoju ewolucyjnego”, a w konsekwencji do „wdrożenia” będzie przebiegać naturalnie i w sposób sekwencyjny, o tyle metoda wskazuje także możliwości powrotu od „wdrożenia” do „podstaw” lub „rozwoju ewolucyjnego”. W efekcie metodyka oferuje kierownikowi projektu i organizacji elementy składowe cyklu („klocki”), natomiast pozostawia dużą swobodę w ich sekwencjonowaniu. Fazy te można w rezultacie ułożyć liniowo, w zasadzie wracając do modelu kaskadowego, jak też skrócić i zdynamizować, tworząc wysoce iteracyjne i zwinne środowiska zarządcze. Taki duży uniwersalizm metodyki stanowi jej istotną zaletę.

Oprócz ciekawego modelu cyklu życia metodyka prezentuje również rozbudowaną strukturę ról zaangażowanych w realizowany projekt (tabela 9). Układ proponowanych ról oddaje złożoność środowiska zarządzania projektem i dalece wykracza poza np. zalecenia w tym względzie metodyki SCRUM. AgilePM proponuje bowiem, aż 13 (!) ról w zespole DSDM. Role te uwzględniają różne poziomy zarządzania projektem oraz reprezentowane interesy.

Tabela 9. Kategorie ról w metodyce Agile PM

Reprezentowane interesy Poziom roli	Biznesowe	Techniczne	Zarządzania	Procesu
Poziom projektu	sponsor biznesowy, wizjoner biznesowy, analityk biznesowy	koordynator techniczny	kierownik projektu	
Poziom zespołu rozwoju rozwiązania	ambasador biznesowy, analityk projektu	twórca rozwiązania, tester rozwiązania, analityk projektu	lider zespołu	
Poziom wspierający	doradca biznesowy	doradca techniczny		facylitator warsztatów, coach DSDM

Źródło: opracowanie własne na podst. Agile Business Consortium, *Agile PM...*, s. 10.

Kompleksowość metodyki odnajdziemy również w zaleceniach dotyczących dokumentowania projektu. Dostarcza ona w tym zakresie 14 elementów, tzw. produktów DSDM. Należy jednocześnie podkreślić, iż zdaniem autorów „nie wszystkie produkty wymaga-

¹⁸¹ DSDM Consortium, *Agile Project Management Handbook. Version 1.2*, DSDM Consortium 2013.

ne są w każdym projekcie, a formalizm łączący się z każdym produktem będzie zmieniał się w zależności od projektu i organizacji”¹⁸². Wśród produktów DSDM znajdziemy¹⁸³:

- 1) warunki odniesienia (ang. *terms of reference*),
- 2) uzasadnienie biznesowe (ang. *business case*),
- 3) listę wymagań z priorytetami (ang. *prioritised requirements list*),
- 4) definicję architektury rozwiązania (ang. *solution architecture definition*),
- 5) definicję podejścia do rozwoju (ang. *development approach definition*),
- 6) plan dostarczania (ang. *delivery plan*),
- 7) definicję podejścia do zarządzania (ang. *management approach definition*),
- 8) ocenę wykonalności (ang. *feasibility assessment*),
- 9) podsumowanie podstaw (ang. *foundation summary*),
- 10) ewoluujące rozwiązanie (ang. *evolving solution*),
- 11) plan timeboxa (ang. *timebox plan*),
- 12) zapis przeglądu timeboxa (ang. *timebox review record*),
- 13) raport z przeglądu projektu (ang. *project review report*),
- 14) ocenę korzyści (ang. *benefits assessment*).

W obrębie metodyki każdy z powyższych produktów przypisany został do określonej fazy projektu; oprócz szczegółowego opisu treści i przeznaczenia wskazano również macierze RACI przypisujące poszczególnym produktom związane z ich stosowaniem role. Kompleksowość metody odnajdziemy również w obszernym zestawie tzw. praktyk DSDM pomagających kierownikom projektów zwinnie realizować prowadzone przez siebie przedsięwzięcia. Będą to: priorytetyzacja MoSCoW, stosowanie timeboxów, warsztaty facylitowane, modelowanie oraz rozwój iteracyjny¹⁸⁴. W podręczniku dużo miejsca poświęcono także na zapewnienie efektywnej współpracy w zespole projektowym, definiowaniu wymagań i historii użytkownika oraz szacowaniu w projekcie. Dodatkowo poruszono w nim zagadnienia związane z planowaniem projektów w cyklu życia, zapewnienia jakości, zarządzania ryzykiem, a także dopasowania metodyki do potrzeb organizacji.

2.9.2. Podsumowanie

Metodyka DSDM/Agile PM jest rozwiązaniem, które relatywnie rzadko spotykane jest w dzisiejszych przedsiębiorstwach¹⁸⁵. Nie uzyskało ono takiej popularności jak np. SCRUM czy XP. Warto jednakże pamiętać o niewątpliwych mocnych stronach tego

¹⁸² Agile Business Consortium, *Agile PM...*, s. 38.

¹⁸³ *Ibid.*, s. 39–40.

¹⁸⁴ *Ibid.*, s. 49–64.

¹⁸⁵ D. West et al., *Agile development: Mainstream adoption has changed agility*, “Forrester Research” 2020, vol. 2, no. 1; A. Komus et al., *Study Status Quo (Scaled) Agile 2019/20*, <https://www.hs-koblenz.de/en/bpm-labor/status-quo-scaled-agile-2020> (dostęp: 14.06.2021).

rozwiązania. Przede wszystkim jest to metodyka o niewątpliwie kompleksowym charakterze, w której w centrum uwagi pozostaje projekt oraz odpowiedzialny za niego kierownik. Jest to znaczna przewaga, gdyż większość innych rozwiązań kładzie nacisk albo na prace *stricte* programistyczne (np. XP, TDD, FDD, Lean), albo na poziom zespołu projektowego i kierowania jego pracą (np. SCRUM, LeanStartup, Kanban). Poza zakresem tych metodyk pozostają istotne przecież w praktyce działania przedsiębiorstwa zagadnienia dotyczące zainicjowania projektu, oceny korzyści, uzasadnienia jego realizacji czy planowania przebiegu – nie tylko z perspektywy kolejnych iteracji wytwórczych, ale i całego cyklu życia projektu. Metodyka AgilePM pozostaje w nurcie zwinności, spełniając wszystkie kryteria dla tej rodziny rozwiązań. Jednocześnie jednak stanowi znaczące ich uzupełnienie o kontekst zarządczy. Jest również metodyką opracowaną z punktu widzenia kierownika projektu i jako narzędzie w jego rękach. Wieloletnie doświadczenia praktycznego stosowania metod zwinnych pokazały bowiem, że mimo początkowych prób usunięcia tej roli z zespołów projektowych czy rozbicia zadań między inne osoby, to nadal kierownik projektu pozostaje istotnym stanowiskiem, mającym określone zadania w środowisku przedsiębiorstwa i projektu zwinnego. Zastanawiający jest przy tym fakt, iż współcześnie potrzeby o charakterze biznesowym i organizacyjnym częściej zaspokajane są poprzez wdrażanie metod skalowania, takich jak np. SAFe, a nie przez DSDM/AgilePM, który nie zdobyła takiej popularności. Nie zmienia to faktu, iż metodyka ta pozostaje nadal aktualnym i cennym zbiorem zaleceń co do realizacji projektów zwinnych.

2.10. Agile PMI – Software Extension to the PMBOK Guide Fifth Edition

Ostatnia dekada rozwoju zarządzania projektami związana jest zdecydowanie z rozwojem metodyk zwinnych. Można wręcz mówić o kolejnej fazie rozwoju całej dziedziny, w której aktualnie to właśnie problemy realizacji projektów informatycznych i metodyki zwinne angażują umysły profesjonalistów w dziedzinie zarządzania projektami.

Dynamiczny rozwój metodyk zwinnych nie mógł pozostać poza zainteresowaniem największego, globalnego stowarzyszenia skupiającego osoby zajmujące się zarządzaniem projektami, czyli Project Management Institute (PMI). PMI jest organizacją non profit założoną w 1969 r. w Pensylwanii. Od początku swojej działalności stanowi platformę wymiany wiedzy i doświadczeń dla specjalistów zarządzania projektami. PMI pełni także rolę regulatora, podejmując liczne działania związane z opisem najlepszych praktyk realizacji przedsięwzięć w formie standardów. Organizacja zrzesza obecnie ponad 440 tys. członków w 104 krajach, w których istnieją 273 oddziały (PMI Chapter). Od 1998 r. PMI jest akredytowany przez American National Standards Institute (ANSI) jako twórca standar-

dów w zakresie zarządzania projektami – tzw. Accredited Standards Developer¹⁸⁶. O sile wpływu PMI na środowisko zarządzania może świadczyć fakt, że najważniejszy standard zarządzania projektami – PMBoK Guide – od chwili publikacji pierwszej wersji w 1986 r. został sprzedany w ponad 4,6 mln egzemplarzy, zaś certyfikat Project Management Professional oparty na PMBoK-u posiada 625 tys. osób na całym świecie¹⁸⁷.

Mimo popularyzacji podejść zwinnych od końca lat 90. XX w. PMI podjął pracę nad opracowaniem własnego standardu w tym zakresie dopiero w drugiej dekadzie wieku XXI. Wtedy to instytut rozpoczął współpracę z wiodącą organizacją w środowisku branży informatycznej: The Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS). W jej efekcie 10 września 2013 r. ukazał się wspólny standard pod nazwą: *Software Extension to the PMBOK® Guide – Fifth Edition* (w skrócie SWX). Zgodnie ze swoją nazwą SWX stanowi dodatek rozszerzający wiedzę zawartą w najnowszym, piątym wydaniu standardu PMBoK Guide o elementy pomocne przy realizacji projektów informatycznych. Oba te standardy mogą być stosowane razem i mają komplementarny, a nie wykluczający charakter. Przygotowując się do wydania tegoż standardu PMI zmodyfikował treści PMBoK Guide, rozszerzając podejście do cyklu życia projektu o tzw. *project lifecycle continuum* – poczynawszy od podejścia planistycznego (ang. *predictive/plan-driven*), poprzez cykle iteracyjne i inkrementalne (ang. *iterative and incremental*), aż do adaptacyjnych (ang. *adaptive life cycles*)¹⁸⁸. SWX, skierowane przede wszystkim (choć nie tylko) do menedżerów projektów informatycznych, stanowi zatem źródło wiedzy o metodach, narzędziach i technikach pomocnych przy zarządzaniu projektami tworzenia oprogramowania we wszystkich wariantach cyklu życia projektów.

Mimo, że SWX wydaje się bezpośrednią odpowiedzią PMI na wzrost popularności metodyk zwinnych, to sami autorzy zdają się unikać wprost odniesienia do *agile*, co argumentują różnorodnym znaczeniem takiego terminu. W zamian za to proponują swój standard jako zbiór „najlepszych praktyk” pomocnych w projektach informatycznych, w szczególności uwzględniając adaptacyjne podejścia do ich realizacji. Standard SWX w ich intencji pozycjonowany jest zatem jako element nadrzędny, rama odniesienia względem innych metod i metodyk realizacji projektów.

¹⁸⁶ Wśród najważniejszych standardów PMI można wskazać m.in. A Guide to the Project Management Body of Knowledge PMBOK® Guide Fifth Edition 2013, Construction Extension to the PMBOK® Guide Third Edition, Government Extension to the PMBOK® Guide Third Edition, Software Extension to the PMBOK® Guide Fifth Edition, Practice Standard for Earned Value Management – Second Edition, Practice Standard for Project Configuration Management, Practice Standard for Work Breakdown Structures – Second Edition, Practice Standard for Scheduling Second Edition, Practice Standard for Project Estimating, Practice Standard for Project Risk Management, Practice Standard for Requirements Management, The Standard for Program Management – Third Edition, Project Manager Competency Development Framework – Second Edition, Organizational Project Management Maturity Model (OPM3®) – Third Edition czy The Standard for Portfolio Management – Third Edition.

¹⁸⁷ PMI, „PMI Today”, October 2014, s. 4.

¹⁸⁸ PMI, A Guide to the Project Management Body of Knowledge (PMBOK Guide), Fifth Edition, Project Management Institute 2013, s. 38–46.

2.10.1. SWX – założenia wstępne

Struktura SWX zawarta w oficjalnym podręczniku zbliżona jest do struktury PMBoK Guide. W pierwszym rozdziale, który ma charakter wprowadzający, przedstawione są podstawowe zagadnienia wspólne dla wszystkich praktyków zarządzania projektami, ale ukazane tym razem z perspektywy specyfiki projektów informatycznych. Są to: zdefiniowanie projektów oraz dziedziny zarządzania projektami, określenie relacji między portfelem, programem i projektami, omówienie organizacyjnych zagadnień zarządzania projektami oraz m.in. roli kierownika projektu.

W kolejnej części omówione zostały szerzej aspekty organizacyjne (kultura, komunikacja, struktury oraz aktywa procesów organizacyjnych i czynniki środowiska organizacyjnego), interesariusze i ład organizacyjny, zalecenia dotyczące różnorodnych wariantów zespołu projektowego. Stosunkowo dużo uwagi poświęcono w SWX na omówienie poszczególnych wariantów cyklu życia projektu. Ich zróżnicowanie przedstawiono na poniższym schemacie:

Rysunek 30. Kontinuum cyklu życia projektów informatycznych



Źródło: PMI, *Software Extension to the PMBOK® Guide – Fifth Edition*, Newtown Square 2013, s. 26.

Podejście do wytwarzania oprogramowania rekomendowane przez SWX opiera się na kilku wariantach cyklu życia projektów.

Pierwszy określa tzw. cykle predyktywne (ang. *predictive*) – lub w bardziej swobodnym tłumaczeniu: „presytuacyjne”, „planistyczne”, „zdeteminowane”, „liniowe”. W takim wa-

riancie poszczególne fazy projektu następują po sobie w sposób sekwencyjny, przy niewielkim „zachodzeniu” na siebie. Poszczególne parametry projektu (zakres, czas, budżet) określone i definiowane są tak wcześnie, jak to tylko możliwe. Według autorów standardu taki sposób realizacji przynosi największe korzyści w przypadku projektów o dobrze zdefiniowanych wymaganiach, realizowanych w ramach dobrze poznanej dziedziny, stabilnej technologii oraz dla poznanego klienta.

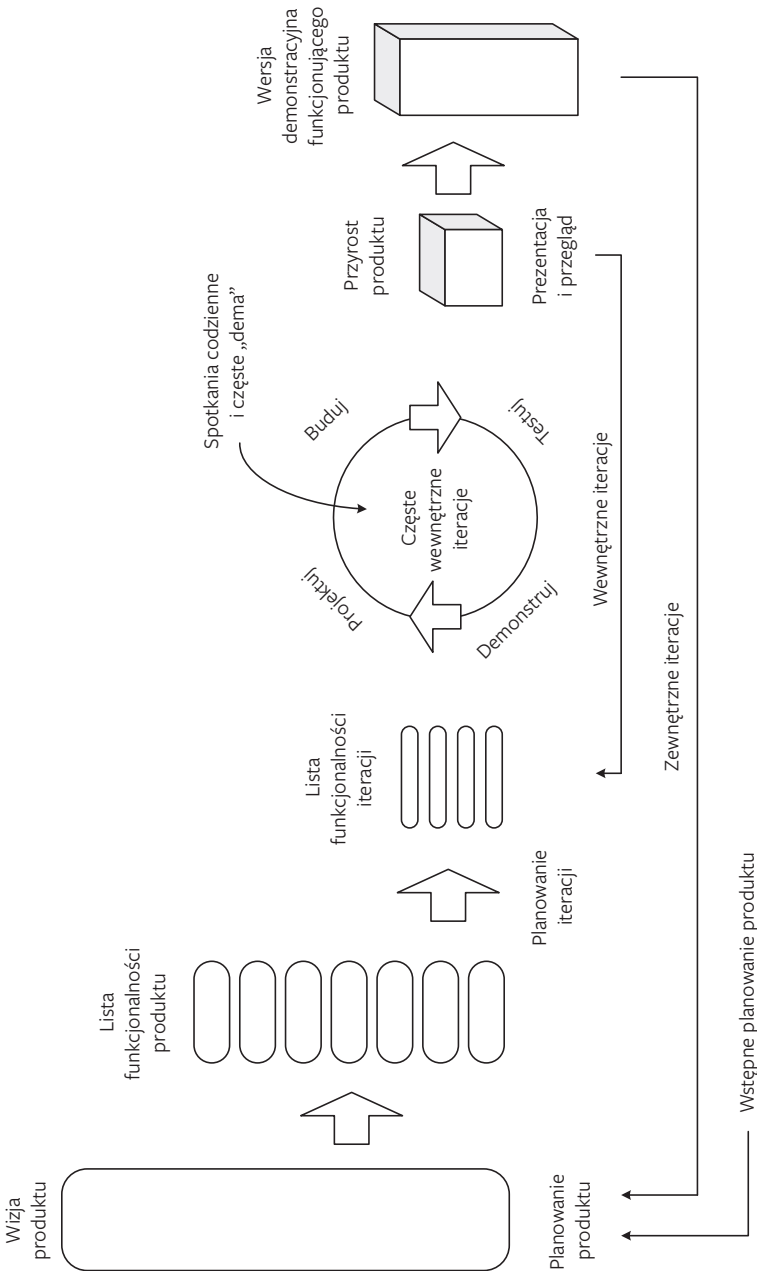
Drugi wariant cyklu życia to cykle iteracyjne i inkrementalne. Ich cechą szczególną jest określanie zakresu na początku projektu, a szacunków czasu i kosztów w późniejszych fazach, wraz z wzrostem wiedzy zespołu o tworzonej produkcie. Zespół projektowy dysponuje większą swobodą wprowadzania zmian do projektu, ale musi mieć świadomość relacji wymiennych pomiędzy poszczególnymi parametrami i ograniczeniami projektu. Mimo że autorzy SWX stwierdzają, iż większość cykli życia projektów łączy iteracyjność i inkrementalność, to podkreślają jednocześnie odmiennność tych dwóch terminów. Iteracyjność ma w ich rozumieniu być związana z procesem tworzenia rozwiązań i powtarzaniem jednej lub kilku faz projektu. W efekcie możliwe jest uzyskanie informacji zwrotnej, poznanie nowych wymagań lub ich modyfikacja. Praca wykonywana w iteracjach może dotyczyć tego samego zakresu, doskonalenia jego jakości itp. Za to pojęcie inkrementalności charakteryzuje sposób rozwoju produktu, który budowany jest poprzez kolejne przyrosty, czyli funkcje i funkcjonalności rozszerzające zakres projektu. Podejście przyrostowe pozwala skupić się zespołowi na elementach dających największą wartość klientowi, a temu drugiemu na uzyskanie ich stosunkowo wcześnie, nie czekając na koniec projektu. Co ciekawe, autorzy SWX podkreślają, że podejście przyrostowe może być elementem zarówno cykli predyktywnych, gdzie bardzo wcześnie określamy i planujemy układ kolejnych przyrostów, jak i adaptacyjnych, gdzie zakres przyrostów może być repriorytetyzowany i modyfikowany pomiędzy fazami.

Trzeci, skrajny wariant cykli życia projektów to tzw. cykle adaptacyjne (ang. *adaptive*), inaczej zwane zwinnymi (ang. *agile*) czy nakierowanymi na zmiany (ang. *change-driven*). Według autorów termin „zwinny” nie powinien być przywoływany w odniesieniu do nazwy cykli życia, ale do określenia pewnych jego charakterystyk wspólnych dla różnych cykli adaptacyjnych. Cechami tymi są m.in.¹⁸⁹:

- 1) regularnie dostarczane przyrosty działającego oprogramowania,
- 2) krótkie (dzienne, tygodniowe, miesięczne) okresy adaptacyjnych iteracji,
- 3) cykl życia projektu składający się z adaptacyjnych iteracji zwykle o tym samym czasie trwania (ang. *time-boxing*),
- 4) rozróżnienie iteracyjności od inkrementalności – nie wszystkie iteracje muszą kończyć się działającym fragmentem oprogramowania,

¹⁸⁹ PMI, *Software Extension to the PMBOK® Guide – Fifth Edition*, Newtown Square 2013, s. 33.

Rysunek 31. Adaptacyjny model wytwarzania oprogramowania



Źródło: *ibid.*, s. 35.

- 5) wymagania, projektowanie i sam produkt powstaje w miarę ewoluowania projektu,
- 6) przedstawiciele klienta, użytkownicy pozostają stale zaangażowani w projekt; m.in. poprzez uczestniczenie w okresowych i końcowych demonstracjach produktu; przedstawiciel klienta współpracuje z zespołem i dostarcza mu wiedzy odnośnie do pożądanego kierunku rozwoju produktu i istniejących ograniczeniach projektu.
- 7) realizacja projektu przez małe (do 10 osób), samoorganizujące się zespoły; w przypadku dużych projektów stosowane struktury gronowej (wiele, małych zespołów),
- 8) praca w warunkach jednozadaniowości – każdy członek zespołu pracuje tylko nad jednym projektem w danym czasie,
- 9) zespoły projektowe łączą osoby o kompetencjach uniwersalnych oraz specjalistów; do zespołu, w miarę potrzeb zapraszani są eksperci funkcjonalni.

W praktyce poszczególne cechy mogą występować w różnej konfiguracji i natężeniu w zależności od specyfiki projektu i jego otoczenia. Według autorów SWX uzasadnienie dla stosowania adaptacyjnych cykli życia projektów pojawia się, gdy trudno jest na wczesnych etapach projektu dokładnie określić potrzeby i oczekiwania klienta lub gdy mamy do czynienia z innowacyjnym zastosowaniem technologii.

Jako ilustrację podejścia adaptacyjnego standard SWX przytacza załączony powyżej schemat (rysunek 31). Ma on charakter generyczny i wprawiony czytelnik z pewnością dostrzeże w nim analogię to metod zwinnego zarządzania projektami takich jak: SCRUM, XP, FDD, TDD i DSDM.

2.10.2. SWX – zalecenia właściwe

Ze względu na funkcjonowanie SWX w charakterze rozszerzenia dla uniwersalnego standardu PMBoK Guide, w głównej części standardu jego strukturę podporządkowano najnowszej, piątej edycji kompendium wiedzy. Przyjęto zatem istniejących dziesięć obszarów wiedzy oraz 47 procesów zarządzania projektami z PMBoK Guide, dodając do ich opisów zagadnienia specyficzne dla projektów tworzenia oprogramowania. Podobnie też istniejące opisy procesów oparte na zasileniach/wejściach (ang. *inputs*), narzędziach i technikach (ang. *tools and techniques*) oraz produktach/wyjściach (ang. *outputs*) zostały uzupełnione o dodatkowe elementy stosowane w tego typu przedsięwzięciach. Co ważne, wszystkie zalecenia metodyczne zawarte w oryginalnym PMBoK Guide pozostają w mocy i traktowane są jako uniwersalne, a więc przydatne także w przypadku projektów informatycznych. Ponieważ omówienie klasycznych metodyk zarządzania projektami (w tym PMBoK Guide) leży w zakresie innego podzadania projektu, w dalszej części opracowania autor skupi się wyłącznie na nowych rekomendacjach wniesionych przez dodatek SWX w dotychczasowym układzie obszarów wiedzy o zarządzaniu projektami.

Zarządzanie integracją projektu

Obszar zarządzania integracją projektu jest kluczowym obszarem, koniecznym do zapewnienia powodzenia w jego realizacji. Jest to obszar, w ramach którego wykonywane są podstawowe działania (procesy) konieczne do identyfikowania, definiowania, łączenia i koordynacji wszystkich aktywności wykonywanych w realizowanym przedsięwzięciu. Procesy w ramach zarządzania integracją można zatem porównać do centralnego układu nerwowego – mózgu i rdzenia kręgowego – które zapewniają sprawną współpracę i integrację poszczególnych części ciała tak, aby organizm funkcjonował jako jedna całość. W odniesieniu do specyfiki projektów informatycznych warto podkreślić, iż „integracja projektu” rozumiana jest w tym przypadku jako element zarządzania projektem, a nie integracji technicznej fragmentów oprogramowania.

W ramach zarządzania integracją projektu zalecenia SWX pozostają zgodne z rekomendacjami PMBoK Guide opartymi na sześciu procesach zarządzania:

- 4.1. Opracowanie karty projektu
- 4.2. Opracowanie planu zarządzania projektem
- 4.3. Kierowanie i zarządzanie pracami w projekcie
- 4.4. Monitorowanie i kontrolowanie pracy w projekcie
- 4.5. Przeprowadzanie zintegrowanej kontroli zmian
- 4.6. Zamykanie projektu lub etapu

W zakresie procesu 4.1 zalecenia pozostają bez zmian.

W zakresie procesu 4.2 (opracowanie planu zarządzania projektem) SWX podkreśla fakt, iż przebieg projektu informatycznego rzadko jest liniowy, a częściej opiera się na ewolucji oraz odkrywaniu i uczeniu się w trakcie projektu. W efekcie planowanie wymaga nie tylko wysiłku na początku projektu (ang. *upfront planning*), ale także intensywnej aktualizacji i replanowania w trakcie jego przebiegu. Specyficzny zakres i szczegółowość planowania determinowane będą głównie przez wybór konkretnego modelu cyklu życia projektu. W przypadku podejścia planistycznego wysiłki na wstępie będzie znacznie większy niż w przypadku podejść adaptacyjnych, które z kolei będą wymagały większego nacisku na planowanie podczas realizacji projektu.

Aby oddać specyfikę projektów informatycznych, plan (będący głównym rezultatem tegoż procesu) nazwany został „planem zarządzania projektem oprogramowania” (w oryginale *software project management plan*). Oprócz dotychczasowych elementów planu zarządzania projektem autorzy SWX dodali do niego m.in.¹⁹⁰:

- 1) plan zarządzania wymaganiami,
- 2) plan zarządzania konfiguracją,

¹⁹⁰ PMI, *Software Extension...*, s. 51.

- 3) plany bezpieczeństwa (fizycznego, projektu, danych),
- 4) plan zarządzania jakością,
- 5) plan zarządzania integracją technologiczną przedsiębiorstwa (ang. *enterprise technology insertion plan*),
- 6) plan bezpieczeństwa informacji,
- 7) plan testów i ewaluacji,
- 8) plan zarządzania informacją,
- 9) plan zarządzania wydaniem i wdrożeniem,
- 10) plan infrastruktury technologicznej,
- 11) plan szkolenia zespołu projektowego.

W obrębie procesu 4.3 (kierowanie i zarządzanie pracami projektu) autorzy rozszerzenia podkreślają specyfikę kultury pracy zespołów adaptacyjnych polegającą na samokontroli, samoorganizacji i wysokiej integracji między ich członkami. W takich okolicznościach kierownik projektu realizowanego według adaptacyjnych cykli życia może zmniejszyć swoje zaangażowanie w codzienne kierowanie swoimi ludźmi. Nie oznacza to oczywiście wyłączenia go z pracy. Nadal do jego obowiązków należeć będzie m.in.: komunikowanie ograniczeń zakresu, zasobów i czasu projektu zespołowi projektowemu oraz innym interesariuszom, zapewnianie niezbędnych zasobów, tworzenie mechanizmów kontroli całości projektu, nadzorowanie zmian, umożliwienie współpracy między wieloma podzespołami projektowymi, zapewnienie jej z innymi jednostkami organizacji i wiele innych¹⁹¹.

Wśród listy narzędzi i technik pomocnych w wykonaniu omawianego procesu PMI dodał rozpowszechnianie informacji (ang. *information dissemination*). Działanie to ma być w uznaniu autorów szczególnie istotne w zakresie projektów informatycznych ze względu na ich niematerialny, trudno uchwytne charakter. Dlatego też kierownik projektu powinien dołożyć wszelkich starań aby kluczowi interesariusze byli dobrze poinformowani o stanie projektu, jego aktualnych i prognozowanych postępach, informacjach o efektach testowania itp. Jako narzędzie wspierające komunikację SWX rekomenduje tablice i ekrany zarządcze (ang. *information radiators*) umieszczane w widocznych, uczęszczanych miejscach.

Wśród rezultatów procesu 4.3 w SWX dodatkowo umieszczono prezentacje działających elementów oprogramowania (ang. *demonstrations of working, deliverable software*).

W zakresie procesu 4.4 (monitorowanie i kontrolowanie pracy w projekcie) autorzy SWX zwracają uwagę na dodatkową możliwość kontroli pracy poprzez ocenę dostarczanych przyrostów opracowywanego oprogramowania. Pozostałe zalecenia dotyczące procesu pozostają zgodne z PMBoK Guide.

¹⁹¹ PMI, *Software Extension...*, s. 52.

Jeśli idzie o proces 4.5 (przeprowadzanie zintegrowanej kontroli zmian) autorzy podkreślają zróżnicowanie sposobu obsługi zmian w zależności od przyjętego modelu cyklu życia projektu. W przypadku modeli predyktywnych ograniczenia (tolerancje) parametrów projektu ustanawiane są zwykle podczas inicjowania i planowania, a ich przekroczenie uruchamia proces kontroli zmian wraz z żądaniem zmiany oraz (często) komitetem ds. zmian. W przypadku cykli adaptacyjnych dla sporadycznych odchyłeń podejście formalne najczęściej nie jest wymagane, o ile projekt zachowuje zgodność z wizją i zasadniczymi ograniczeniami. Żądania zmian dotyczące zakresu obsługiwane są na poziomie listy funkcjonalności produktu oraz zakresu funkcjonalności wchodzących do poszczególnych iteracji. Pozostałe zalecenia dotyczące procesu pozostają zgodne z PMBoK Guide.

Ostatni proces w ramach zarządzania integracją to 4.6 (zamykanie projektu lub etapu). Autorzy SWX zachowują w jego przypadku zalecenia PMBoK Guide, dodając informację, iż proces ten może mieć miejsce nie tylko, jak dotychczas, przy końcu projektu lub jego fazy, ale także przy końcu poszczególnych iteracji. Pozostałe zalecenia dotyczące procesu pozostają zgodne z PMBoK Guide.

Zarządzanie zakresem projektu

Obszar zarządzania zakresem projektu ma na celu wyznaczenie granic projektu i zakresu prac koniecznych do wykonania. Zarządzanie zakresem w projekcie dostarcza zespołowi procesów, których zadaniem jest zapewnienie, że wszystkie konieczne (i tylko takie) prace zostaną wykonane. W projekcie informatycznym szczególnie istotne jest opisanie zakresu produktu, czyli zestawu funkcjonalności i cech jakościowych oczekiwanego rezultatu (oprogramowania) oraz zakresu projektu, czyli prac koniecznych do dostarczenia produktu. W zależności od okoliczności kolejność określania obu zakresów może być różna.

W ramach zarządzania zakresem projektu zalecenia SWX pozostają zgodne z rekomendacjami PMBoK Guide opartymi na sześciu procesach zarządzania:

- 5.1. Planowanie zarządzania zakresem
- 5.2. Zbieranie wymagań
- 5.3. Precyzowanie zakresu
- 5.4. Tworzenie struktury podziału pracy
- 5.5. Przeprowadzanie walidacji zakresu
- 5.6. Kontrolowanie zakresu

Zalecenia SWX dotyczące procesu 5.1 (planowanie zarządzania zakresem) koncentrują się na rozróżnieniu pomiędzy planowaniem zakresu w cyklach predyktywnych i adaptacyjnych. W tych pierwszych działania związane z planowaniem i definiowaniem zakresu realizowane są we wczesnych fazach projektu, dostarczając dokumentacji wyma-

gań oraz architektury oprogramowania niezbędnych do opracowania struktury podziału pracy (WBS). Podejście predyktywne zalecane jest w przypadku projektu o stabilnych, dających się szczegółowo opisać wymaganiach, realizowanego w poznanej technologii. W przypadku projektów innowacyjnych bardziej sprawdzać się będzie podejście adaptacyjne, gdzie zakres projektu i produktu wspólnie ewoluują poprzez kolejne iteracje rozwoju oprogramowania. Autorzy zadbali też o dodatkowe zasilenie do procesu – planowanie wydań dla planowania zarządzania zakresem (ang. *release planning for planning scope management*). Założenia odnośnie do wydań produktu czynione są najczęściej podczas inicjowania i planowania projektu. Ich liczba, daty oraz zakres funkcjonalności do dostarczenia są kluczowymi informacjami koniecznymi do poprawnego planowania zarządzania zakresem projektu informatycznego. Pozostałe zalecenia dotyczące procesu pozostają zgodne z PMBoK Guide.

Podobnie jak w przypadku pozostałych procesów 5.2 (zbieranie wymagań SWX) różnicuje sposób wykonania procesu w zależności od wybranego modelu cyklu życia projektu. W przypadku modeli predyktywnych wymagania gromadzone są podczas faz inicjowania i planowania projektów, dąży się zaś do opracowania ich pełnego, kompletnego i szczegółowego kształtu zgodnie z aktualną wiedzą. Jako że ich ostateczny zestaw staje się podstawą do planowania pozostałych obszarów projektów, to ewentualne zmiany są w takich przypadkach obsługiwane poprzez proces zarządzania zmianami. W modelach adaptacyjnych opracowuje się wstępne wymagania konieczne do rozpoczęcia pracy, następnie zaś rozwija się je w sposób ewolucyjny w trakcie trwania projektu. Wymagania odnośnie do produktu przechowywane są na odrębnej liście i podlegają rozpatrzeniu przy planowaniu kolejnych iteracji.

Precyzowanie (definiowanie) zakresu ma na celu precyzyjne opisanie projektu i produktu. W zakresie projektów informatycznych oznacza to najczęściej analizę wymagań i funkcjonalności produktu pod kątem priorytetów i z perspektywy oczekiwań interesariuszy. Należy jednak pamiętać, że wstępny zakres projektu będzie ulegał zmianom w miarę ewolucyjnego postępu projektu. Znajdzie to swoje odzwierciedlenie w nowym rezultacie procesu 5.3, czyli dodatkowych uwagach (ang. *additional considerations*), które mają umożliwić kierownikowi projektu zrealizowanie systematycznej ewolucji zakresu projektu i produktu.

W zakresie tworzenia struktury podziału pracy autorzy SWX – ze względu na niematerialny charakter projektów informatycznych – sugerują stosowanie techniki struktury podziału prac zorientowanej na czynności (ang. *activity-oriented work breakdown structure*). Ewolucyjne podejście do tworzenia zakresu projektu zostało także uwzględnione w nowych technikach krocącego tworzenia WBS-u (ang. *rolling wave elaboration of WBS*) oraz planowania krocącego dla adaptacyjnych cykli życia projektu (ang. *rolling wave planning for adaptive life cycle projects*).

Walidacja zakresu jest procesem, którego celem jest uzyskanie od interesariuszy potwierdzenia wykonania i odbioru części dostarczanego zakresu projektu. Zalecenia dotyczące procesu pozostają zgodne z PMBoK Guide. Dla potrzeb projektów realizowanych adaptacyjnie podkreślono, iż walidacja zakresu dotyczy przyrostów oprogramowania uzyskiwanych na koniec każdego cyklu iteracyjnego i może, ale nie musi, być opisana w formalnym planie walidacji. Wśród rezultatów procesu walidacji wyszczególnione zostały zaakceptowane produkty (ang. *accepted deliverables*), które w takim przypadku przybierają postać w pełni funkcjonującego przyrostu produktu.

Kontrola zakresu jest procesem, który ma umożliwić monitorowanie projektu pod kątem wykonywanych prac oraz wprowadzanych w tym obszarze zmian. Autorzy SWX podkreślają, iż oryginalne zalecenia PMBoK Guide mają zastosowanie w przypadku projektów w modelach predyktywnych, które oparte są najczęściej na tradycyjnych technikach zarządzania. W przypadku modeli adaptacyjnych kontrola zakresu polega na współpracy pomiędzy kierownikiem projektu, zespołem oraz przedstawicielami klienta przy ustalaniu zestawów funkcjonalności włączanych w zakres poszczególnych iteracji i przyrostów. Kontrolowanie zakresu ma na celu zapewnienie satysfakcji klienta i dostarczenie mu najbardziej wartościowego kształtu finalnego produktu. W miejsce formalnych decyzji odnośnie do akceptacji bądź odrzucenia zmian podstawowym rezultatem kontroli zakresu są ustalenia dotyczące listy funkcjonalności najbliższej iteracji.

Zarządzanie czasem projektu

Obszar zarządzania czasem projektu dostarcza kierownikowi projektu zaleceń metodycznych pozwalających na jego realizację terminowo i według dostępnego czasu. Ze względu na fakt, iż koszt projektów informatycznych determinowany jest głównie czasem pracy zasobów, jest to obszar znaczący dla powodzenia realizacji projektu. Zalecenia SWX dotyczące zarządzania czasem projektu informatycznego pozostają zgodne z zaleceniami PMBoK Guide, wzbogacając jednocześnie poszczególne procesy o dodatkowe narzędzia i techniki. W ramach obszaru wiedzy wyszczególniono siedem procesów:

- 6.1. Planowanie zarządzania harmonogramem
- 6.2. Określanie działań
- 6.3. Określanie kolejności działań
- 6.4. Szacowanie zasobów działań
- 6.5. Szacowanie czasu trwania działań
- 6.6. Opracowanie harmonogramu
- 6.7. Kontrolowanie harmonogramu

W zakresie planowania zarządzania harmonogramem autorzy SWX zwrócili uwagę na problem zagadnień bezpieczeństwa i ochrony w tworzeniu oprogramowania. Element

ten został dodany jako zasilenie tegoż procesu. Pozostałe zalecenia dotyczące procesu pozostają zgodne z PMBoK Guide.

W zakresie definiowania czynności rozszerzenie dostarcza kierownikom projektów trzech nowych technik i narzędzi. Są to: struktury podziału historii użytkownika (ang. *story breakdown structures*), makiety (ang. *storyboards*) oraz przypadki użycia (ang. *use cases*). Zastosowanie tych technik ułatwia określenie szczegółowych zadań służących wytworzeniu przyjętych funkcjonalności produktów.

W zakresie określenia kolejności zadań SWX zwraca uwagę na wpływ architektury systemowej przedsiębiorstwa i innych ograniczeń architektury na optymalny układ działań w projekcie. Elementy te, wraz z analizami bezpieczeństwa i ochrony, dodane zostały jako zasilenia omawianego procesu. Dodano także narzędzia i techniki takie jak: model SAIV (ang. *schedule as independent variable*) oraz *time boxing* (rozwiązania mające na celu dostarczenie najbardziej wartościowych funkcjonalności w z góry określonym czasie), oceny zestawów funkcjonalności (ang. *feature set evaluation*) oraz umowy SLA (ang. *service level agreement*). Oprócz istniejących już wcześniej wykresów sieciowych oraz aktualizacji dokumentacji projektowej do grupy rezultatów procesu dodano: zestawy funkcjonalności (ang. *features set*), plany wydań (ang. *release plans*) oraz zależności architektoniczne oraz niefunkcjonalne (ang. *architectural and nonfunctional dependences*).

Podczas szacowania zasobów działań w sposób naturalny kierownik projektu koncentrować się będzie na wydajności pracy (ang. *velocity*) dostępnych mu programistów. Pomocne w tym zakresie będą dane historyczne dotyczące jakości i wyników pracy. Sam zestaw zasileń i rezultatów tego procesu pozostał bez zmian, zaś w ramach narzędzi i technik dodano umowy SLA oraz „inne narzędzia i techniki”, tzn. techniki szacowania czasu, mogące zostać wykorzystane także przy szacowaniu zasobów działań.

W ramach szacowania czasu trwania działań autorzy podkreślają jego trudność ze względu na niematerialny charakter rezultatów oraz nieliniowy związek czasu i stopnia złożoności czy ilości pracy do wykonania. Z perspektywy zaleceń SWX zarówno zasilenia, jak i narzędzia/techniki oraz rezultaty pozostają bez zmian w stosunku do oryginalnego PMBoK Guide. Autorzy dodali też w ostatnim punkcie tzw. inne zasilenia (ang. *additional inputs*), wymieniając wśród nich: historie użytkownika, listy, grupy, zestawy funkcjonalności oraz informacje o wydajności zespołu.

W zakresie opracowania harmonogramu zwrócono uwagę na fakt, iż plan przebiegu projektu może przybierać w projektach informatycznych inną postać niż typowe harmonogramy. Może być to np. spriorytetyzowana lista zadań (ang. *backlog of work activities*). Z perspektywy zaleceń SWX zasadniczo elementy procesu pozostają bez zmian w stosunku do oryginalnego PMBoK Guide. W ramach zasileń autorzy dodali w ostatnim punkcie tzw. inne zasilenia (ang. *additional inputs*) wymieniając wśród nich: listy czynności, funkcjonalności i ich zestawy, historie użytkowników, dane historyczne o wydajności pracy

zespołu oraz umowy SLA. Narzędzia i techniki uzupełniono o dwa elementy. Pierwszy to kompresja harmonogramu (ang. *schedule compression*), czyli skrócenie czasu poprzez zmniejszenie zakresu funkcjonalności do dającego się uzyskać przy istniejących warunkach i ograniczeniach pracy, zaś drugi to przyrostowe planowanie produktu (ang. *incremental product planning*) Wśród rezultatów procesu dodano aktualizację planu wydań i iteracji (ang. *release and iteration plan updates*).

Kontrolowanie harmonogramu w przypadku projektów informatycznych polegać będzie, zgodnie z zaleceniami SWX, przede wszystkim na kontrolowaniu wydajności pracy zespołów w trakcie iteracji, jej ocenie podczas spotkań retrospektywnych, repriorytetyzacji listy działań do pozostałych do wykonania oraz współpracy w tym zakresie z zamawiającym. Lista rekomendowanych zasileń procesu pozostała bez zmian. W zakresie narzędzi i technik uwzględniono: kompresję harmonogramu, przeglądy oparte na dowodach (czyli przeglądy na bazie działającego fragmentu oprogramowania), retrospektywy, diagramy skumulowanego przebiegu (ang. *cumulative flow diagrams*), tablice przepływu pracy wraz z codziennymi spotkaniami aktualizującymi, przeglądy repriorytetyzujące, wykresy rozpalania i spalania (ang. *burnup and burndown charts*) oraz analiza odchyleń. Wśród rezultatów procesu dodano tzw. dodatkowe rezultaty (ang. *additional outputs*), wyszczególniając wśród nich: informacje o wydajności pracy zespołu, aktualizacje planów iteracji i wydań oraz modyfikacje umów SLA.

Zarządzanie kosztem projektu

Obszar zarządzania kosztem projektu dostarcza kierownikowi projektu metod zarządzania pozwalających zachować kontrolę nad jego kosztami i budżetem. Nawet w przypadku projektów realizowanych adaptacyjnie kierownik projektu oraz interesariusze powinni być świadomi związanych z nimi kosztów (głównie kosztów pracy zespołu), ich dotychczasowych wartości oraz prognoz. Elastyczność dotyczącą zakresu produktu i projektu będzie najczęściej ograniczana dostępnymi zasobami, w tym możliwymi nakładami finansowymi. Muszą one pozostać pod kontrolą kierownictwa projektu i organizacji. W zakres działań w tym obszarze wchodzić cztery procesy, tj.:

7.1. Planowanie zarządzania kosztami

7.2. Szacowanie kosztów

7.3. Określanie budżetu

7.4. Kontrolowanie kosztów

Odnosnie do pierwszego procesu autorzy SWX podkreślają konieczność określenia odpowiednich mechanizmów finansowania projektów w przypadku przyjęcia przyrostowych modeli rozwoju oprogramowania oraz konieczność ustanowienia właściwego podejścia do kontroli zmian w zakresie odchyleń od bazowych wartości budżetu. Jeśli

idzie o zasilenia procesu, zmodyfikowano treść „aktywów procesów organizacyjnych”, uwzględniając trzy dodatkowe elementy, tj.:

- 1) czynniki kosztów (ang. *cost drivers*); konieczność uwzględnienia parametrów tworzonego oprogramowania, takich jak m.in.: rozmiar, złożoność, konieczne umiejętności programistów, koszty infrastruktury, koszty zarządzania konfiguracją czy testowania,
- 2) zasady ładu organizacyjnego w zakresie obszaru IT (m.in. COBIT, COSO, ITIL ISO/IEC 20000 i inne) oraz
- 3) wytyczne wynikające z zarządzania portfelowego w organizacji (np. kosztów utrzymania infrastruktury, podejścia do otwartego oprogramowania czy praw autorskich).

Lista technik pomocnych przy planowaniu zarządzania kosztami projektów informatycznych pozostała niezmieniona, natomiast w zakresie rezultatów dodano trzy elementy: poziom dokładności szacunków (ang. *accuracy of estimate*), jednostki miary kosztów (ang. *units of measure*) oraz metody pomiaru wydajności kosztów (ang. *cost performance measurement method*).

W zakresie szacowania kosztów projektów informatycznych autorzy SWX, podkreślając trudność planowania, zwracają uwagę na wartość płynącą z posiadania wielu początkowych szacunków, na bazie których można opracować wiarygodne wartości. Wstępne szacunki mają umożliwić określenie bardziej rzędów wielkości niż dokładnych wartości, tak aby można było sprawnie przejść do wstępnego planowania. Twórcy rozszerzenia dodatkowo wyszczególniają czynniki wpływające na wysokość kosztów, na które szczególnie powinien zwrócić uwagę kierownik takiego projektu. Są to: czynniki bezpośrednio związane z projektem (wielkość zespołu, rozmiar i złożoność rozwiązania itp.), wymagania prawne, zgodność ze standardami, zmiany organizacyjne, ryzyko projektu, koszty finansowania projektu.

Wprowadzone zmiany w zakresie zasileń procesu szacowania kosztów dotyczą: planu bazowego zakresu (ang. *scope baseline*), harmonogramu projektu, rejestru ryzyk oraz czynników środowiska organizacji. Dodatkowo kierownicy projektów powinni uwzględnić w szacowaniu kosztów wspomniane wcześniej wielkość i złożoność oprogramowania oraz tempo pracy zespołu. W zakresie technik szacowania kosztów autorzy SWX opierają się na tych już istniejących w PMBoK Guide, jak również dodają nowe, specyficzne dla projektów informatycznych: szacowanie w ramach iteracji (ang. *time-boxed estimating*), szacowanie oparte na punktach funkcjonalnych i liniach kodu źródłowego, szacowanie oparte na historiach użytkownika i przypadkach użycia, szacowanie wysiłku związanego z tworzeniem kodu wielokrotnego użytku oraz szacowanie ceny do uzyskania (ang. *price-to-win*). Zalecenia dotyczące rezultatów procesu oraz rozszerzenia dotyczące procesu określania budżetu projektu pozostają bez zmian w stosunku do oryginalnych zaleceń PMBoK Guide.

Ze względu na adaptacyjny charakter projektów informatycznych, podobnie jak w przypadku kontroli zakresu, kontrola kosztów projektu będzie musiała oddawać ewolucyjny charakter projektu oraz dynamikę jego zmian. Najczęściej koszty bezpośrednie liczone będą w zależności od poziomu wysiłku włożonego przez zespół przy dostarczaniu uzgodnionych w ramach iteracji zestawów funkcjonalności. Lista zasileń i rezultatów procesu pozostała bez zmian, dopasowano zaś opis trzech z zalecanych technik:

- 1) technika wartości uzyskanej – uznanie użyteczności metody w przypadku projektów informatycznych przy założeniu, iż postęp projektu oddawany będzie przez kolejne przyrosty końcowego projektu oraz przy ostrzeżeniu co do możliwych błędów pomiaru wynikających z jego niematerialnego charakteru,
- 2) prognozowanie oraz mierniki zarządcze – odniesienie do metod takich jak EV, wykresy spalania, wykresy skumulowanego przepływu; nacisk na łatwość interpretacji i czytelność przekazu.

Zarządzanie jakością projektu

Zarządzanie jakością projektu jest obszarem wiedzy, który rekomenduje kierownikowi projektu zestaw działań mających na celu zapewnienie, że projekt spełni wymagania, dla których został powołany.

W ramach obszaru zalecane są trzy procesy:

- 8.1. Planowanie zarządzania jakością
- 8.2. Przeprowadzanie zapewniania jakości
- 8.3. Kontrolowanie jakości

Przeprowadzenia zapewnienia jakości projektu informatycznego (ang. *software quality assurance*) ma charakter ciągłego procesu zapewniającego przestrzeganie przyjętych standardów względem procesu tworzenia oprogramowania. Kontrola jakości projektu informatycznego (ang. *software quality control*) ma na celu zastosowanie odpowiednich metod sprawdzających, czy opracowywany produkt spełnia określone kryteria jakości i wymagania. Zapewnienie jakości wykonywane na poziomie zespołu najczęściej przyjmuje formę ocen *ex post*, spotkań retrospektywnych, przeglądów doświadczeń projektowych oraz ocen mierników wydajności. W zakresie kontroli jakości są to zwykle przeglądy, inspekcje oraz testowanie.

W przypadku projektów realizowanych zgodnie z modelami adaptacyjnymi częste iteracje oraz demonstracje fragmentów oprogramowania wymagają zintegrowanego podejścia do zarządzania jakością. Wynika to z kluczowej zasady, iż demonstrowane przyrosty powinny być wykonane w całości, przetestowane i pozbawione błędów.

Pod względem zasileń, narzędzi i technik oraz rezultatów przyporządkowanych do poszczególnych procesów wszystkie dotychczasowe zalecenia – po uwzględnieniu specy-

fiki projektów informatycznych – zostały utrzymane w mocy. Uzupełnienie uwzględnia dodatkowy rezultat w procesie kontroli jakości, tzw. dodatkowe rezultaty, czyli:

- 1) mierniki cech jakościowych zawartych w planie zarządzania jakością oraz kryteriach wydań,
- 2) zmiany w oprogramowaniu zatwierdzone podczas testów,
- 3) produkty prac zwalidowane po testach odniesione do pierwotnego zakresu projektu,
- 4) identyfikację odchyłeń między planowanymi i rzeczywistymi wynikami wraz z ich wytłumaczeniem,
- 5) zaktualizowane listy kontrolne, procedury testowe i inne aktywa procesów,
- 6) doświadczenia projektowe uzyskane podczas retrospektyw,
- 7) aktualizacje planów zarządzania projektem¹⁹².

Zarządzanie zasobami ludzkimi projektu

Obszar zarządzania zasobami ludzkimi projektu dostarcza kierownikowi projektu narzędzi planowania, organizowania, zarządzania i przewodzenia zespołowi projektowemu. W ramach obszaru wiedzy znajdują się cztery procesy:

- 9.1. Planowanie zarządzania zasobami ludzkimi
- 9.2. Pozyskiwanie zespołu projektowego
- 9.3. Rozwijanie zespołu projektowego
- 9.4. Zarządzanie zespołem projektowym

Specyficzne zalecenia SWX dotyczące tego obszaru koncentrują się na szczególnych cechach takich zespołów, przypisywanych im zarówno przez praktyków, jak i teoretyków adaptacyjnego zarządzania projektami. W szczególności dotyczy to uznania członków zespołów za kompetentnych specjalistów w swoich dziedzinach, bliskich modelowi „pracownika wiedzy” i teorii Y. McGregora. W konsekwencji kierownik projektu powinien mniej mikrozarządzać, a więcej uwagi poświęcić na komunikację i facylitację pracy małych, płaskich, samoorganizujących i samodyscyplinujących się zespołów.

Pod względem zasileń, narzędzi i technik oraz rezultatów przyporządkowanych do poszczególnych procesów wszystkie dotychczasowe zalecenia, po uwzględnieniu specyfiki projektów informatycznych, zostały utrzymane w mocy. Uzupełnienie uwzględnia nowe techniki w procesach rozwijania zespołu projektu (tzw. dodatkowe narzędzia i techniki) oraz w procesie zarządzania zespołem projektowy (tzw. dodatkowe uwagi). W pierwszym przypadku są to techniki takie jak: programowanie w parach, programowanie nastawione na testy (ang. *test-driven development*), kolokacja zespołów oraz budowanie zaufania. W drugim autorzy podkreślili delikatny charakter pomiaru wydajności

¹⁹² PMI, *Software Extension...*, s. 160.

członków zespołów programistycznych oraz potrzebę relacji w tym zakresie „jeden do jednego” z kierownikiem projektu, a także możliwość eksperymentowania z przypisywaniem ról poszczególnym członkom zespołu pomiędzy kolejnymi iteracjami. Daje to szansę jego lepszego poznania oraz znalezienia optymalnej konfiguracji. Iteracje oraz retrospektywy umożliwiają dodatkowo uzyskanie informacji zwrotnej z pracy zespołu. Ponadto autorzy zwracają uwagę na wpływ niematerialnego charakteru oprogramowania na ryzyko pojawiania się konfliktów.

Zarządzanie komunikacją projektu

Obszar zarządzania komunikacją jest obszarem który pozwala kierownikowi projektu na terminowe, precyzyjne tworzenie, zbieranie, rozpowszechnianie, przechowywanie i ostateczne usuwanie informacji związanych z projektem.

10.1. Planowanie zarządzania komunikacją

10.2. Zarządzanie komunikacją

10.3. Kontrolowanie komunikacji

Jako że projektom informatycznym, szczególnie tym realizowanym w podejściach adaptacyjnych, towarzyszą dynamizm i szybkie tempo zmian, to powinny iść za tym odpowiednie rozwiązania w zakresie komunikacji. Będą to przede wszystkim środki komunikacji bezpośredniej („twarzą w twarz”): spotkania planistyczne, codzienne spotkania „na stojąco” (ang. *daily stand-up meetings*), częste demonstrowanie postępów i przyrostów pracy. W przypadku dużych projektów w celu ograniczenia liczby kanałów komunikacji zaleca się stosowanie większej liczby mniejszych, koordynowanych zespołów niż jednego, dużego.

Pod względem zasileń, narzędzi i technik oraz rezultatów przyporządkowanych do poszczególnych procesów wszystkie dotychczasowe zalecenia, po uwzględnieniu specyfiki projektów informatycznych zostały utrzymane w mocy. Uzupełniono je o dodatkowe rozwiązania dotyczące procesu zarządzania komunikacją oraz procesu kontrolowania komunikacji.

W ramach zarządzania komunikacją dodano:

- 1) nowe zasilenie w postaci planów wydań oraz iteracji,
- 2) nowe narzędzia i techniki: tablice i ekrany zarządcze (ang. *information radiators*), informacje o aktualnej wydajności zespołów (ang. *velocity*), informacje o historycznej wydajności zespołów (ang. *historical velocity*) oraz narzędzia współpracy online,
- 3) nowe rezultaty procesu: specjalistyczne narzędzia komunikacji, narzędzia współpracy online oraz zaktualizowane tablice i ekrany zarządcze.

W ramach kontrolowania komunikacji dodano:

- 1) nowe zasilenia: spriorytetyzowaną listę funkcjonalności (ang. *prioritized backlog*) oraz statystyki i prognozy wydajności zespołów,

- 2) nowe narzędzia i techniki: taktowną komunikację (tj. komunikację, która nie zakłóca pracy w skolokowanych zespołach projektowych) oraz zautomatyzowane systemy wspierające komunikację (np. wiki, strony internetowe projektu i inne),
- 3) nowe rezultaty procesu: aktualizacje planu iteracji i wydań oraz zrepriorytetyzowaną listę funkcjonalności.

Zarządzanie ryzykiem projektu

Obszar zarządzanie ryzykiem dostarcza kierownikowi projektu rozwiązań umożliwiających ograniczenie prawdopodobieństwa i skutków negatywnych zdarzeń (zagrożeń) w projekcie oraz zwiększenie tychże parametrów dla zdarzeń pozytywnych (szans). Z perspektywy projektów informatycznych kluczowe obszary ryzyka, którymi powinno się zarządzać dotyczą ryzyk: technicznych, związanych z harmonogramem, kosztami, jakością, dynamiką zespołów oraz współpracą z interesariuszami. Oryginalnie PMBoK Guide rekomenduje stosowanie w tym celu sześciu procesów:

- 11.1. Planowanie zarządzania ryzykiem
- 11.2. Rozpoznawanie ryzyk
- 11.3. Przeprowadzenie jakościowej analizy ryzyk
- 11.4. Przeprowadzanie ilościowej analizy ryzyk
- 11.5. Planowanie reakcji na ryzyka
- 11.6. Kontrolowanie ryzyk

Pod względem zasileń, narzędzi i technik oraz rezultatów przyporządkowanych do poszczególnych procesów wszystkie dotychczasowe zalecenia, po uwzględnieniu specyfiki projektów informatycznych, zostały utrzymane w mocy. Uzupełniono je o dodatkowe rozwiązania dotyczące procesów planowania zarządzania ryzykiem, rozpoznawania ryzyk, przeprowadzania jakościowej analizy ryzyk oraz planowania reakcji na ryzyka.

W ramach pierwszego procesu narzędzia i techniki zostały uzupełnione o dodatkowe uwagi (ang. *additional considerations*), w których autorzy SWX zwracają uwagę na znaczną redukcję całościowego poziomu ryzyka w skutek stosowania adaptacyjnych modeli rozwoju oprogramowania. Dzięki częstej repriorytetyzacji funkcjonalności oraz przyrostowemu budowaniu produktu zespół projektowy ma szansę uniknąć wielu zagrożeń i wykorzystać pojawiające się szanse. Może także ocenić funkcjonalności pod kątem towarzyszących im ryzyk oraz przystąpić do ich wytworzenia odpowiednio wcześniej.

W odniesieniu do drugiego procesu autorzy SWX uzupełnili zasilenia procesu o specyficzne taksonomie ryzyk dotyczące projektów informatycznych np. struktura podziału ryzyk (RBS) opracowana przez Software Engineering Institute. Lista narzędzi i technik wzbogacona została także o spotkania retrospektywne, podczas których możliwe jest identyfikowanie nowych, niedostrzeżonych dotychczas zagrożeń.

W zakresie procesu przeprowadzania jakościowej analizy ryzyk narzędzia i techniki zostały także uzupełnione o dodatkowe uwagi (ang. *additional considerations*). W uwagach tych autorzy podkreślają subiektywność oceny ryzyk w projektach informatycznych, nieczęsty dostęp: do wiarygodnych statystyk pomocnych w ocenie zagrożeń oraz możliwy wpływ ostatnich projektów na zaburzenie oceny ryzyk w bieżącym przedsięwzięciu. W przypadku projektów realizowanych zgodnie z modelami adaptacyjnymi oceny tego typu mogą być jednym z parametrów wspierającym podejmowanie decyzji odnośnie do prac włączanych w zakres kolejnych iteracji.

Podobnie jak we wcześniejszych procesach planowanie odpowiedzi na ryzyka zostało uzupełnione o dodatkowe uwagi w części dotyczącej nie tylko narzędzi i technik, ale i rezultatów. Odnośnie do technik zalecenia te dotyczą uwzględnienia poziomu ryzyka elementów oprogramowania w procedurach testowych (ang. *risk-based testing* – czyli intensywniejsze testowanie najbardziej narażonych na błędy funkcjonalności) oraz stosowania tzw. czynników dźwigni ryzyka (ang. *risk leverage factors*, RLF), czyli matematycznej formuły oceniającej koszt zmniejszenia ryzyk z poziomu inherentnego do rezydualnego. Uwagi w ramach rezultatów dotyczą zalecenia, aby szczegółowo planować reakcje na zagrożenia, uwzględniając zatwierdzone aktywności, sposób monitorowania ryzyk, ich wskaźniki (ang. *triggers*), właścicieli ryzyk oraz właścicieli reakcji na ryzyka, a także koszty i czas potrzebny na realizację działań. Autorzy w odrębnej tabeli przedstawili proponowane działania polegające na unikaniu, transferze i mitygacji wybranych grup ryzyk (np. technicznych, bezpieczeństwa czy zespołu).

Zarządzanie zamówieniami projektu

Przedostatni obszar wiedzy o zarządzaniu projektami, tj. zarządzanie zamówieniami projektu, ustanowiony został w celu umożliwienia kierownikowi projektu zorganizowania zaopatrzenia projektu w dobra i usługi spoza organizacji realizującej projekt. W ramach zarządzania zamówieniami projektu stosowane są cztery procesy:

- 12.1. Planowanie zarządzania zamówieniami
- 12.2. Przeprowadzanie zamówień
- 12.3. Kontrolowanie zamówień
- 12.4. Zamykanie zamówień

Pod względem zasileń, narzędzi i technik oraz rezultatów przyporządkowanych do poszczególnych procesów wszystkie dotychczasowe zalecenia, po uwzględnieniu specyfiki projektów informatycznych, zostały utrzymane w mocy. Autorzy SWX nie dodali żadnych nowych elementów.

Obszar ten, w swoim podstawowym rozumieniu traktuje organizację realizującą projekt jako zamawiającego (przypadki zakupu gotowego lub otwartego oprogramowania).

Zalecenia, opisane np. w planowaniu zarządzania zamówieniami, dotyczą sposobów identyfikacji dostawców, przygotowania deklaracji celów (ang. *statement of objective / statement of work*), określenia kryteriów oceny ofert oraz przygotowania umowy i regulaminu dostaw. Autorzy zwracają także uwagę na zagadnienie licencji towarzyszących oprogramowaniu, których zapisy mogą regulować zakres stosowania i modyfikacji, istotne ograniczenia w dostępie czy transferze, jak również związane z tym opłaty. Realizacja projektu na potrzeby zewnętrzne może wymagać zmiany podejścia do proponowanych procesów na perspektywę dostawcy produktu. Innymi istotnymi zagadnieniami zarządzania zamówieniami mogą być w tym zakresie prawa autorskie i licencje związane z kupowanym, budowanym lub modyfikowanym oprogramowaniem (np. warunki dostępu do kodu źródłowego) oraz sposób wykorzystania zewnętrznych zasobów do celów projektu (*outsourcing*).

Zarządzanie interesariuszami projektu

Ostatnim dziesiątym obszarem wiedzy o zarządzaniu projektami zawartym w PMBoK Guide oraz SWX jest zarządzanie interesariuszami projektu. Znaczenie interesariuszy w projektach jest znaczne, zaś w projektach informatycznych, ze względu na niematerialny charakter produktu, jeszcze większe – a nawet decydujące. Zgodnie z zaleceniami PMBoK Guide kierownik powinien uważnie zarządzać satysfakcją kluczowych interesariuszy, zaś obszar ten dostarcza mu czterech procesów dzięki którym jest to możliwe. Są to:

- 13.1. Rozpoznawanie interesariuszy
- 13.2. Planowanie zarządzania interesariuszami
- 13.3. Zarządzanie angażowaniem interesariuszy
- 13.4. Kontrolowanie zaangażowania interesariuszy

Według oceny autorów SWX w projektach informatycznych angażowanie interesariuszy odbywa się najczęściej podczas planowania i określania wymagań projektu oraz w okresowych przeglądach przy „kamieniach milowych”. Zaangażowanie to, nawet w modelach predyktywnych, może być wzmocnione, jeśli zastosowane zostanie podejście inkrementalne z regularnymi demonstracjami przyrostów. W przypadku projektów adaptacyjnych współpraca ta musi być ścisła i częsta, gdyż w inny sposób zespołowi brakować będzie informacji zwrotniej koniecznej do ewolucji projektu zgodnie z oczekiwaniami klienta.

Pod względem zasileń, narzędzi i technik oraz rezultatów przyporządkowanych do poszczególnych procesów wszystkie dotychczasowe zalecenia, po uwzględnieniu specyfiki projektów informatycznych, zostały utrzymane w mocy. Uzupełniono je o dodatkowe rozwiązania dotyczące identyfikowania interesariuszy, planowania zarządzania nimi oraz zarządzania ich zaangażowaniem.

W ramach pierwszego procesu dodano technikę modelowania użytkowników (ang. *persona modeling*) polegającą na opracowaniu zbiorczej charakterystyki kluczowych interesariuszy projektu z perspektywy ich oczekiwań i interakcji z produktem.

Odnosnie do drugiego procesu dodano jedno nowe zasilenie w postaci informacji o dostępności interesariuszy (ang. *stakeholder availability*) oraz jeden nowy rezultat: przeglądy kamieni milowych oraz plany iteracji.

W zakresie zarządzania zaangażowaniem interesariuszy dodano jedno, nowe zasilenie tj. przeglądy, spotkania i plany oraz trzy nowe techniki i narzędzia, tj.:

- 1) tablice i ekrany zarządzające,
- 2) statystyki dotyczące aktualnej i historycznej wydajności zespołów oraz
- 3) narzędzia komunikacji tj. kompleksowe narzędzia informatyczne wspierające pracę grupową i komunikację z zespołem i zewnętrznymi interesariuszami.

2.10.3. Podsumowanie

The Software Extension to the PMBoK Guide należy przede wszystkim rozpatrywać w charakterze dodatku, a nie samodzielnego standardu metodycznego. Co więcej, ze względu na grono autorów oraz produkt bazowy, jakim jest PMBoK Guide, jego treść i wartość merytoryczna prezentuje zarówno mocne, jak i słabe strony.

Kluczowa wartość płynąca z SWX wynika z odniesienia i zmapowania dotychczasowego dorobku zwinnych metodyk zarządzania projektami do struktury jednego z najpopularniejszych standardów metodycznych, czyli PMBoK Guide. W efekcie PMI umożliwił stosującym standard organizacjom „przetłumaczenie” stosowanych praktyk zwinnych na język PMBoK Guide oraz integrację tychże w ramach szerszego systemu zarządzania projektami w organizacji. Można dzięki temu wykazać kompatybilność metodyk oraz wykazać podejścia zwinne jako jedną z dopuszczalnych modyfikacji oryginalnego standardu, zachowując spójność stosowanych rozwiązań i łącząc je w jedno. Stało się to możliwe, w szczególności dzięki wprowadzeniu różnych modeli cyklu życia projektów: począwszy od modelu predyktywnego, do adaptacyjnego.

Ze względu na to, iż ani PMI, ani IEEE CS nie dysponowały wcześniej własnymi, promowanymi rozwiązaniami dedykowanymi projektom informatycznym, a ich działalność skupiona była na rozwijaniu wiedzy oraz dystrybucji jej w postaci norm, to kształt i treść SWX ma charakter uniwersalny, tzn. niezależny od jakiegokolwiek konkretnej metodyki z rodziny zwinnych. Mimo iż w SWX można znaleźć odniesienia do niektórych z nich, a model rozwoju iteracyjnego i inkrementalnego jest niemal identyczny z rozwiązaniami proponowanymi przez np. SCRUM, to samo opracowanie stanowi niejako podsumowanie i zbiór rozwiązań metodycznych bazujący na całym dostępnym dorobku metodyk zwinnych – a więc wielu różnych metodykach oraz ich zaleceniach. Dzia-

łanie takie było możliwe ze względu na relatywne opóźnienie opracowania standardu w stosunku do dynamiki rynkowej i mody na Agile Project Management. W SWX wyciągnięto niejako „przed nawias” wszystkie elementy wspólne oraz odniesiono je do 10 obszarów wiedzy, tradycyjnie kojarzonych z zarządzaniem projektami. W efekcie SWX jest interesującym rozwiązaniem dla osób i organizacji posiadających już wiedzę i doświadczenie w stosowaniu metodyk zwinnych i gotowych spojrzeć na nie z innej niż zwykle perspektywy, która może być dla początkującego odbiorcy niejasna, nieintuicyjna i trudna w odbiorze. Zapewne, ze względu na prawa autorskie, częściowej zmianie uległo nazewnictwo popularnych narzędzi i technik np. często stosowana do prezentacji przepływu pracy technika Kanban w SWX nazwana została *workflow board*. Uzasadnione wydaje się, aby jednostki pragnące zapoznać się z metodykami zwinnymi sięgnęły po inne źródła na ten temat.

Z drugiej jednak strony, SWX przyjmując za ramę odniesienia 10 obszarów wiedzy daje możliwość uzupełnienia podejść zwinnych o obszary dotychczas rzadziej uwzględniane. Będą to m.in. zarządzanie kosztami, czy zarządzanie zamówieniami. Może to być istotna szansa w dalszym porządkowaniu metodologicznym zwinnych metodyk zarządzania projektami.

2.11. PRINCE2 Agile

Przy omawianiu rodziny zwinnych metodyk zarządzania projektami rzadko napotkamy w tym zestawieniu na hasło PRINCE2¹⁹³. Metodyka ta, mająca przecież podobne jak *agile* źródła w branży informatycznej, związana jest tradycyjnie z kaskadowym modelem rozwoju projektu. Manifest zwinności przeciwstawiał właśnie takim metodykom jak PRINCE2 nowe zachowania i praktyki wytwarzania oprogramowania. Jednocześnie jednak popularność PRINCE2, który ma charakter standardu w Wielkiej Brytanii oraz jest obecny w 150 krajach świata, powoduje, iż zasadnym stało się pytanie o reakcję autorów tej metodyki wobec rozwoju metodyk zwinnych.

Pierwszą próbą adaptacji PRINCE2 do założeń zwinności była metodyka o nazwie XPrince¹⁹⁴, opracowana w 2004 r. przez prof. Jerzego Nawrockiego z Instytutu informatyki Politechniki Poznańskiej¹⁹⁵. Było to twórcze połączenie Extreme Programming, PRINCE2 oraz Rational Unified Process¹⁹⁶. W celu promocji metody powołano Konsorcjum XPrince, jednak nie zyskała ona szerszego rozgłosu. Należy dodać, że było przedsięwzięcie

¹⁹³ PRINCE2, PRINCE2 Agile są zarejestrowanym znakiem handlowym należącym do Axelos Ltd.

¹⁹⁴ <https://web.archive.org/web/20140207224318/http://www.xprince.net/> (dostęp: 24.11.2021).

¹⁹⁵ *XP + Prince2 = XPrince*, <https://wolski.pro/2008/03/xp-prince2-xprince/> (dostęp: 24.11.2021).

¹⁹⁶ *XPrince*, <https://teamquest.pl/baza-wiedzy/project-management/xprince/137.html> (dostęp: 24.11.2021).

indywidualne, bez oficjalnego wsparcia ówczesnych właścicieli metodyki PRINCE2, którzy przez dłuższy czas nie czuli potrzeby odnoszenia się do rozwijającej się konkurencji w zakresie podejść do zarządzania projektami. Zasadniczy zwrot nastąpił dopiero w 2015 r., gdy Axelos opublikował obszerny poradnik zatytułowany *PRINCE2 Agile*¹⁹⁷, a trzy lata później kolejny podręcznik *A Guide to AgileSHIFT*¹⁹⁸. PRINCE2 Agile (P2A) miał spełnić podobną rolę jak publikacja *PMI Software Extension to the PMBoK*, czyli pozwolić połączyć kompleksową, tradycyjną metodykę zarządzania projektami z duchem nowych, zwinnych i dynamicznych czasów ery *agile*. Choć może się to wydawać łączeniem ognia z wodą, to autorzy P2A podkreślają, iż z perspektywy zaleceń PRINCE2 od wersji 2009 nie ma przeszkód, aby poprzez odpowiednie dopasowanie oryginalnej metodyki ukształtować ją zgodnie z podejściem zwinnym¹⁹⁹. Podręcznik P2A ma być w tym zakresie dodatkowym źródłem omawiającym, jak „skonfigurować i dopasować PRINCE2 tak aby mógł być on najbardziej efektywnie stosowany w połączeniu ze zwinnymi zachowaniami, koncepcjami, modelami i technikami”²⁰⁰. Jak podkreślają autorzy, głównymi grupami docelowymi podręcznika są przedsiębiorstwa i organizacje zarządzające metodyką PRINCE2, chcące zaadaptować podejścia zwinne lub już robiące to lokalnie, np. w dziale IT, jak również te z nich, które spotykają się u swoich partnerów biznesowych z takimi podejściami. Autorzy dostrzegają też potencjalne zainteresowanie wśród organizacji, które już zarządzają zwinnie, ale chciałyby zaadaptować PRINCE2 do zarządzania projektami i uporządkowania szerszego środowiska projektowego²⁰¹. Na tą chwilę trudno jest ocenić rozpowszechnienie P2A w organizacjach na świecie. Ruch zwinności rozwija się w pewnym dystansie do metodyk tradycyjnych, jednakże w podręczniku P2A przebija się zauważalny szacunek, jaki mają autorzy do rozwoju i dorobku podejść zwinnych. Przedstawiono w nim historię, podstawy i narzędzia zwinnego zarządzania, jak również załączono pełną, oryginalną treść *SCRUM Guide* autorstwa J. Sutherlanda i K. Schwabera. Autorzy P2A pozycjonują swoją publikację, podobnie jak w przypadku DSDM/AgilePM, jako dodatek, nadbudowę zarządzczą do zwinnych działań wytwórczych i zwinnych zespołów uzupełniając oryginalnego PRINCE2 o zwinne: koncepcje (*concepts*), techniki (*techniques*), obszary problemowe (*focus areas*), zachowania (*agile behaviours*) i modele (*frameworks*).

Całość podręcznika jest dość obszerna, gdyż liczy sobie niemal tyle samo stron, co oryginalny podręcznik metodyki PRINCE2, czyli prawie czterysta stron. Pozostawiając oryginalną strukturę PRINCE2 (pryncypia, tematy, procesy) autorzy przeanalizowali każdy

¹⁹⁷ Axelos Ltd., *PRINCE2 Agile*, The Stationery Office, Londyn 2018.

¹⁹⁸ Axelos Ltd., *A Guide to AgileSHIFT*, The Stationery Office, London 2018.

¹⁹⁹ Axelos Ltd., *PRINCE2 Agile...*, s. 47.

²⁰⁰ *Ibid.*, s. 29.

²⁰¹ *Ibid.*, s. 43.

element oryginalnej metody i opisali, w jaki sposób efektywnie wykorzystać go w środowisku zwinnym. Wśród kluczowych założeń nowego podejścia spotkać można m.in.:

- 1) ukierunkowanie planowania w projekcie na zestawy funkcjonalności oraz kolejne wydania budowanego produktu z wykorzystaniem technik priorytetyzacji oraz koncepcji minimalnego użytecznego produktu (MVP),
- 2) planowanie oparte na określonych w czasie iteracjach czyli *timeboxach*,
- 3) stosowanie samoorganizujących się zespołów oraz podkreślenie znaczenia współpracy przy planowaniu i realizacji projektu,
- 4) nieformalny charakter części dokumentów np. raportów z kamieni milowych (ang. *checkpoint reports*) lub strategii zarządzania komunikacją,
- 5) podkreślenie znaczenia terminów (patrz: *timeboxing*) i jakości w projekcie (stosowanie *definition of done*) oraz określania tolerancji na jakość i zakres projektu w ramach uelastycznienia i gotowości do podejmowania zmian.

Z perspektywy pryncypiów metoda proponuje między innymi²⁰²:

- 1) ciągłą zasadność biznesową – stosowanie pojęcia „wartość” zamiast „korzyść”; budowanie uzasadnienia biznesowego wokół MVP,
- 2) korzystanie z doświadczeń – przyspieszenie pętli informacji zwrotnej między interesariuszami projektu, jak również stosowanie retrospektyw na zakończenie sprintów,
- 3) zdefiniowane role i odpowiedzialności – uzupełnienie struktury zespołu PRINCE o role *agile*,
- 4) zarządzanie etapowe – generalne skrócenie czasu etapów, zastosowanie *timeboxów* oraz wydań w ramach etapów zarządczych,
- 5) zarządzanie przez wyjątki – generalne rozszerzenie zakresu tolerancji w celu dania samoorganizującym się zespołom swobody decydowania w warunkach zmian i niepewności; przyjęcie zerowej tolerancji na opóźnienia czasu w poszczególnych sprintach,
- 6) zorientowanie na produkty – określanie wymagań z wykorzystaniem opisów funkcjonalności i historii użytkownika, stosowanie priorytetyzacji względem kryteriów jakości i jej tolerancji,
- 7) dopasowanie do warunków projektu – P2A proponuje narzędzie o nazwie Agilometr, pomocne w rozpoznaniu warunków i koniecznych adaptacji w zakresie wdrożenia zwinnych praktyk zarządzania.

Dodatkowo P2A na poziomie wartości wprowadza do metody pięć zwinnych zachowań: transparentność, współpracę, intensywną komunikację, samoorganizację oraz eksplorację. Zachowania te powinny być wprowadzone, praktykowane i wzmacniane w ramach zespołu zarządzającego projektem.

²⁰² Ibid., s. 80.

W ramach tematów P2A rekomenduje między innymi następujące zmiany i zalecenia:

- 1) uzasadnienie biznesowe – przede wszystkim uwzględnienie MVP i modelu przyrostowego rozwoju produktu w uzasadnieniu biznesowym, identyfikowanie założeń i szybka ich weryfikacja w celu radzenia sobie z niepewnością,
- 2) organizację – pozostawienie struktury zespołu PRINCE2 wraz z zmapowaniem ról *agile* na metodykę, rolę głównego użytkownika w charakterze „superwłaściciela produktu” (ambasadora biznesowego z DSDM); pozostawienie roli kierownika projektu chociaż z innym, bardziej zwinnym zestawem oczekiwanych postaw i zachowań,
- 3) jakość – większe podkreślenie różnicy między zakresem, a jakością produktu w relacjach z klientem; uwzględnienie rozwiązań pro jakościowych pojęcia długu technologicznego, historii użytkownika, *definition of done*, *definition of ready* i innych w projekcie,
- 4) plany – brak zasadniczych zmian w temacie po stronie PRINCE2; stosowanie planowania iteracyjnego na poziomie funkcjonalności i zespołów; zachowanie harmonogramów na wyższym poziomie projektu, ale przy ich powiązaniu z planowaniem zwinnym; dopasowanie hierarchii planów: plan projektu, plan etapu, plan wydania, plan zespołu (sprintu),
- 5) ryzyko – mimo, że *agile* nie odnosi się wprost do ryzyk, to sam proces, ceremonie itp. przeciwdziałają wielu z nich poprzez m.in. lepszą komunikację i szybszą reakcję na wydarzenia w projekcie; zalecenie ograniczenia formalizmu zarządzania ryzykiem,
- 6) zmiana – brak zasadniczych zmian w temacie po stronie PRINCE2; dopasowanie kształtu i treści produktów planistycznych tak, aby umożliwiały zmiany, a nie je utrudniały; ze względu na sztywny czas i budżet projektu zwykle budżet na zmiany nie będzie funkcjonował, zaś rada ds. zmian raczej nie będzie zajmowała się zmianami technicznymi pozostającymi w gestii zespołu,
- 7) postępy – wykorzystanie technik pomiaru postępów zaczerpniętych z metodyk zwinnych oraz oparcie oceny na zakresie regularnych przyrostów produktu; odejście od pomiaru czasu i kosztu jako miar postępu projektu.

W zakresie procesów PRINCE2 Agile potwierdza aktualność wszystkich siedmiu oryginalnych procesów PRINCE2 i nadal wymaga ich stosowania. Rekomenduje jednak powiązanie i zmapowanie typowego procesu zwinnego dostarczania produktu na siatkę własnych procesów. I tak w procesie przygotowania projektu (ang. *starting up a project*) tworzona będzie wizja i wykaz prac produktu, w ramach procesu inicjowania projektu pojawi się wykaz prac wydania (ang. *release backlog*), a w poszczególnych etapach dodatkowo wykazy prac sprintu (ang. *sprint backlog*). Autorzy P2A podkreślają znaczenie i wartość dobrego przygotowania projektu, zaznaczając jednocześnie, iż metody zwinne, koncentrując się na realizacji, pomijają w większości te obszary problemowe projektu. Mimo iż PRINCE2 kojarzony jest z znacznym zakresem planowania, w nowym podejściu

preferowane jest stosowanie filozofii *just-enough-upfront planning*²⁰³. Odnośnie do procesu zarządzania strategicznym projektem położono nacisk na funkcjonowanie komitetu sterującego bardziej działającego jako właściciel produktu, skupionego na dostarczonym zakresie oraz korzystającego z bogatszej komunikacji, uczestniczącego w przeglądach itp. Jedną z silnych stron PRINCE2 był kształt procesu zarządzania etapem, który wyposażał kierownika projektu w niezbędne narzędzia zapewniające sprawny przebieg prac projektowych. W środowisku zwinnych musiało się to oczywiście zmienić. Autorzy rekomendują w tym zakresie oparcie etapów na serii timeboxów lub/i wydań, powiązanie „grup zadań” z jednym lub większą liczbą sprintów, raportowanie postępów z wykorzystaniem narzędzi agile oraz procedur, które pozwalają również identyfikować problemy i ewentualne ryzyka. W największym zakresie zwinność P2A wdrażana jest tam, gdzie można by się tego najbardziej spodziewać, czyli na poziomie procesu zarządzania dostarczaniem produktu. Kierownik projektu współpracuje w nim *de facto* z zespołem scrumowym, uczestnicząc wspólnie w spotkaniach planowania sprintu, czy spotkaniach planowania wydania. Raportowanie może odbywać się w formie ustnej, zaś opisy produktów, zawierając charakterystykę pracy do wykonania, nie powinny wchodzić zbyt szczegółowo we wskazywanie, jak ma być to zrobione. Autorzy P2A przywołują również metody takie jak Kanban, ScrumBan oraz LeanStartup jako pomocne w zarządzaniu dostarczaniem produktu. Odnośnie do zarządzania końcem etapu P2A akcentuje zbieżność z podejściami zwinnymi w zakresie zakończenia prac w etapie, podsumowania rezultatów, wyciągnięcia wniosków oraz zaplanowania dalszego toku przedsięwzięcia (zwykle kolejnych wydań). Procesy rozpoczynające projekt, jak i zamykające go, są zwykle w podejściach zwinnych albo traktowane po macoszemu, albo w ogóle pomijane. Produkt i jego poszczególne przyrosty są przekazywane do użytkownika na bieżąco w kolejnych wydaniach, a przez to i działania realizowane na końcu projektu mają miejsce wcześniej. Mimo to nadal projekt należy formalnie zamknąć i podsumować. P2A proponuje w tym zakresie formę warsztatów lub/i przeglądu ostatniego wydania.

Obok opisu zwinnego (lub „uzwinnionego”) postępowania w ramach istniejącej struktury zaleceń metodyki PRINCE2 podręcznik P2A dostarcza czytelnikowi dodatkowych wyjaśnień i zaleceń w zakresie m.in.: dopasowywania metodyki z wykorzystaniem Agilometru, wymagań projektowych, efektywnej komunikacji, częstych wydań produktu, kontraktowania i kształtowania umów w środowisku zwinnym, jak również oferuje listę kontrolną oceniającą kondycję projektu – *P2A healthcheck*.

Ocena omawianego rozwiązania przysparza wiele trudności. Wpływają na nią zarówno czynniki endogenne, jak i egzogenne. Metodyki i podejścia zwinne – i to na tym etapie jest chyba oczywiste – powstały w opozycji do metod takich jak PRINCE2. Miały

²⁰³ Ibid., s. 188.

w zamyśle swoich autorów być antytradycyjne, zbuntowane przeciw kierownikowi projektu i staremu, planistycznemu porządkowi rzeczy – lekkie, ewolucyjne, a nie planistyczne. Tymczasem w 2015 r. do drzwi rodziny zwinnej metodyk zwinnych zapukał dawno nie widziany krewny, od którego rodzina ta wcześniej uciekła, a który chciałby z nią na powrót zamieszkać i ogrzać się przy wspólnym kominku. Oczywiście takie obrazowe przedstawienie sprawy nie oddaje całego problemu i jest w pewnym stopniu krzywdzące dla tej metodyki. PRINCE2 ma bardzo silną pozycję na świecie i nadal stosowany jest w przypadku realizacji projektów informatycznych w wielu krajach. Kwestie tego, czy zwinność kłóci się z PRINCE2, czy i jak łączyć te podejścia, są niezwykle istotne np. dla przedsiębiorstw i organizacji brytyjskich czy australijskich, gdzie PRINCE2 jest obowiązującym standardem. Objaśnienie, iż podejścia są kompatybilne i wskazanie sposobu na adaptację zwinności (*de facto* standardu branżowego) w środowisku PRINCE2 (standardu na poziomie państw) było niezwykle potrzebne i wartościowe. W tym zakresie Brytyjczycy poszli tą samą drogą, co Amerykanie w Project Management Institute z PMBoK-iem. Drugim argumentem za zainteresowaniem się P2A jest fakt, iż podejście to funkcjonuje na podobnych zasadach jak DSDM/AgilePM. Jest odpowiedzią na relatywnie ubogi zakres oryginalnych zaleceń metod zwinnych ograniczony niemal wyłącznie do problemów wytwarzania oprogramowania w warunkach zmieniających się wymagań. P2A jest zatem źródłem zaleceń w stosunku do całego cyklu życia projektu, odnoszącym się do jego inicjowania i definiowania, organizacji zespołu także na szczeblu biznesu, tworzenia i utrzymywania uzasadnienia biznesowego w warunkach zmienności, planowania przedsięwzięcia na wyższych poziomach niż tylko sprint czy też np. zamykania i rozliczania projektu. Pod tym względem jest to rozwiązanie ciekawe i rozwijające nurt zwinny także i w tym kierunku.

3 METODY SKALOWALNEJ ZWINNOŚCI

Początki elastycznego i adaptacyjnego zarządzania projektami wywodzą się z problemów realizacji projektów informatycznych, których specyfika, złożoność i podatność na niekontrolowane zmiany zakresu powodowały dramatycznie niski odsetek projektów ukończonych powodzeniem²⁰⁴. Wśród dostrzeganych problemów wskazywano w szczególności na: niewłaściwe zrozumienie potrzeb użytkownika końcowego, niezdolność radzenia sobie ze zmieniającymi się wymaganiami, spóźnione odkrycie poważnych usterek projektu, niską jakość oprogramowania, nieakceptowalne wskaźniki wydajnościowe oprogramowania czy brak właściwej komunikacji członków zespołu²⁰⁵. Krytycy stosowania w tym zakresie tradycyjnych metodyk opartych na podejściu kaskadowym podważali ponadto celowość szczegółowego, długookresowego planowania produktów i przebiegu projektu, a także stosowania krępujących norm i standardów zarządzania projektami²⁰⁶.

Rozwiązaniem powyższych problemów stały się opracowane przez wybitnych przedstawicieli branży informatycznej metodyki zwinne oparte na iteracyjnym i inkrementalnym podejściu do wytwarzania produktów oraz na zasadach pracy członków zespołu wskazanych we wcześniejszej części opracowania. Adresatami ww. metod były przede wszystkim zespoły projektowe oraz ich liderzy, zaś same metodyki w większości mają bardziej charakter wytwórczy niż zarządczy. Skupiają się na stronie technicznej i procesach wykonawczych projektów, przy ograniczonych zaleceniach dotyczących strony organizacyjnej i biznesowej przedsięwzięcia. Poniższy schemat ilustruje to zjawisko na przykładzie wybranych metodyk zwinnych.

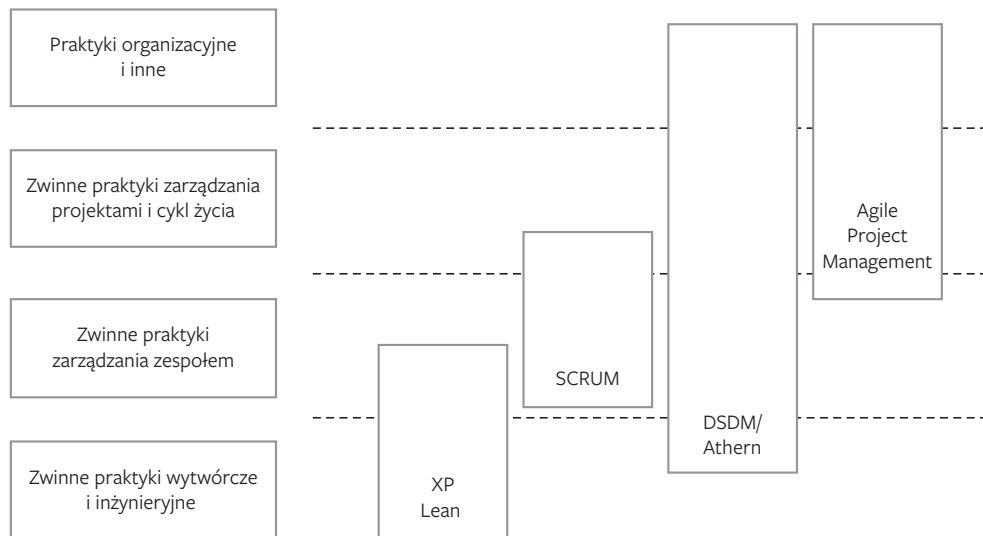
Dodatkowym ograniczeniem stosowania metodyk zwinnych jest wielkość zespołu. Aby spełnić warunki sprawnej pracy, swobodnej komunikacji i samoorganizacji zespoły zwinne utrzymują niewielką liczebność (5–9 osób)²⁰⁷.

²⁰⁴ M. Flasiński, *Zarządzanie projektami informatycznymi*, Wydawnictwo Naukowe PWN, Warszawa 2006.

²⁰⁵ P. Kruchten, *The Rational Unified Process...*

²⁰⁶ P. Wyrozębski, *Elastyczne podejście do zarządzania projektami*, w: *Problemy współczesnej praktyki zarządzania*, S. Lachiewicz, M. Matejun (red.), Wydawnictwo Politechniki Łódzkiej, Łódź 2007.

²⁰⁷ A. Komus et al., *Studie Status Quo Agile 2014...*; G. Asproni, *Wstęp do SCRUM...*; J. Sutherland, K. Schwaber, *The SCRUM Papers...*; P. Wyrozębski, *Zwinne metodyki zarządzania projektami*, w: *Metodyki zarządzania projektami*, M. Trocki (red.), Bizarre, Warszawa 2011; P. Wyrozębski, *Metodyka SCRUM*, w: *Metodyki zarządzania projektami...*

Rysunek 32. Zakres zaleceń wybranych zwinnych metodyk zarządzania projektami

Źródło: Agile Business Consortium, *Agile Project Management White Paper*, https://www.agilebusiness.org/page/Resource_paper_AgilePM (dostęp: 24.11.2021).

O ile zatem wykorzystanie zwinnych metodyk zarządzania projektami wnosi do organizacji znaczny wkład w budowanie zwinnego przedsiębiorstwa, o tyle oddziaływanie skierowane na poziom procesów wytwórczych i do małych zespołów jest niewystarczające z perspektywy całościowego podejścia do systemu zarządzania organizacją, gdyż grozi tworzeniem odizolowanych od siebie, wyspowych i fragmentarycznych rozwiązań. Rozwiązań tego problemu szukać należy w metodach skalujących podejście zwinne z poziomu poszczególnych zespołów, do poziomu całej, zwinnie zarządzanej **organizacji**²⁰⁸. Wśród modeli rozszerzających zakres działania tzw. pierwszej generacji metodyk zwinnych (SCRUM, XP, Crystal) wyróżnić należy m.in.:

- 1) Scaled Agile Framework (SAFe) opracowane przez Deana Leffinwella i Scaled Agile Inc.²⁰⁹,
- 2) Large Scale SCRUM (LeSS) autorstwa Craiga Larmana i Basa Vodde'a²¹⁰,
- 3) Disciplined Agile Delivery (DaD) opracowane przez Scotta Amblera²¹¹,
- 4) LeadingAgile (LA) autorstwa firmy o tej samej nazwie²¹².

²⁰⁸ P. Wyrozębski, *Zwinne koncepcje i modele planowania przebiegu projektów*, w: *Planowanie przebiegu projektu*, M. Trocki, P. Wyrozębski (red.), Oficyna Wydawnicza SGH, Warszawa 2015.

²⁰⁹ Scaled Agile Framework (SAFe): <http://www.scaledagileframework.com/> (dostęp: 24.11.2021).

²¹⁰ Large Scale SCRUM (LeSS): <http://less.works/> (dostęp: 24.11.2021).

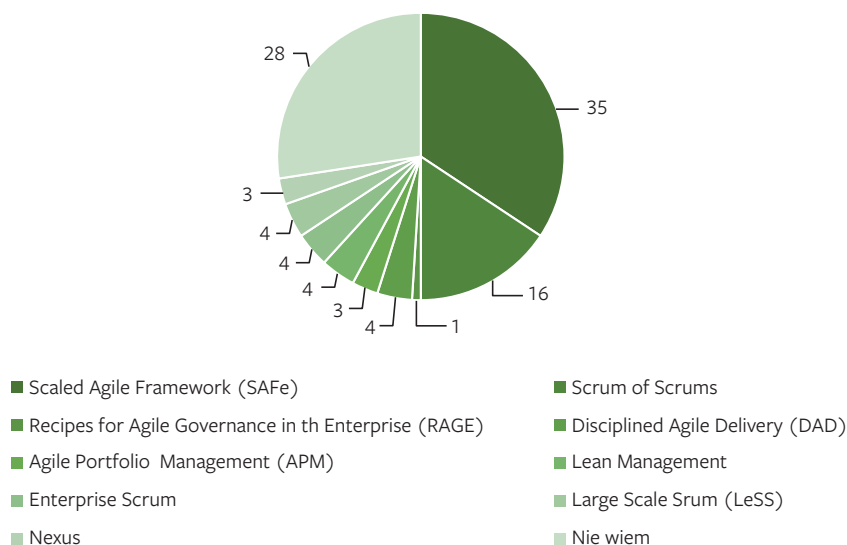
²¹¹ Disciplined Agile Delivery (DaD): <http://www.disciplinedagiledelivery.com/> (dostęp: 24.11.2021).

²¹² LeadingAgile (LA): <http://www.leadingagile.com/> (dostęp: 24.11.2021).

Dodatkowo w ostatnim czasie pojawiły się mniej obszerne rozwiązania, takie jak:

- 1) SCRUM Lean in Motion (Slim) opracowane przez Sama Lainga jako uzupełnienie modelu LeSS²¹³,
- 2) Sustainable Cultural Agile Release in the Enterprise (SCARE)²¹⁴,
- 3) FAST Agile stworzone przez Rona Quartela²¹⁵,
- 4) SCRUM at Scale opublikowane przez Jeffa Sutherlanda (współautora oryginalnej metodyki SCRUM) i SCRUM Inc.²¹⁶,
- 5) Nexus opracowany przez Kena Schwabera (drugiego współautora oryginalnej metodyki SCRUM) i SCRUM.org²¹⁷.

Rysunek 33. Wykorzystanie metod skalowania agile na świecie wg. badań VersionOne (w %)



Źródło: VersionOne, *14th Annual...*, s. 12.

Znaczna liczba rozwiązań wynika z zapotrzebowania praktyki na kompleksowe metody pozwalające stosować *agile* w dużych przedsiębiorstwach i organizacjach. Ich indywidualne znaczenie jest zróżnicowane. Popularność metod ilustrują wyniki cyklicznego

²¹³ SCRUM Lean in Motion (Slim): <http://www.growingagile.co.za/2013/08/slim-an-agile-scaling-pattern/> (dostęp: 24.11.2021).

²¹⁴ Sustainable Cultural Agile Release in the Enterprise (SCARE): <http://itknowledgeexchange.techtarget.com/uncharted-waters/introducing-the-scare-method/> (dostęp: 24.11.2021).

²¹⁵ FAST Agile: <http://www.fast-agile.com/> (dostęp: 24.11.2021).

²¹⁶ SCRUM at Scale: <https://www.scruminc.com/scrum-scale-case-modularity/> (dostęp: 24.11.2021).

²¹⁷ Nexus: <https://www.scrum.org/Resources/The-Nexus-Guide> (dostęp: 24.11.2021).

badania firmy VersionOne dotyczące stanu *agile* na świecie. W badaniu wzięło udział 1121 respondentów z różnych branż, w których wytwarzane jest oprogramowanie komputerowe²¹⁸. W dalszej części opracowania omówione zostaną najważniejsze z nich.

3.1. Scaled Agile Framework (SAFe)

Pośród rozwiązań „dużego *agile*” na szczególną uwagę zasługuje model Scaled Agile Framework (SAFe)²¹⁹. Jego autorem jest Dean Leffingwell, jeden z najbardziej uznawanych światowych ekspertów w zakresie praktyk *lean* i *agile*. Model powstał w 2011 r. i do dziś za jego rozwój odpowiada założona przez autora firma Scaled Agile Inc. Firma ta jest także właścicielem znaku handlowego i praw autorskich do metody, której głównym źródłem są podręczniki autorstwa D. Leffingwella²²⁰ oraz strona internetowa poświęcona modelowi²²¹.

Jest to najbardziej obszerny model, jednoznaczny w swojej konstrukcji, a jednocześnie bardzo popularny i z szerokim wsparciem merytorycznym²²². Według badań VersionOne SAFe stosowany jest w 30% badanych przedsiębiorstwach stosujących podejścia zwinne²²³. Umożliwia przyjęcie zwinnych zasad zarządzania projektami w obszarze informatyki dla dużych, liczących od kilkudziesięciu do kilkuset pracowników, organizacjach i firmach. Model ma charakter kompleksowy. Nie tylko wspiera duże zespoły w wykorzystaniu wybranych praktyk *agile*, ale również proponuje oryginalne narzędzia koordynacji prac, takie jak tzw. *agile release train* (ART), *value streams*, *portfolio backlog* czy *lean budgets*. Metoda integruje poziom zwinnych zespołów, programów oraz portfela i strategii organizacji do jednego predefiniowanego systemu zarządzania organizacją.

Według dostępnych źródeł wdrożenie modelu umożliwia osiągnięcie istotnych korzyści. Jego autorzy, na bazie dostępnych studiów przypadków, zwracają uwagę na wyższy zwrot z zainwestowanego kapitału (ROI), szybszy czas dostarczenia rozwiązań na rynek (30–75%), podniesienie jakości i spadek liczby błędów (25–75%), wzrost produktywności (20–50%), a także poprawę satysfakcji i zaangażowania pracowników oraz poprawę

²¹⁸ VersionOne, *14th Annual State...*, s. 6.

²¹⁹ SAFe i Scaled Agile Framework są zastrzeżonymi znakami handlowymi Scaled Agile Inc.

²²⁰ D. Leffingwell, *Agile Software Requirements...*; id., *Scaling Software Agility: Best Practices for Large Enterprises*, Addison-Wesley Professional, Boston 2007; Scaled Agile Inc., *Achieving Business Agility with SAFe® 5.0*, https://issuu.com/scaledagile/docs/5.0_white_paper_digital (dostęp: 24.11.2021).

²²¹ <https://www.scaledagileframework.com/> (dostęp: 24.11.2021).

²²² M. Heusser, *Comparing scaling agile frameworks*, <http://www.cio.com/article/2974436/agile-development/comparing-scaling-agile-frameworks.html> (dostęp: 24.11.2021).

²²³ VersionOne, *13th Annual State of the Agile Report*, https://www.duxdiligens.com/wp-content/uploads/2019/09/13th-annual-state-of-agile-report_7_May_2019.pdf (dostęp: 24.11.2021).

związku realizowanych przedsięwzięć z potrzebami klientów²²⁴. Wśród firm prezentujących studia przypadków na stronie internetowej autorów metody dostrzec można m.in. Lockheed Martin, PepsiCo, Bosch, Cisco, Deutsche Bahn, Sony Interactive Entertainment, Nordea oraz Accenture.

Od 2011 r. i pierwszej edycji model SAFe przechodził kolejne zmiany. Ich źródłem były doświadczenia praktyczne oraz, w pewnej mierze, głosy krytyczne wobec modelu związane z zakresem wykorzystania zwinnych pryncypiów i wartości, planowania, hierarchii i nacisku na procesy²²⁵. Aktualna wersja nosi numer 5.0 i została opublikowana w styczniu 2020 r. Poczynione zmiany w stosunku do wcześniejszej wersji mają istotny charakter i wprowadzają większą elastyczność stosowania modelu.

Aktualny kształt SAFe zaprezentowany został na kolejnej stronie. Elementami jego struktury są:

- 1) siedem kluczowych kompetencji zwinności organizacji (ang. *business agility*),
- 2) role i odpowiedzialności,
- 3) praktyki zarządzania na trzech poziomach organizacji: podstawowym (essential), dużych rozwiązań (ang. *large solution*) oraz portfela organizacji (ang. *portfolio*).

Model SAFe oparty jest na siedmiu kluczowych wartościach zwinnej organizacji. Dotyczą one zwinności w aspekcie strategii, realizacji działań oraz przywództwa (*lean* oraz zwinnego). Ich zakres autorzy prezentują na poniższym rysunku

Autorzy oferują cztery wersje konfiguracyjne modelu²²⁶:

- 1) podstawowy (ang. *essential*) – bazowy dla pozostałych wersji oraz model startowy do wdrożenia w organizacji,
- 2) duże rozwiązanie (ang. *large solution*) – rozbudowujący potencjał organizacji o zdolność do koordynacji prac przy największych, złożonych produktach przede wszystkim poprzez wprowadzenie tzw. pociągów *agile* (ang. *agile solutions train*, ART) oraz metod współpracy z dostawcami,
- 3) portfel (ang. *portfolio*) – model umożliwiający całościowe zarządzanie organizacją w sposób zwinny; proponuje powiązanie strategii z wizją portfela, a następnie „szczupłym zarządzaniem portfelem” (ang. *lean portfolio management*) i koordynacją strumieni wartości (ang. *value streams*),

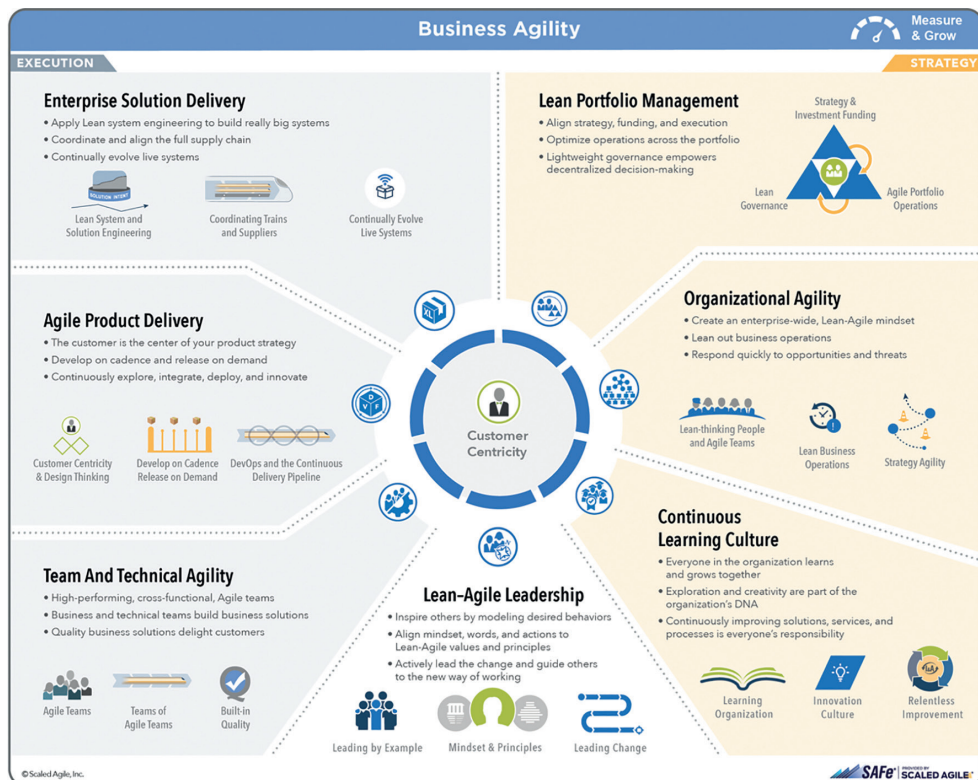
²²⁴ Studia przypadków wykorzystania SAFe: <https://www.scaledagile.com/customer-stories/> (dostęp: 22.04.2020); Scaled Agile Inc., *Applying the Scaled Agile Framework in Outsourcing Context: Infogain Case Study*, <http://www.scaledagileframework.com/infogain-case-study> (dostęp: 22.04.2020).

²²⁵ A. Elssamadisy, *Has SAFe cracked the large agile adoption nut?*, <http://www.infoq.com/news/2013/08/safe#> (dostęp: 24.11.2021); R. Jeffries, *Issues with SAFe*, <https://ronjeffries.com/xprog/articles/issues-with-safe/> (dostęp: 22.04.2020); O. Turetken, I. Stojanov, J.J.M. Trienekens, *Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework*, “Journal of Software: Evolution and Process” 2017, vol. 29, no. 6, DOI: 10.1002/smr.1796.

²²⁶ Scaled Agile Inc., *Achieving Business Agility with SAFe® 5.0...*

- 4) całościowy (ang. *full*) – najbardziej rozwinięta i złożona postać modelu zawierająca w sobie wszystkie opisywane wcześniej poziomy oraz wprowadzająca do organizacji wszystkich siedem kluczowych kompetencji.

Rysunek 34. Siedem kluczowych kompetencji zwinnej organizacji

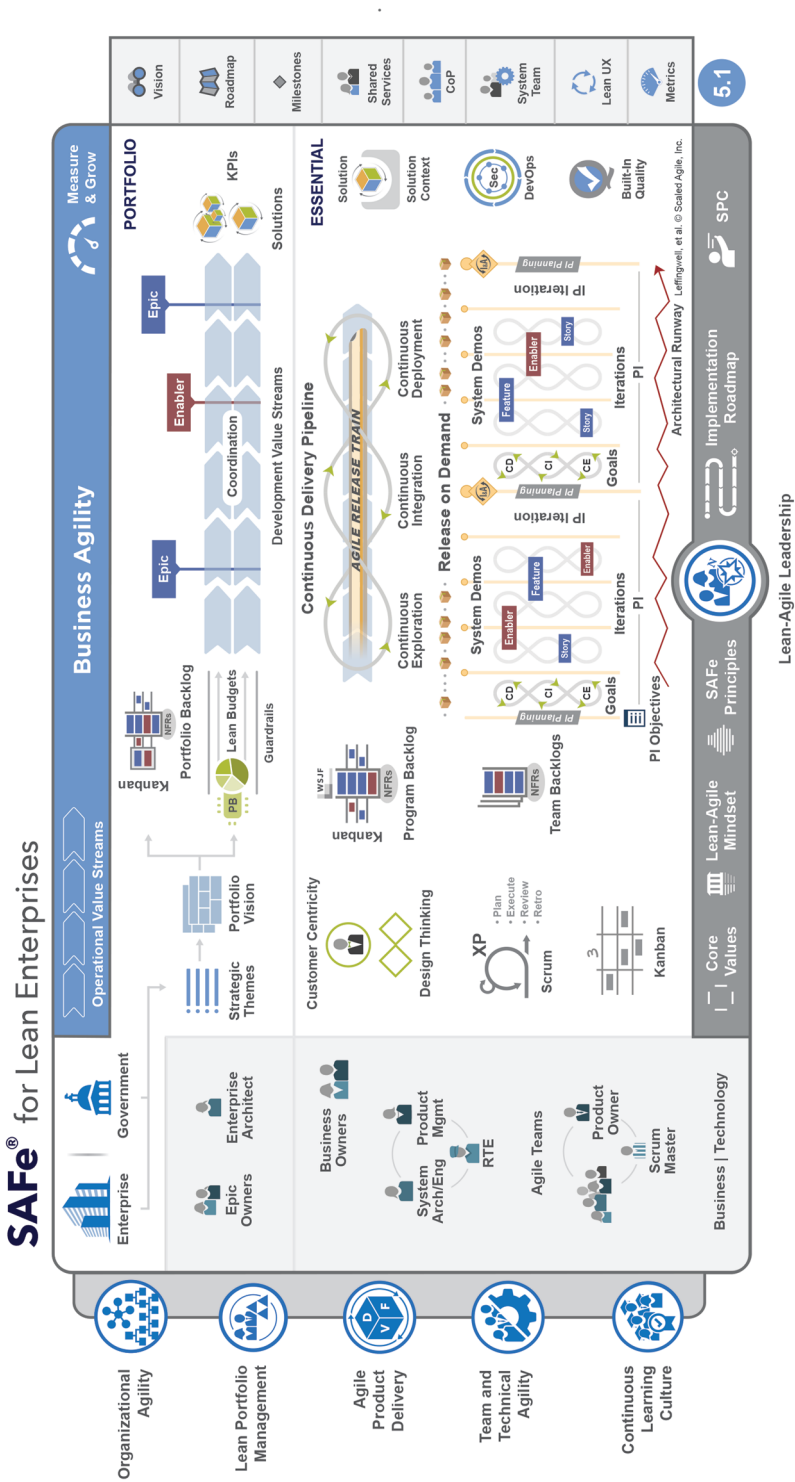


Źródło: <https://www.scaledagileframework.com/whats-new-in-safe-5-0/> (dostęp: 22.04.2020).

Podstawowy poziom SAFe opisuje praktyki realizacji pracy w zwinnych zespołach oraz ich skalowania do większych rozwiązań. Poziom ten powstał poprzez połączenie poziomów zespołów i programu z wcześniejszej edycji modelu.

Kluczowym mechanizmem pracy zapewniającym zwinność organizacji jest tzw. *agile release train* (ART). ART jest funkcjonującym w długim okresie, samoorganizującym się zespołem międzyfunkcyjnych, zwinnych zespołów (ang. *team of agile teams*). Model SAFe odchodzi tym samym od skupienia na pojedynczych projektach, kierując większą uwagę na aspekt skalowania i organizację pracy w programy.

Rysunek 35. Model SAFe for Lean Enterprises (wersja całościowa)



Zgodnie z przyjętą filozofią i w przeciwieństwie do projektów, które mają z góry określony czas realizacji, w przypadku programów kolejne wydania realizowane są dotąd, aż tworzą wartość dla klienta i organizacji. W skład ART-u wchodzi najczęściej od 5 do 12 zespołów liczących od 50 do 125 osób. Celem ich działania jest wytworzenie pożądanego z punktu widzenia organizacji produktu (rozwiązania) opisanego wspólną misją i wykazem prac programu (ang. *program backlog*).

Za definiowanie zakresu opisanego wykazem prac programu, określenie iteracyjnych przyrostów programu (ang. *program increments*) odpowiedzialny jest manager produktu (ang. *product manager*). Podlega mu najczęściej kilku właścicieli produktu (ang. *product owner*) pracujących z poszczególnymi, typowo scrumowymi zespołami developerskimi. Rolę SCRUM mastera na poziomie ART-u pełni tzw. *release train engineer* (RTE) wspierający pracę zespołów. Cykl pracy zespołów developerskich wyznaczany jest poprzez kolejne iteracje tworzące przyrost produktu i łącznie budujące przyrost programu.

W celu zapewnienia koordynacji i spójności przebiegu programu model SAFe zakłada, iż wszystkie zespoły w ramach jednego ART-u razem planują, integrują, demonstrowają, jak również razem się uczą. Jest to możliwe dzięki różnym typom spotkań m.in.:

- 1) spotkaniom planowania przyrostu programu (ang. *program increment planning*),
- 2) spotkaniom retrospektywnym – organizowanym na zakończenie każdej iteracji oraz warsztatom przeglądu i adaptacji (ang. *inspect and adapt*) na zakończenie każdego przyrostu programu.

ART cechuje ustalony harmonogram i rytm, podobny do scrumowych sprintów. Iteracyjna praca zespołów realizowana jest z pomocą SCRUM, XP, metody Kanban oraz innych technik zapewniających jakość wytwarzanego oprogramowania (ang. *build-in-quality*). Wśród dodatkowych elementów autorzy włączyli do poziomu podstawowego także metodę Design Thinking i towarzyszący jej klientocentryzm, a także metodę DevOps.

Zgodnie z modelem najwyższym poziomem zarządzania jest poziom portfela (ang. *portfolio*). Założenia jego realizacji wynikają z szerszego kontekstu działania organizacji oraz założeń strategicznych wyrażonych w modelu przez tematy strategiczne (ang. *strategic themes*) oraz wizję portfela (ang. *portfolio vision*). W skład portfela wchodzi poszczególne przepływy wartości (ang. *value streams*), czyli „serie kroków, w których organizacja wdraża rozwiązania dostarczając klientom ciągły strumień wartości”²²⁷. W ramach rozwiązań umożliwiających skuteczną koordynację działań pod kątem strategicznym SAFe stosuje zwinne i szczupłe budżetowanie (ang. *lean-agile budgeting*), zarządzanie portfelem programów, metodę Kanban dla portfela, wykaz prac portfela (ang. *portfolio backlog*) oraz tzw. epiki portfela, czyli istotne, przekrojowe inicjatywy produktowe i architektoniczne w portfelu. Autorzy wyraźnie odróżniają tu epiki od projektów, wskazując, iż te pierwsze

²²⁷ <https://www.scaledagileframework.com/value-streams/> (dostęp: 22.04.2020).

mają długookresowy, ciągły i nieograniczony w czasie charakter skupiony na dostarczanej wartości²²⁸. W ramach przewidzianych stanowisk model posługuje się dwoma nowymi: właścicielami epików (ang. *epic owners*), pilotującymi epiki przez kanbanowy cykl życia w portfolio oraz architektami przedsiębiorstw (ang. *enterprise architects*) dbającymi o infrastrukturę technologiczną organizacji.

Poziomem pośrednim stosowanym w przypadku realizacji największych systemów jest tzw. *large solution*. Poziom ten jest dodawany w przypadku gdy skala budowanych rozwiązań przekracza zakres pojedynczego ART-a. Dodatkowe role, zdarzenia i artefakty mają pomóc w zapewnieniu sprawnej koordynacji prac oraz zapewnienia współpracy z zasobami zewnętrznymi. Model SAFe grupuje ART-y w tzw. *solution train*, dzięki któremu możliwe jest określanie wizji, backlogu, „mapy drogowej” rozwoju rozwiązania oraz synchronizacja przyrostów programów. Dodatkowymi rolami są *solution train engineer* (STE), kierownictwo rozwiązania (ang. *solution management*), a także architekt rozwiązania (ang. *solution architect/engineering*). Role te są skalowanymi odpowiednikami ról z poziomu ART i zespołów zwinnych.

Scaled Agile Framework jest obszernym i kompleksowym rozwiązaniem zbudowanym przede wszystkim dla dużych i bardzo dużych organizacji. Jego zasadniczą zaletą jest zapewnienie integracji wszystkich poziomów zarządzania wokół idei *lean* i *agile*. Bardzo szczegółowe zalecenia modelu obejmują nie tylko poziom zarządzania operatywnego, tj. poziom zespołów wykonawczych (co jest domeną większości metodyk zwinnych), ale także poziom taktyczny (łączenie pracy zespołów i koordynacja w ramach ART-ów), jak również dotychczas niedostrzegany w metodykach zwinnych poziom zarządzania strategicznego (odniesienie do celów strategicznych, mechanizmy zarządzania, budżetowania i kontroli portfela przepływów wartości). Mechanizmy SAFe wzbogacają także dotychczasowe praktyki umożliwiające uczenie się zespołów zwinnych o rozwiązania integrujące na poziomie organizacji²²⁹. Oprócz wspomnianych wcześniej spotkań i warsztatów SAFe rekomenduje choćby ustanowienie wspólnot praktyków (ang. *communities of practice*), w skład których wchodziłoby członkowie poszczególnych zespołów wytwórczych.

Istotę modelu SAFe i zwinnego zarządzania przedsiębiorstwem odnaleźć można także w tzw. kluczowych wartościach (ang. *SAFe core values*), zasadach myślenia (ang. *agile mindset*) oraz pryncypiach SAFe, które także stanowią elementy modelu.

Autorzy modelu zdefiniowali także ścieżkę wdrożenia modelu SAFe w organizacji²³⁰. Ma ona charakter fazowy i opiera się w pierwszych etapach na szkoleniu agentów zmiany i kierownictwa organizacji, następnie na identyfikacji strumieni wartości i ART-ów,

²²⁸ <https://www.scaledagileframework.com/epic/> (dostęp: 22.04.2020).

²²⁹ Więcej nt. zarządzania wiedzą i uczenia się w zwinnych metodykach zarządzania projektami odnaleźć można w: P. Wyrozębski, *Zarządzanie wiedzą projektową...*

²³⁰ <https://www.scaledagileframework.com/implementation-roadmap/> (dostęp: 24.11.2021).

opracowaniu planu wdrożenia, przygotowaniu uruchomienia pierwszego ART-a oraz szkoleń personelu. W dalszych krokach następuje wsparcie pilotażowego ART-a, uruchomienie kolejnych i rozszerzanie portfela przedsięwzięć. Ostatnią fazą jest przyspieszenie po udanym wdrożeniu.

Wśród wartości przyświecających zwinnemu, inteligentnemu przedsiębiorstwu odnaleźć można: ekonomiczny punkt widzenia, myślenie systemowe, przyrostowe budowane produktów zintegrowane z cyklami uczenia się, uwolnienie wewnętrznej motywacji pracowników wiedzy, decentralizacja procesów decyzyjnych, ale także: utrzymanie spójności zarządzania organizacją na poziomie strategicznym, taktycznym i operacyjnym (ang. *alignment*), zachowanie przejrzystości (ang. *transparency*), doskonałość jakościowa (ang. *build-in quality*) i efektywność realizacji programów (ang. *program execution*).

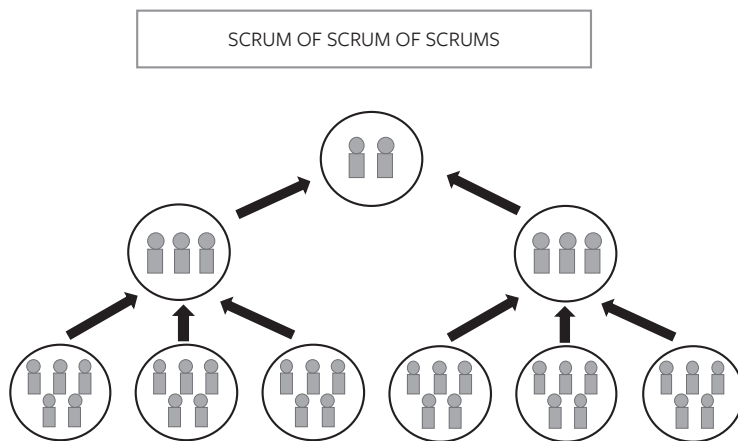
3.2. SCRUM of SCRUMs (SoS)²³¹

Istotą techniki SCRUM of SCRUMs (pol. *młyn młynów*) jest skalowanie metodyki SCRUM do większych grup pracowniczych liczących powyżej 12 osób, które dzielone mogą być na zwinne zespoły o liczebności od 5 do 10 osób. Polega ono na procesie asygnowania przez każdy zespół typu SCRUM przedstawiciela do o szczebel wyższego zespołu SCRUM of SCRUMs. W przypadku, gdy skalowanie ma większy zakres, tj. gdy mamy do czynienia z wielością zespołów SCRUM of SCRUMs, niezbędnym jest stworzenie trzeciego poziomu koordynacji SCRUM of SCRUM of SCRUMs, który tworzony jest przez analogię, tj. wydelegowanie do tego zespołu przedstawicieli każdego z zespołów zwinnych typu SCRUM of SCRUMs.

SCRUM of SCRUMs może być także traktowany jako podejście (ang. *an approach*) do problemu koordynacji między wieloma zespołami ludzkimi, w szczególności koordynacji pracy w środowiskach wewnątrzzespołowych. Delegowanie do zespołu SCRUM of SCRUMs bazuje na kompromisie zespołu deweloperskiego co do tego, kto będzie najbardziej efektywnym komunikatorem – kompetencja ta jest niezbędna do rozwiązywania zagadnień między zespołami. Osoby te mogą się jednak zmieniać wraz z upływem czasu oraz być zastępowane przez innych członków zespołu, którzy mogą bardziej efektywnie realizować zadania w przypadku innego typu problemów występujących w relacjach między zespołami.

²³¹ Rozdział ten powstał na podstawie informacji zawartych na stronach internetowych firm szkoleniowych oraz społeczności sympatyków koncepcji zwinności, w szczególności metodyki SCRUM. Zob. <https://www.agilest.org/scaled-agile/scrum-of-scrums/> (dostęp: 28.04.2020); <https://www.knowledgehut.com/tutorials/scrum-tutorial/scrum-of-scrums> (dostęp: 28.04.2020).

Rysunek 36. Struktura zespołów i ich koordynacja za pomocą metody SCRUM of SCRUMs



Źródło: <https://www.agilest.org/scaled-agile/scrum-of-scrums/> (dostęp: 22.04.2020).

Podstawowymi pytaniami, na które uczestnicy spotkań typu SCRUM of SCRUMs muszą znaleźć odpowiedzi, są analogiczne do codziennych spotkań scrumowych:

- 1) Co mój zespół zrealizował od czasu ostatniego spotkania – co może wpływać na prace innych zespołów?
- 2) Co mój zespół powinien zrobić – co może wpływać na pracę innych zespołów – przed kolejnym spotkaniem?
- 3) Jakiego rodzaju problem, przed którym stoi mój zespół, wymagał będzie pomocy ze strony innego zespołu?

Cele, które dotyczą SCRUM of SCRUMs to przede wszystkim efektywna synchronizacja międzyzespołowa, tak by można było odbywać codzienne spotkania, w które angażowana jest wiedza wszystkich zespołów. Podstawowym motywem jest wspieranie zwinnych zespołów deweloperskich w celu zwiększenia ich produktywności oraz pomoc w przepływie informacji między zespołami. Co więcej, czytamy iż SCRUM of SCRUMs pełni także funkcję platformy do rozwiązywania problemów i podejmowania decyzji.

Agenda spotkań SCRUM of SCRUMs formułowana jest najczęściej za pomocą trzech pytań:

- 1) Co zrobiliśmy wczoraj?
- 2) Co zrobiliśmy dziś?
- 3) Jakiego rodzaju przeszkody napotkaliśmy lub co spowalniało naszą pracę?

O częstotliwości spotkań koordynacyjnych decydować powinny zespoły, biorąc pod uwagę bieżące potrzeby. Autorzy metody sugerują, żeby odbywały się codziennie, podobnie jak codzienne spotkania zespołów scrumowych, i trwały ok. 15 minut.

W spotkaniach powinny brać udział takie osoby, jak SCRUM master lub *agile coach*, właściciel produktu oraz przedstawiciele zespołów deweloperskich, którzy odpowiedzialni są za dostarczenie cząstkowych produktów.

Z technicznego punktu widzenia SCRUM of SCRUMs jest po prostu przeniesieniem zasad SCRUM na poziom międzyzespołowy, aby wspólnie koordynować działania. Nie wiadomo jednak, czym w rzeczywistości SoS różni się od tradycyjnych struktur organizacyjnych lub organizacji sieciowych. Autorzy opisują SCRUM w próżni, tzn. bez odniesienia się do innych form czy sposobów organizowania działalności, stąd też trudno stwierdzić, czym tak naprawdę – merytorycznie, a nie technicznie – wyróżnia się SoS. Efektem tego jest zamęt pojęciowy i tendencja do nadmiernego oraz często niepotrzebnego wyróżniania dodatkowych definicji, podczas gdy merytorycznie takiej potrzeby nie ma. O ile jest to zrozumiałe z punktu widzenia działalności gospodarczej na rynkach szkoleniowo-doradczych, jako swoista próba sprzedaży takich usług czy marketingu, to nie znajduję dowodów, aby takie ujęcie faktycznie dodawało nowy element do istniejącej wiedzy o organizacji czy zarządzaniu, nie wspominając o fakcie, że brak jest rzetelnych badań naukowych o efektach skalowania metodyk zwinnych.

3.3. Nexus

Metodyka Nexus powstała w celu dostosowania zasad ramowych metodyki SCRUM do pracy większej liczby zespołów projektowych (mówi się o 3–9 zespołach). Jak czytamy w oficjalnym podręczniku: „Nexus to model postępowania składający się z ról, zdarzeń, artefaktów i technik, które łączą i splatają pracę około trzech do dziewięciu Zespołów SCRUMowych (ang. *SCRUM Teams*) pracujących nad jednym Backlogiem Produktu (ang. *Product Backlog*) i wspólnie wytwarzających Zintegrowany Przyrost (ang. *Integrated Increment*) spełniający założony cel”²³².

Istotą Nexusa jest przeniesienie niektórych rozwiązań i pojęć, charakterystycznych dla metodyki SCRUM do poziomu międzyzespołowego, który sprawdzałby się przy realizacji większych projektów, wymagających większej liczby zaangażowanych pracowników lub większej rozpiętości zadań. Sami autorzy metodyki twierdzą, iż Nexus jest odpowiedzią na potrzebę koordynowania pracy nad kodem źródłowym oprogramowania, gdy pracuje nad nim więcej niż jeden zespół typu SCRUM, co tworzy problemy komunikacyjne, integracyjne oraz związane z testowaniem, które to potęgują się w przypadku współpracy więcej niż jednego zespołu deweloperskiego. Podstawowy model Nexus został zaprezentowany na poniższym diagramie, będącym elementem tłumaczenia oficjalnego podręcznika tej metodyki:

²³² K. Schwaber, *Nexus – przewodnik*, <http://www.agile247.pl/wp-content/uploads/2016/02/NexusGuide-v1.1-Polish-nfv3.pdf> (dostęp: 24.04.2020).

3.3.1. Nexus a SCRUM²³³

Tak jak wspomniano na początku, Nexus jest rodzajem narzędzia integrującego prace deweloperskich zespołów scrumowych. Należy więc omówić elementy, które odróżniają Nexus od SCRUM. W zakresie ról Nexus przewiduje stworzenie zespołu integracyjnego Nexusa (ang. *Nexus integration team*), który jest koordynatorem stosowania metodyk Nexus i SCRUM między zespołami. – istnieje by koordynować, uczyć i nadzorować stosowanie Nexusa i SCRUM w celu osiągnięcia jak najlepszych wyników. Zespół ten złożony jest z właściciela produktu (ang. *product owner*), SCRUM mastera oraz członków zespołu integracyjnego. Kluczowym zadaniem tego zespołu jest nadzór nad tym, aby przyrost powstawał przynajmniej raz na sprint. Co istotne, według informacji zawartych w podręczniku, skład zespołu integracyjnego może być zmieniany w zależności od potrzeb bieżących projektu. Jego działania polegają m.in. na coachingu, doradztwie oraz koordynowaniu problemów międzyzespołowych, w szczególności tych związanych z integracją, co „oznacza w tym wypadku rozwiązanie wszystkich – technicznych i pozatechnicznych – międzyzespołowych zależności, które mogą ograniczać zdolność Nexusa do dostarczania stale zintegrowanego przyrostu”.

W kwestii artefaktów ważne jest, żeby wszystkie zespoły deweloperskie używały tego samego backlogu produktu. Jego elementy są następnie przyporządkowywane do konkretnych zespołów, które będą pracowały nad nim w ramach sprintu. Nowym artefaktem jest backlog sprintu Nexusa (ang. *Nexus sprint backlog*), co nie zmienia faktu, że poszczególne zespoły utrzymują właściwe dla siebie backlogi sprintu (ang. *sprint backlogs*).

Funkcjonuje także zdarzenie nazwane planowaniem sprintu Nexusa (ang. *Nexus sprint planning*), polegające po prostu na spotkaniu wyznaczonych przedstawicieli zespołów scrumowych spotykających się, aby przeanalizować przygotowany backlog produktu i podzielić się poszczególnymi jego elementami. Na podstawie tego podziału każdy z zespołów przygotowuje własny sprint. Wynikiem takiego podziału jest zestaw celów sprintu, zgodnych z nadrzędnym celem Nexusa (ang. *Nexus goal*). Kolejnym elementem metodyki jest codzienny SCRUM Nexusa (ang. *Nexus daily SCRUM*), polegający na spotkaniach wyznaczonych przedstawicieli zespołów deweloperskich, których celem jest omówienie problemów związanych z integracją, które po identyfikacji podlegają „przeniesieniu” na poziom konkretnego codziennego SCRUM-a dla danego zespołu. Czym innym jest zdarzenie zwane przeglądem sprintu Nexusa (ang. *Nexus sprint review*), które polega na spotkaniu wszystkich zespołów z właścicielem produktu w celu analizy zintegrowanego przyrostu. W przypadku konieczności aktualizacji backlogu produktu dokonu-

²³³ Tekst znajdujący się w poniższym rozdziale jest opracowany na podstawie polskiego tłumaczenia podręcznika Nexus Guide.

je się tego na takim właśnie spotkaniu. Kolejnym elementem jest retrospektywa sprintu Nexusa (ang. *Nexus sprint retrospective*) polegająca na spotkaniu wyznaczonych przedstawicieli zespołów deweloperskich w celu identyfikacji „wspólnych wyzwań”. Nie zmienia to faktu, że każdy z zespołów przeprowadza swoje własne spotkanie w typie retrospektywnym. Po takim spotkaniu ponownie odbywa się narada przedstawicieli zespołów, aby omówić wnioski wynikające ze spotkań w zespołach deweloperskich.

Wydaje się, że Nexus tak naprawdę nie wnosi do metodyki SCRUM niczego nowego. Jest to wyłącznie kwestia zmiany nazewnictwa. Analizując treść metodyki nie sposób pozbyć się wrażenia, że ornamentyka przeważa nad treścią. Co więcej, autorzy metodyki w żaden sposób nie prezentują wyników badań czy ewaluacji, które pozwoliłyby na rzetelne stwierdzenie, że implementacji metodyki Nexus na poziomie wielu zespołów scrumowych przynosi spodziewane efekty, a więc wzrost efektywności prac zespołów oraz jakości produktu dostarczanego klientowi. Nie istnieją wobec powyższego żadne dowody na to, że Nexus jest czymś więcej niż luźną propozycją zespołów konsultantów, na podstawie której stworzono osobny produkt na rynek szkoleniowy i doradczy. Istnieją, co prawda, przykłady przedsiębiorstw, które zastosowały tę metodykę, nie ma jednak żadnych szczegółowych informacji w zakresie liczb czy danych, które pozwoliłyby na zweryfikowanie informacji przedstawionych w źródłach internetowych²³⁴.

Analiza treści metodyki każe stwierdzić ponadto, że używa ona tautologii w sensie logicznym. Występuje w niej niezwykle nagromadzenie zdań, których treść spełnia kryteria tego typu wyrażen, takich jak np.:

Im większa złożoność i im więcej zależności, tym więcej pracy trzeba włożyć w doskonale nie elementów backlogu produktu, aby te zależności usunąć.

Wizualizacja umożliwia zespołom monitorowanie i zmniejszanie do minimum liczby i wpływu zależności.

W dużej skali, backlog produktu musi być zrozumiały na takim poziomie, aby umożliwiał wykrywanie i minimalizowanie zależności.

Zakres poszczególnych wymagań może się nakładać, również sposoby ich implementacji mogą na siebie oddziaływać.

Backlog produktu należy zdekomponować w taki sposób, aby można było łatwo zidentyfikować, a następnie usunąć lub ograniczyć zależności.

Takie – i podobne – oczywiste zdania, będące przy tym niekonkretne oraz wieloznaczne, sprawiają, że w praktyce stają się niemożliwe do jakiejkolwiek interpretacji. W odczuciu autora powoduje to wyłącznie zamęt oraz skupianie się na **procesie** zamiast

²³⁴ Zob. <https://kendis.io/framework-for-scaling-scrum/companies-scaling-scrum-with-nexus/> (dostęp: 24.04.2020).

na **celu**. Autorzy podręcznika na kilkunastu stronach zamieścili informacje, które sprzeczają się do tego, że w przypadku współpracy wielu zespołów pracowniczych pojawiają się problemy związane z komunikacją, a te problemy można rozwiązywać za pomocą stworzenia zespołu koordynującego, który będzie złożony z członków poszczególnych zespołów pracowniczych.

3.4. SCRUM@Scale (S@S)

Metodyka SCRUM@Scale stworzona została przez jednego z dwóch pierwotnych twórców bazowej metodyki SCRUM – Jeffa Sutherlanda. Została wprowadzona przez autora m.in. w firmie tworzącej oprogramowanie IDX Systems Corporation, wykupionej później przez spółkę GE Healthcare. Opisując swoje doświadczenia, stwierdza, że jakkolwiek większość zespołów w ramach przedsiębiorstwa IDX S.C. osiągała przeciętne wyniki, kilka zespołów deweloperskich weszło w stan hiperproduktywny, który cechował się cztero- lub pięciokrotnym zwiększeniem średniej wydajności w branży²³⁵. Mimo to konsultanci promujący metodykę SCRUM@Scale oferują skalowanie produktywności (tak aby cała organizacja dostarczała dwa razy wyższą wartość za cenę połowy kosztów), a także generowanie wyższe zwroty dla wszystkich interesariuszy²³⁶. Autorzy definiują S@S jako zasady ramowe, w ramach których zespoły operują zgodnie z metodyką SCRUM, by zaadresować problemy adaptacyjne w dużej skali a jednocześnie dostarczać produkty najwyższej możliwej jakości²³⁷.

Niezbędnym do skalowania SCRUM jest wcześniejsze ustanowienie odpowiedniej kultury organizacyjnej oraz edukowanie członków zespołów deweloperskich, a także wszystkich zaangażowanych pracowników w zakresie treści podstawowej metodyki SCRUM. Istotą S@S jest skalowanie zespołów, tj. stworzenie nowego zespołu typu SCRUM of SCRUMs, składającego się z przedstawicieli poszczególnych deweloperskich zespołów scrumowych. Jest to jednak tylko pierwszy etap skalowania. W większych organizacjach niezbędne może być stworzenie zespołu typu SCRUM of SCRUM of SCRUMs, tak jak na diagramie widocznym poniżej:

Kolejnym elementem niezbędnym do wdrożenia jest skalowanie wydarzeń oraz ról wynikających z bazowej metodyki SCRUM. Spotkanie typu SDS (ang. *scaled daily SCRUM*) oraz typu retrospektywa są lustrzanym odbiciem spotkania typu *daily SCRUM*/

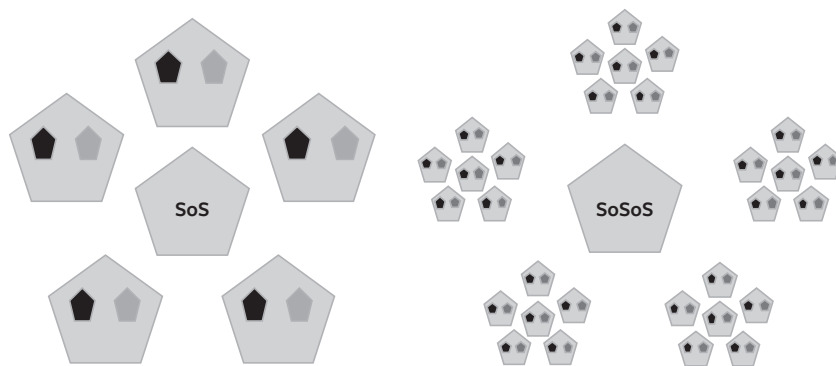
²³⁵ J. Sutherland, *Inventing and Reinventing SCRUM in Five Companies*, https://www.researchgate.net/publication/238252809_Inventing_and_Reinventing_SCRUM_in_Five_Companies (dostęp: 24.04.2020).

²³⁶ www.SCRUM-atscale.org (dostęp: 24.04.2020).

²³⁷ Rozdział napisany na podstawie informacji zawartych na oficjalnej stronie internetowej metodyki, www.SCRUM-atscale.org (dostęp: 24.04.2020).

retrospective w metodyce bazowej, z tą różnicą, że odbywają się na kolejnych poziomach skalowania zespołów.

Rysunek 38. Model SCRUM of SCRUM of SCRUMs

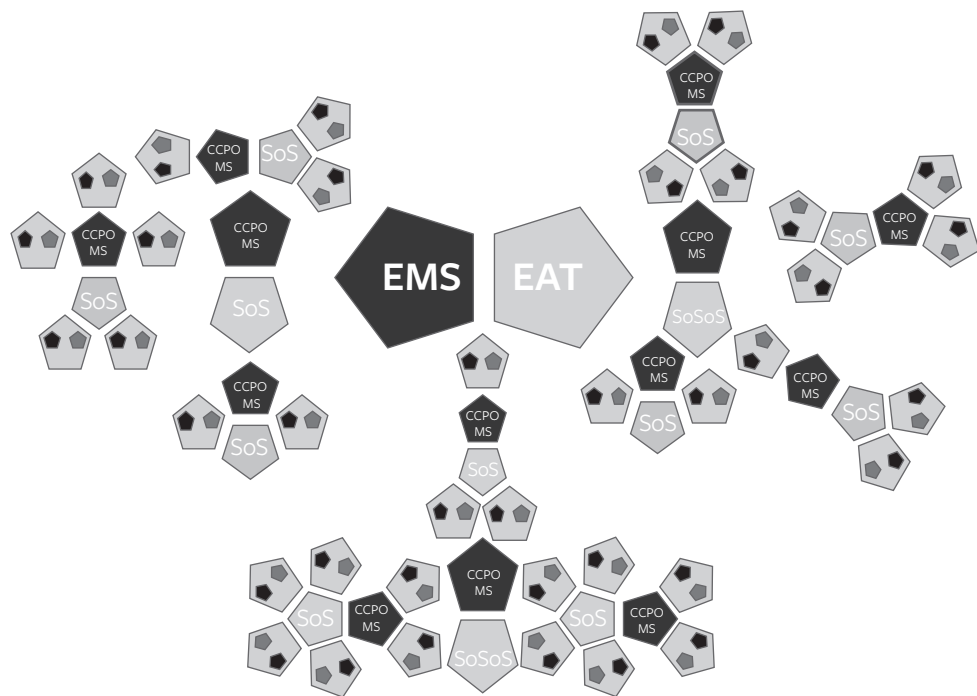


Źródło: <https://www.scrumatscale.com/scrum-at-scale-guide-online/> (dostęp: 4.12.2021).

Jeśli idzie o role, to również stanowią one lustrzane odbicia elementów bazowej metodyki SCRUM. SCRUM master zastąpiony został przez SCRUM of SCRUM mastera, wykonując jednak te same role na skalowanym poziomie zespołu scrumowego. Nowym tworem organizacyjnym jest natomiast EAT (ang. *executive action team*), który pełni funkcje SCRUM mastera dla całej organizacji, zarządzając zwinną transformacją oraz implementacją zasad i kultury wynikających z przyjętej metodyki, a także odpowiada m.in. za backlog. Odpowiednikiem roli typu *product owner* jest natomiast CPO (ang. *chief product owner*), który wykonuje zadania polegające na ustanowieniu wizji strategicznej dla produktów, priorytezuje elementy backloga, które mają być dostarczone przez poszczególne zespoły czy monitoruje informacje zwrotne od klienta.

Pewną nowością względem bazowej metodyki SCRUM jest grupa typu *executive metaSCRUM* (EMS). Jest to swoisty związek liderów, którzy ustanawiają wizję organizacji, ustalają priorytety strategiczne czy zapewniają, że wszystkie grupy w organizacji skupione są na zbieżnych celach. Spotkanie takie odbywa się jeden raz na sprint, a obecni muszą być na nim m.in. CPO oraz przedstawiciele poszczególnych zespołów właścicieli produktów. Grupa ta operuje jako zespół typu SCRUM.

Na rysunku 39 widoczne jest usytuowanie poszczególnych zespołów i ról w ramach struktury organizacyjnej po skalowaniu:

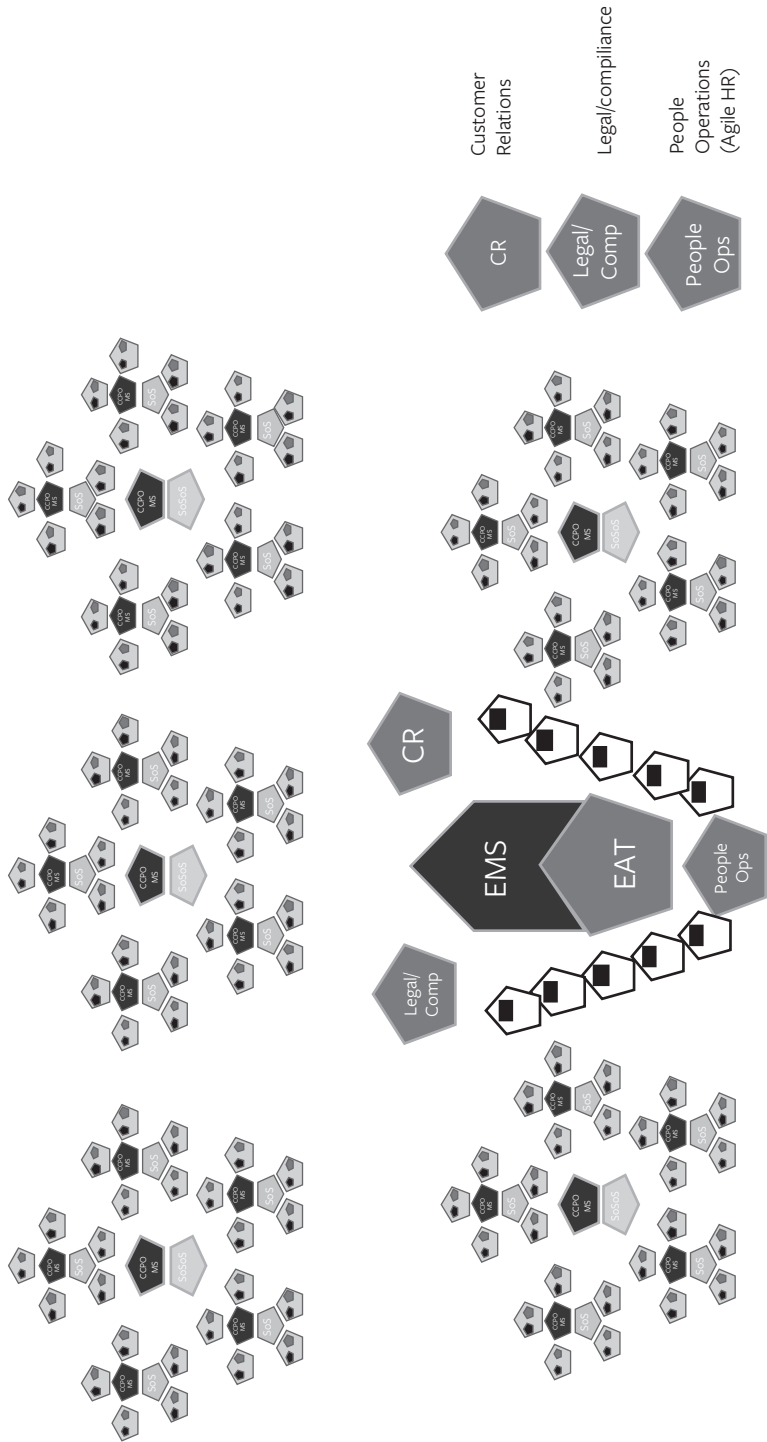
Rysunek 39. Rozbudowana struktura zespołów w modelu S@S

Źródło: *ibid.*

Wydaje się, że nieskończone skalowanie zespołów scrumowych napotyka na pewne przeszkody nieodłącznie związane z biurokracją i funkcjonowaniem wewnątrz dużych organizacji korporacyjnych. Dla przykładu, na oficjalnej stronie metodyki możemy znaleźć także diagram, który pokazuje w schemacie SaS różne funkcje organizacji. Wygląda to jednak po prostu na inną formę graficzną przedstawienia standardowej struktury organizacji hierarchicznej.

Co więcej, należałoby postawić pytania o istotę skalowania metodyki SCRUM. Jeżeli bowiem w założeniu jej implementacja ma wprowadzić wyższą wydajność do zespołów deweloperskich, to czy faktycznie dokonanie tego w formie ukazanej na powyższym przykładzie korzystnie wpływa na efektywność tych zespołów lub efektywność organizacji? Czy koszty komunikacji i integracji wynikające z funkcjonowania w ramach organizacji hierarchicznej i biurokratycznej, które mają być ograniczone przez pracę w metodyce SCRUM będą istotnie niższe w przypadku implementacji SCRUM@Scale, czy też podobne jak w hierarchicznych układach korporacyjnych? Sam autor twierdzi bowiem, że w przypadku implementacji skalowania metodyki SCRUM większość zespołów deweloperskich pozostała na przeciętnym w branży poziomie efektywności.

Rysunek 40. Model organizacji S@S



Źródło: *ibid.*

3.5. LeadingAgile (LA)

LeadingAgile nie jest *sensu stricto* modelem skalowania metodyk zwinnych na poziom całej organizacji. Jest to założona w 2010 r. w Atlancie przez Mike'a Cottmeyera firma specjalizująca się w doradztwie wyższemu kierownictwu w zakresie zwinnych transformacji dla dużych przedsiębiorstw. W przeciwieństwie do innych modeli, które silnie skupiają się na docelowym kształcie zwinnej organizacji LeadingAgile dostarcza w większym stopniu ścieżkę i rozwiązania dotyczącego samego procesu transformacji. Ze względu na osiągnięte sukcesy w firmach takich jak Walmart, Verizon, Ericsson, Coca-Cola, Amway, czy General Electric²³⁸ firma i jej zalecenia zdobyły rosnące uznanie w społeczności *agile*²³⁹.

Podstawowym źródłem zaleceń ze strony LeadingAgile jest strona internetowa²⁴⁰ oraz zawarte na niej publikacje autorstwa założyciela i partnerów firmy²⁴¹.

Koncepcja LeadingAgile opiera się na czterech pytaniach: dlaczego? co? jak? i kto? dotyczących powodu, treści, przebiegu i ról w procesie zwinnej transformacji. Autor podkreśla, iż powodem transformacji nigdy nie może być transformacja sama w sobie, zaś kluczowe jest jej solidne uzasadnienie biznesowe. Uzasadnienie to najczęściej operuje pojęciami takimi jak: przewidywalność realizacji zobowiązań, jakość budowanych rozwiązań, oszczędności kosztów, szybszy zwrot z inwestycji, lepsze dopasowanie produktu do potrzeb klientów oraz wzrost innowacyjności²⁴². LeadingAgile przygotowuje także organizacje pod kątem przejścia procesu organizacyjnej zmiany w zakresie trzech wymiarów: kultury organizacyjnej, systemów i praktyk. Dodatkowymi aspektami do rozważania jest rozmiar organizacji, opór przed zmianą oraz skala współzależności wewnątrzorganizacyjnych ograniczających samodzielność zespołów.

W pracy autor rozwiązania kładzie nacisk na trzy kluczowe elementy będące, jego zdaniem, wspólne dla wszystkich podejść zwinnych: backlogi, zespoły oraz działające, przetestowane oprogramowanie²⁴³. Każdy z tych elementów stawia przed organizacją istotne wyzwania do przezwyciężenia na poziomie podstaw i dodatkowo w przeskalowanej, zmnożonej formie (*governance*, struktury, mierniki).

Autorskim rozwiązaniem LeadingAgile jest model „czterech kwadrantów”²⁴⁴. Pozwala on nie tylko ocenić stan i uwarunkowania działania organizacji, ale także umożli-

²³⁸ M. Cottmeyer, *The Three Things You Need to Know to Transform Any Sized Organization into an Agile Enterprise*, <https://www.agilealliance.org/resources/sessions/three-things-you-must-know-to-transform-any-sized-organization-into-an-agile-enterprise/> (dostęp: 23.04.2020).

²³⁹ M. Heusser, *SaFE vs LeSS vs DaD vs LeadingAgile: Comparing scaling agile frameworks*, (dostęp: 23.04.2020).

²⁴⁰ <https://www.leadingagile.com/> (dostęp: 23.04.2020).

²⁴¹ M. Cottmeyer, *Agile Transformation*, <https://www.leadingagile.com/whitepaper/> (dostęp: 23.04.2020).

²⁴² *Ibid.*, s. 6–7.

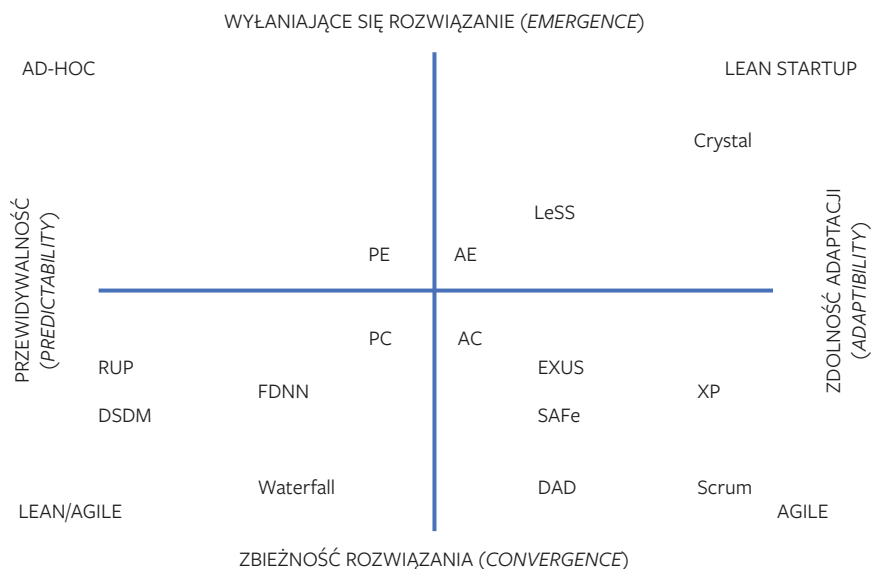
²⁴³ *Ibid.*, s. 13.

²⁴⁴ *Ibid.*, s. 16.

wia wskazanie stanu docelowego, drogi przejścia oraz konkretnych metod pomocnych w każdej z czterech sytuacji. Wymiarami modelu są:

- 1) przewidywalność vs adaptacyjność (ang. *predictability* vs. *adaptability*) – stopień niepewności, w jakim operuje organizacja, możliwość operowania zmianami, stopień w jaki zespoły rozliczane są względem poczynionych zobowiązań,
- 2) wyłanianie się vs zbieżność (ang. *emergence* vs. *convergence*) – cechy rynku, na którym operuje organizacja; czy jest to rynek dynamiczny, podatny na eksplorację, którego wymagania nie są jeszcze zdefiniowane, czy też rynek dojrzały, nakierowany na optymalizację, zgodność i minimalizację ryzyka?

Rysunek 41. Model czterech kwadrantów LeadingAgile



Źródło: M. Cottmeyer, *Agile Transformation*, <https://www.leadingagile.com/whitepaper/> (dostęp: 23.04.2020).

Każdy z czterech kwadrantów charakteryzuje się innymi uwarunkowania działania organizacji i potrzebami metodologicznego wsparcia zarządzania projektami. W przypadku pola tworzonego przez *predictive* i *convergence* (ze względu na relatywną stałość warunków realizacji przedsięwzięć oraz niską podatność na eksplorację i zmiany) sugerowane są tradycyjne lub bliskie tradycyjnym metody zarządzania projektami. Po przeciwległej stronie sytuacja jest skrajnie inna. Oparta jest na filozofii start-upu, gdzie pracujące małe, niezależne od siebie zespoły muszą bazować na odkrywaniu problemów do rozwiązania, eksperymentowaniu i ciągłym uczelni się. Według LeadingAgile ostoją *agile* jest kwadrat adaptacyjność/konwergencja: regularnych dostaw przyrostów i budowania wartości, ale adaptacyjności w sposobie dostawy i organizacji pracy.

LeadingAgile traktuje konstrukcję modelu jako rozwiązanie autorskie, należy mieć jednak na względzie jego wyraźną inspirację modelem CYNEFIN opracowanym w 1999 r. przez Davida Snowdena z IBM Global Services²⁴⁵.

LeadingAgile organizuje proces transformacji w fazowy proces z wyodrębnionymi tzw. obozami wspinaczkowymi (ang. *basecamps*). Model przewiduje pięć obozów:

- 1) stabilizacja systemu – wykorzystanie zwinnych metod pracy zespołów (SCRUM/Kanban) wraz koordynacją na poziomie programu lub zespołu integracji, przy silniejszym nadzorze i koordynacji ze strony zarządzania portfelem,
- 2) zmniejszenie rozmiaru serii – podział większych na mniejsze, łatwiej zarządzane i bardziej elastyczne partie portfela; uzyskuje się dzięki temu skrócenie czasu wejścia na rynek, częstsze wydania i szybszą informację zwrotną od użytkowników,
- 3) ograniczenie współzależności – pomiędzy strumieniami wartości i zespołami, osiągane dzięki kompletowaniu międzyfunkcyjnych zespołów oraz decyzjom inwestycyjnym; LeadingAgile sugeruje na tym poziomie wprowadzenie DevOps, ciągłej integracji oraz ciągłych wydań,
- 4) delegacja decyzji inwestycyjnych – przebudowanie systemu finansowania inwestycji w kierunku wspierającym rozwój produktów i ich komponentów,
- 5) inwestycja w naukę – *de facto* stan docelowy oparty na daleko posuniętej samodzielności operacyjnej, finansowej i biznesowej zespołów oraz umożliwienia im eksploatacji potencjalnych nowych celów biznesowych.

Przejścia między obozami dokonywane są przez tzw. ekspedycje, czyli zgrupowane zespoły (części organizacji) przechodzące transformację.

W aspekcie struktury organizacyjnej LeadingAgile opiera się czterech typach zespołów. Są to: 1) zespoły wytwórcze – typowe mieszane zespoły scrumowe/kanbanowe, 2) zespoły usługowe – zespoły, których usługi/produkty są współdzielone/współużytkowane przez zespoły wytwórcze, 3) zespoły programów pośredniczące między poziomem portfela i zespołami wytwórczymi, 4) zespoły portfela odpowiedzialne za potrzeby biznesowe, finansowanie działalności i ustalanie ograniczeń projektowych realizowanych przedsięwzięć.

LeadingAgile wyróżnia cztery konfiguracje powyższych zespołów oddające poziom skomplikowania budowanych rozwiązań. Przewidywane rozwiązania zaprezentowane są na rysunku 42.

²⁴⁵ D.J. Snowden, M.E. Boone, *A Leader's Framework for Decision Making*, <https://hbr.org/2007/11/a-leaders-framework-for-decision-making> (dostęp: 23.04.2020); R. O'Connor, R. Ie, M. Lepmets, *Exploring the Use of the Cynefin Framework to Inform Software Development Approach Decisions*, w: *The International Conference on Software and System Process (ICSSP)*, Tallinn 2015.

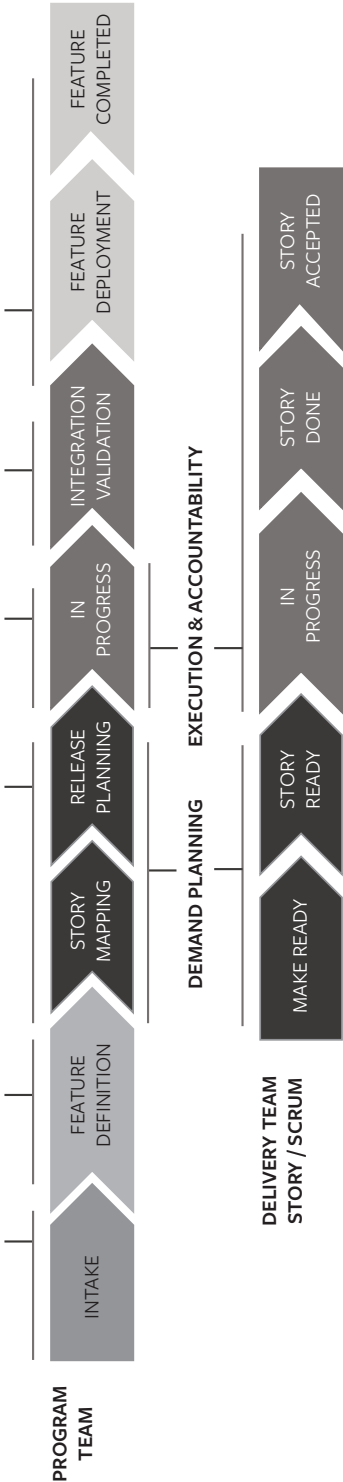
Rysunek 42. Modele organizacji według LeadingAgile

Model jednopoziomowy



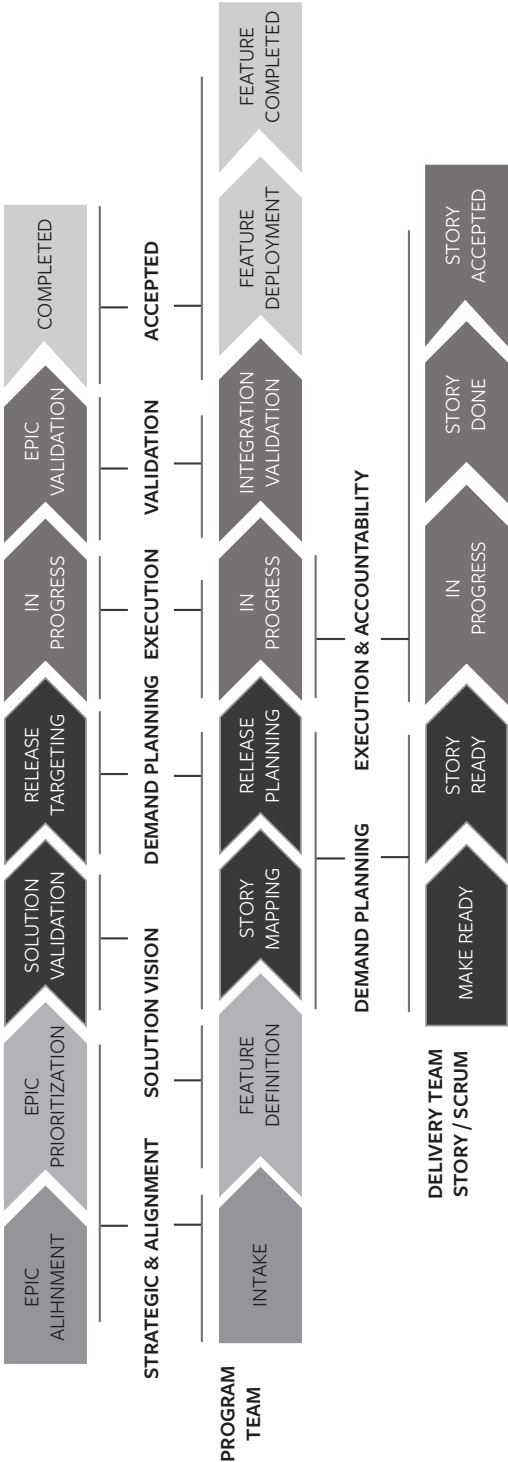
Model dostarczania produktu oparty na pracy jednego zespołu scrumowego. Najprostsze, bazowe rozwiązanie.

Model dwupoziomowy



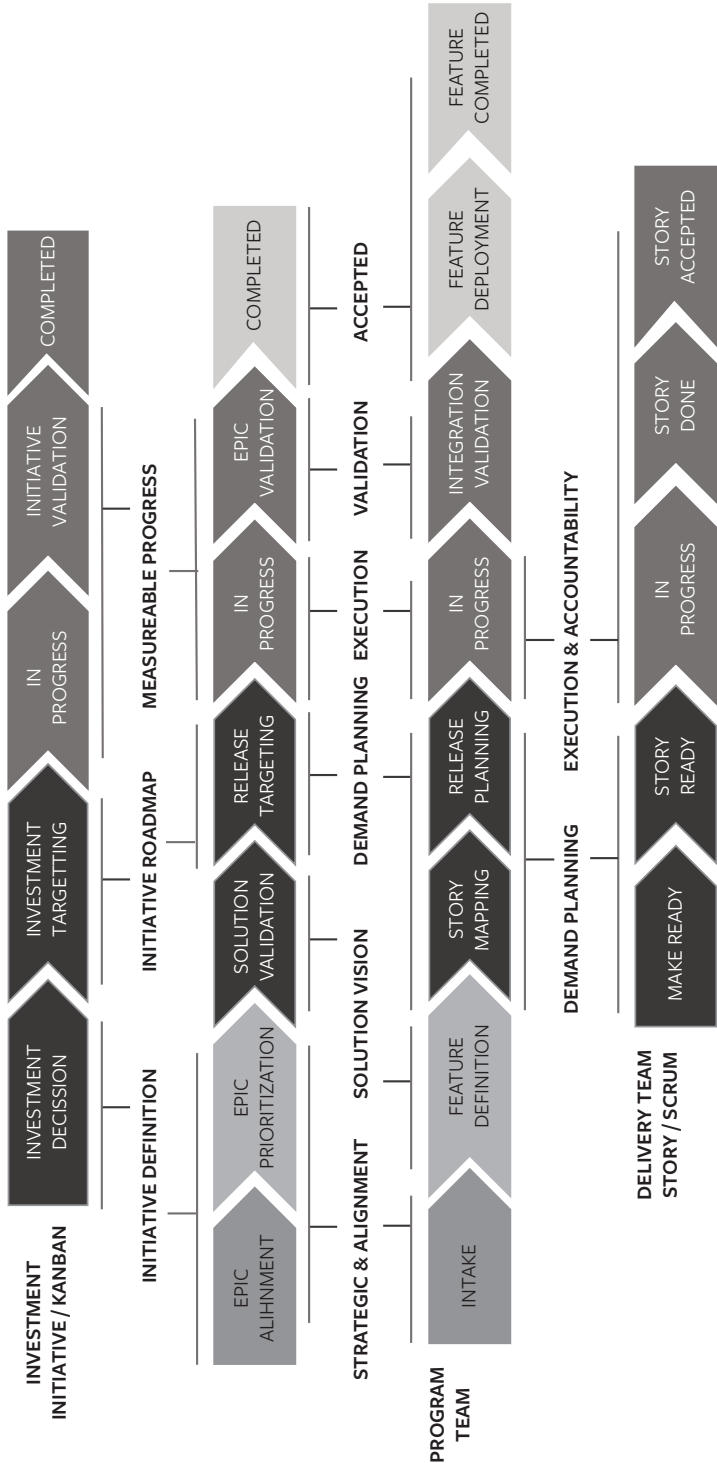
Model zapewniający koordynację zespołów wytwarzających przez zespół programu. LeadingAgile dopuszcza w tym zakresie model SCRUM-of-SCRUMs oraz np. metodę Kanban.

Model trzypoziomowy



Model dla bardziej złożonych środowisk zarządzania.

Model czteropoziomowy



Przeznaczony dla najbardziej rozbudowanych organizacji, z wieloma zespołami wyodrębnionych portfeli realizujących wspólną strategię inwestycyjną.
Źródło: *ibid.*

Istotnym elementem rozwiązań proponowanych przez LeadingAgile jest propozycja procesu transformacji. W ramach modelu zmiany autor proponuje następujące kroki:

- 1) hipotezę transformacji – angażowanie interesariuszy, budowa wraz z liderami organizacji wspólnej wizji transformacji, celów biznesowych i spodziewanych trudności,
- 2) definiowanie stanu docelowego – opracowanie kompleksowego opisu docelowych rozwiązań organizacyjnych oraz planów związanych z wdrożeniem (m.in. *business case*, struktura organizacyjna, *model governance*, model działania zespołów wytwórczych, analiza interesariuszy, plan zarządzania zmianą i inne),
- 3) pilot – uruchomienie „ekspedycji” do „obozu”, czyli transformacja pilotażowego zespołu według opracowanych planów,
- 4) wdrożenie zasadnicze – przeprowadzenie wszystkich kolejnych, przewidzianych zakresem transformacji zespołów do docelowego „obozu”.

W ramach każdego z kroków LeadingAgile opisuje specyficzne podejście (ang. *approach*) oraz oczekiwane rezultaty (ang. *outcomes*), czyli dokumenty i narzędzia związane z danym etapem transformacji. Dla samej transformacji opisane są także wskaźniki pomocne w jej śledzeniu (ang. *transformation outcomes*) oraz wskaźniki biznesowe (ang. *business outcomes*)²⁴⁶.

Ostatnim elementem LeadingAgile są role i stanowiska w zwinnej organizacji. Autorzy proponują w tym zakresie następujące rozwiązania:²⁴⁷

- 1) *transformation lead* – agent zmiany będący partnerem dla osób z najwyższego kierownictwa organizacji, prowadzący proces transformacji i odpowiedzialny za jego wizję i strategię,
- 2) *expedition lead* – kierownik wyprawy, osoba odpowiedzialna za realizację opracowanej strategii i przeprowadzenie organizacji przez zmianę,
- 3) *program/portfolio coach* – rola zapewniająca wsparcie, doradztwo dla zespołu portfela w zakresie nowego, zwinnego modelu działania programu i portfela,
- 4) *agile proces coach* – rola wspierająca zespół ekspedycji w zakresie zwinnych pryncypiów, narzędzi i praktyk oraz doradzająca i szkoląca w zakresie ich wykorzystania,
- 5) *technical coach* – doświadczony, wysoko wykwalifikowany deweloper służący wsparciem dla procesu transformacji zarówno w zakresie zrozumienia i stosowania podejść zwinnych, ale też aspektów technicznych budowanych i utrzymywanych rozwiązań,
- 6) *analyst* – taktyczne wsparcie procesu transformacji przede wszystkim w zakresie gromadzenia danych i pomiarów pomocnych w projektowaniu i sterowaniu realizowanej zmiany.

²⁴⁶ Zob. M. Cottmeyer, *Agile Transformation...*, s. 29–34.

²⁴⁷ *Ibid.*, s. 39–45.

W przeciwieństwie do np. SAFe'a, który jest gotowym, preskryptywnym modelem działania organizacji, podejście prezentowane przez LeadingAgile jest nakierowane przede wszystkim na sam proces dojścia do modelu docelowego. Widać w nim doświadczenie pracy konsultantów funkcjonujących w złożonym środowisku interesariuszy dużych firm oraz muszących stawiać czoła wyzwaniom i barierom wdrażania zmian organizacyjnych. Opisane podejście daje organizacjom dużą elastyczność i swobodę w zakresie konstruowania procesu transformacji na podstawie elementów opisanych w podejściu.

Wśród słabszych stron rozwiązania na pewno należy zwrócić uwagę na fakt, iż podejście elementem usług oferowanych przez firmę. Stąd też relatywnie mało jest materiałów i innych źródeł opisujących to podejście, jak również pozwalających ocenić jego faktyczną użyteczność. Niemniej jednak jest to interesujący głos w dyskusji i warto go również zauważyć.

3.6. Sustainable Cultural Agile Release in the Enterprise (SCARE)

Kolejnym innowacyjnym rozwiązaniem w zakresie skalowania *agile* jest podejście o nazwie Sustainable Cultural Agile Release in the Enterprise (SCARE). Nazwa własna nie jest tłumaczona na język polski jednak w wolnym tłumaczeniu mogłaby brzmieć „utrzymywalne (lub zrównoważone), kulturowe wprowadzenie zwinności w przedsiębiorstwie”. SCARE jest bardziej pomysłem na sposób przeprowadzania zmiany niż formalną metodyką transformacji. Nie znane są też przypadki stosowania SCARE²⁴⁸, zaś podawane w opracowaniach główne źródło (podstrona internetowa firmy TechTarget) jest nieaktywne i niedostępne²⁴⁹. Autorstwo metody przypisywane jest M. Heusserowi, który przedstawił ją krótko podczas konferencji Agile2014²⁵⁰. Trudno jest dotrzeć do szczegółów, opierając się na bardzo ograniczonych, dostępnych źródłach²⁵¹.

Pomysłem M. Heussera na wprowadzenie *agile* do organizacji jest skorzystanie z wcześniejszej teorii ograniczeń E. Goldratta. Heusser zaleca rozpoznanie w organiza-

²⁴⁸ https://agile-mercurial.com/2019/02/06/agile-frameworks-fact-sheet/#S_Agile (dostęp: 24.04.2020).

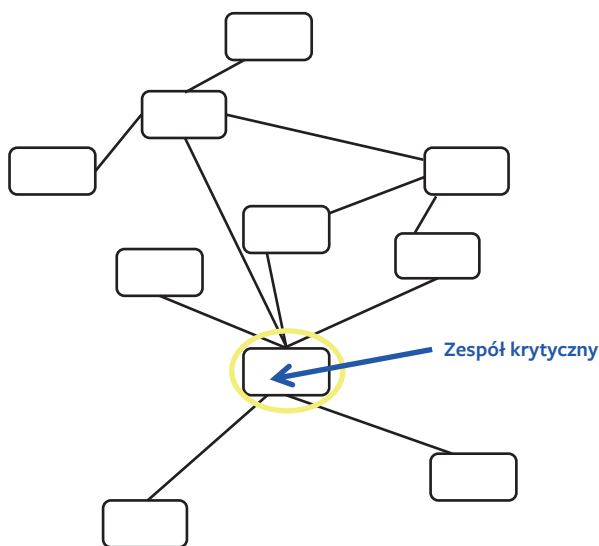
²⁴⁹ <https://itknowledgeexchange.techtarget.com/uncharted-waters/introducing-the-scare-method/> (dostęp: 24.04.2020).

²⁵⁰ M. Heusser, *Learning to (help others learn to) self-organise*, <https://agile2014.sched.com/event/1eE82DI/learning-to-help-others-learn-to-self-organise-matthew-heusser> (dostęp: 24.04.2020).

²⁵¹ *Id.*, *Organize your team with an Agile scaling frameworks comparison*, <https://searchsoftwarequality.techtarget.com/photostory/252477744/Organize-your-team-with-an-Agile-scaling-frameworks-comparison/6/Apply-the-SCARE-framework-to-boost-Agile-productivity> (dostęp: 24.11.2021); *id.*, *What's missing from the large-scale agile frameworks? Practical transition advice*, <https://techbeacon.com/app-dev-testing/whats-missing-large-scale-agile-frameworks-practical-transition-advice> (dostęp: 24.04.2020); J. Žárský, O. Sobisek, J. Bayer, *Sustainable Cultural Agile Release in the Enterprise (SCARE)*, https://spicenter.vse.cz/wp-content/uploads/2018/08/semestralni_prace/18_05/Sustainable_Cultural_Agile_Release_in_the_Enterprise.pdf (dostęp: 24.04.2020).

cji zespołu projektowego będącego zasobem krytycznym, stanowiącym „wąskie gardło”. Następnie następuje praca z nim w celu wykorzystania metod i praktyk zwinnych przede wszystkim w celu podniesienia wydajności jego pracy. Stworzenie wysoce sprawnego, zwinnego zespołu deweloperskiego pozwoli w myśl teorii ograniczeń podnieść wydajność całego systemu i zwiększyć produktywność. Dalsza analiza organizacji wskaże kolejne wąskie gardło, które powinno być przedmiotem kolejnej interwencji i miejscowej transformacji. I tak dalej...

Rysunek 43. Identyfikacja wąskiego gardła w myśl metody SCARE



Źródło: https://agile-mercurial.com/2019/02/06/agile-frameworks-fact-sheet/#S_Agile (dostęp: 24.04.2020).

Według autora zaletą metody SCARE jest jej prostota i skupienie na budowaniu istoty *agile*, czyli zwinnych zespołów. Podejście to nie wymaga według niego przebudowy struktur organizacyjnych, zmian w stanowiskach pracy czy nowych procesów poza samym zespołem i jego członkami. Organizacja jako całość pozostaje stała. Transformacja odbywa się miejscowo, w danym zespole. Metoda ma umożliwiać wprowadzenie realnych zmian, ciągłe doskonalenie i wzrost produktywności w organizacji, a także ma być tańsza i generować niższe ryzyko niż pozostałe „duże” modele²⁵². Rozpowszechnienie *agile* realizowane ma być bezpośrednio i niejako intuicyjnie, bez stosowania narzędzi, takich jak określanie modelu docelowego, strategii, harmonogramów wdrożeń itp. Stopniowe i oddolne wprowadzanie nowego podejścia ma sprawić, iż będzie ono „utrzymywalne/zrównoważone”,

²⁵² M. Heusser, *Organize your team...*

tj. realizowane tylko w takim stopniu, na ile pozwala akceptacja ze strony uczestniczących w niej zespołów, i z czasem oddziałujące (poprzez dyfuzję) na kulturę całej organizacji.

Skąpy charakter źródeł na temat metody nie pozwala dokonać jej szerszej oceny. Wydaje się ona niejako wracać do korzeni zwinności poprzez bardziej ewolucyjny i adaptacyjny model stopniowej transformacji. Poprzez transformacje lokalne, do transformacji organizacji jako całości. Dostępne opisy milczą na temat szczegółów. Wydaje się zatem, iż podejście to pozostanie jedynie ciekawostką i nie będzie rozwijane w przyszłości.

4 ZWINNE TRANSFORMACJE PRZEDSIĘBIORSTW

4.1. Wprowadzenie

Zarządzanie projektami jest uznawane za jedną z najbardziej dynamicznie rozwijających się dziedzin zarządzania. Wykorzystanie projektów jako narzędzia realizacji coraz większego zakresu celów przedsiębiorstw i organizacji nazwane zostało zjawiskiem „projektyfikacji” i badane jest od połowy lat 90. XX w przez licznych specjalistów²⁵³. Wraz z postępującą popularyzacją projektów i zarządzania nimi zaobserwować można jednocześnie wewnętrzne trendy i zróżnicowane kierunki zmian w ramach tego obszaru²⁵⁴. Zmiany te są odpowiedzią na nowe warunki i wyzwania, przed którymi stają współczesne przedsiębiorstwa i organizacje. W szczególności dotyczą one tzw. czwartej rewolucji przemysłowej oraz zmian niesionych przez cyfryzację społeczeństwa i gospodarki, czyli tzw. Gospodarkę/Przemysł 4.0²⁵⁵ oraz tzw. wywrotowe technologie (ang. *disruptive technologies*)²⁵⁶.

²⁵³ Ch. Midler, „Projectification” of the Firm: The Renault Case,,Scandinavian Journal of Management” 1995, vol. 11, no. 4, s. 363–375; Y.-G. Schoper et al., *Projectification in Western Economies: A Comparative Study of Germany, Norway and Iceland*, „International Journal of Project Management” 2018, vol. 36, no. 1, s. 71–82; A. Nieto-Rodriguez, *The Focused Organizations*, Gower, London 2012; D. Norton, *Five reasons organizations...*; A. Wald et al., *Towards a Measurement of “Projectification”: A Study on the Share of Project-Work in the German Economy*, w: *Advanced Project Management: Flexibility and Innovative Capacity*, A. Wald et al. (eds.), Nürnberg 2015, s. 18–36.; M. Trocki, *Projekty w organizacji – problem i wyzwania strategiczne*, w: *Księga jubileuszowa dla Profesora Marii Romanowskiej*, Z. Dworzecki, G. Leśniak-Lebkowska (red.), Oficyna Wydawnicza SGH, Warszawa 2017, s. 195–213.

²⁵⁴ R. Thomsett, *Radical Project Management*, Prentice Hall, Upper Saddle River 2002.

²⁵⁵ M. Hermann, T. Pentek, B. Otto, *Design Principles for Industrie 4.0 Scenarios: A Literature Review*, https://www.researchgate.net/publication/307864150_Design_Principles_for_Industrie_40_Scenarios_A_Literature_Review (dostęp: 24.11.2021); BCG, *Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries*, https://www.bcg.com/publications/2015/engineered_products_project_business_industry_4_future_productivity_growth_manufacturing_industries (dostęp: 24.11.2021); J. Pieriegud, *Cyfryzacja gospodarki i społeczeństwa – wymiar globalny, europejski i krajowy*, w: *Cyfryzacja gospodarki i społeczeństwa – szanse i wyzwania dla sektorów infrastrukturalnych*, J. Gajewski, W. Paprocki, J. Pieriegud (red.), Publikacja Europejskiego Kongresu Finansowego, Instytut Badań nad Gospodarką Rynkową – Gdańska Akademia Bankowa, Gdańsk 2016; Roland Berger, *The digital transformation of industry*, Federation of German Industries (BDI) & Roland Berger Strategy Consultants, Berlin 2015; PWC, *Industry 4.0: Building the digital enterprise*, <https://www.pwc.com/gx/en/industries/industries-4.0/landing-page/industry-4.0-building-your-digital-enterprise-april-2016.pdf> (dostęp: 24.11.2021).

²⁵⁶ T. Zaręba, *Self aware networks – cechy techniczne i implikacje technologiczne*, w: *Cyfryzacja gospodarki i społeczeństwa...*; C.M. Christensen, *Przełomowe innowacje*, Wydawnictwa Profesjonalne PWN, Warszawa 2010.

W ostatnich latach popularność zyskała koncepcja dobrze oddająca obserwowany wzrost dynamiki otoczenia: model VUCA. Został on opracowany jeszcze w końcu lat 80. XX w. i z powodzeniem stosowany w kolejnych dekadach na potrzeby Armii Stanów Zjednoczonych w celu szkolenia oficerów w lepszym stopniu przygotowanych na dynamikę i wyzwania postzimnowojennego pola walki²⁵⁷. Według W.E. Whitemana, który przytacza z kolei tezy Oscara G. Minka, w sytuacjach które charakteryzują warunki takie, jak zmienność (ang. *volatility*), niepewność (ang. *uncertainty*), złożoność (ang. *complexity*) oraz niejednoznaczność (ang. *ambiguity*) konieczne jest budowanie struktur organizacyjnych w sposób odpowiadający wyzwaniom stawianym przez takie środowisko²⁵⁸. Problemy te dostrzegane są w perspektywie wszystkich obszarów zarządzania współczesnymi organizacjami, w tym także przez praktyków i badaczy z zakresu zarządzania projektami. W kolejnych raportach Project Management Institute, największego na świecie, globalnego stowarzyszenia profesjonalistów zarządzania projektami, jednoznacznie wskazuje się na konieczność wdrażania zmian w strukturach organizacyjnych zgodnych z duchem Whitemana, tzn. wspierania kultury przywództwa nastawionego na współpracę, nakierowania na ciągłe uczenie się, pełnego zaangażowania pracowników, dostarczania wartości, wsparcia zwinności organizacyjnej czy stworzenia warunków do odważniejszego eksperymentowania z innowacjami²⁵⁹. Zarysowane powyżej problemy zarządzania organizacjami w doskonały sposób wpisują się rozwiązania znane w zarządzaniu projektami od początku lat 90. XX w., czyli podejście zwinne (ang. *agile project management*). Będące początkowo wyrazem sprzeciwu programistów wobec założeń tzw. tradycyjnego zarządzania projektami, traktowanych jako sztywne i oparte na mikrozarządzaniu zespołem, z czasem docenione zostało jako praktyczny zbiór metod i praktyk zarządzania realizujących założenia uelastycznienia organizacji, budowania zdolności do zmian, adaptacyjności do niestabilnych uwarunkowań otoczenia oraz wzmocnienia pozycji silnie zintegrowanych, zorientowanych na rezultaty, płaskich, samoorganizujących się zespołów zadaniowych. Wśród zasad realizowanych w praktyce przez metodyki zwinne wskazać można te wprost korespondujące z założeniami przedsiębiorstwa w świecie VUCA. Zwinne metodyki zarządzania projektami są źródłem praktycznych zaleceń odnoszących się do sposobu działania zespołów zadaniowych. Ich poznanie i wdrożenie jako modelu realizacji projektów buduje szereg kompetencji organizacji. Według Com-

²⁵⁷ US. Army Heritage & Education Center, *Who first originated the term VUCA (Volatility, Uncertainty, Complexity and Ambiguity)?*, <http://usawc.libanswers.com/faq/84869> (dostęp: 31.01.2020).

²⁵⁸ O.G. Mink et al., *Change at Work...*, s. 15.

²⁵⁹ PMI, Accenture, *Forging the Future: Evolving With Disruptive Technologies*, Project Management Institute, Thought Leadership Series 2018, <https://www.pmi.org/learning/thought-leadership/series> (dostęp: 20.01.2020); PMI, KPMG, *Achieving Greater Agility. The Vital Role of Culture and Commitment*, Project Management Institute, Thought Leadership Series 2017, <https://www.pmi.org/learning/thought-leadership/series> (dostęp: 20.01.2020); PMI, *PMI's Pulse of the Profession 11th Global Project Management Survey: The future of Work. Leading the Way with PMTQ*, <https://www.pmi.org/learning/thought-leadership/pulse> (dostęp: 20.01.2020).

mand and Control Research Program Departamentu Obrony USA są to: **zdolność reakcji** (ang. *responsiveness*) na czas na zmiany otoczenia, **wszechstronność** (ang. *versatility*): zdolność do efektywnej realizacji zadań w różnych sytuacjach i warunkach, **elastyczność** (ang. *flexibility*): możliwość korzystania z wielu rozwiązań i płynnego przejścia od jednego do drugiego, **prężność** (ang. *resilience*): zdolność radzenia sobie z nieszczęśliwymi wypadkami, szkodami lub destabilizującymi zakłóceniami otoczenia lub dostosowanie się do nich, **zdolność innowacyjna** (ang. *innovativeness*): zdolność do robienia nowych rzeczy i zdolność do robienia starych rzeczy w nowy sposób, **zdolność adaptacji** (ang. *adaptability*): możliwość zmiany procesów pracy i zdolność do zmiany organizacji²⁶⁰.

Atrakcyjność podejścia zwinnego w zarządzaniu projektami oraz liczne, opisane w literaturze korzyści dla organizacji wynikające z zastosowania go w realizacji projektów i tworzeniu nowych rozwiązań²⁶¹ stały się przyczyną coraz szerszej adaptacji i skalowania metodyk zwinnych z poziomu pojedynczych, małych zespołów do poziomu całej, zwinnie zarządzanej organizacji²⁶². Proces wdrożenia i przebudowy istniejących rozwiązań organizacyjnych w kierunku „uzwinnienia” organizacji nazywany jest w literaturze i praktyce transformacją *agile* lub transformacją *agilową*²⁶³.

Badając zagadnienia zwinnego zarządzania projektami, wskazać można w tym zakresie istotną lukę badawczą. Wynika ona z jednej strony z rosnącego zainteresowania światu nauki i biznesu popularyzacją i adopcją zwinnych podejść do zarządzania projektami, zaś z drugiej strony, ciągłą potrzebą empirycznego studiowania adaptacji i transformacji uzwiniających się organizacji²⁶⁴.

Dynamika praktyki biznesowej wobec wolniejszych działań ze strony świata nauki powoduje, iż dostrzega się znaczny niedostatek publikacji dotyczących ustrukturyzowanego podejścia do wdrażania zwinnego podejścia w organizacjach²⁶⁵. Badacze zagadnień

²⁶⁰ D.S. Alberts, R.E. Hayes, *Power to the Edge*. Re Di Roma-Verlag, Washington 2009, D.S. Alberts, *The Agility Advantage*, DoD Command and Control Research Program, Washington 2011.

²⁶¹ G. Benefield, *Rolling out Agile...*; J. Sutherland, C. Jacobson, K. Johnson, *SCRUM and CMMI Level 5...*; CHAOS Manifesto 2013...; VersionOne, *The 10th Annual State of Agile Report*, <http://www.agile247.pl/wp-content/uploads/2016/04/VersionOne-10th-Annual-State-of-Agile-Report.pdf>, (dostęp: 10.04.2017).

²⁶² P. Wyróżewski, *Zwinne koncepcje i modele planowania...*; H. Portman, *Will the PMO disappear in the scaling agile world?*, Konferencja Portfolio Experience, Warszawa 2017.

²⁶³ L. Barroca, T. Dingsøyr, M. Mikalsen, *Agile Transformation: A Summary and Research Agenda from the First International Workshop*, w: *Agile Processes in Software Engineering and Extreme Programming – Workshops. XP 2019*, R. Hoda (ed.), Lecture Notes in Business Information Processing, vol. 364, Springer, Cham, Montreal 2019, DOI: 10.1007/978-3-030-30126-2_1; R. Pawlak, *Transformacja zwinna w dużych organizacjach*, w: *Zwinne zarządzanie projektami w dużych organizacjach*, P. Wyróżewski (red.), Oficyna Wydawnicza SGH, Warszawa 2020.

²⁶⁴ M. Laanti, O. Salo, P. Abrahamsson, *Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation*, “Information and Software Technology” 2011, vol. 53, no. 3, s. 276–290; T. Dybå, T. Dingsøyr, *Empirical studies of agile software development: A systematic review*, “Information and Software Technology” 2008, vol. 50, no. 9–10, s. 833–859.

²⁶⁵ A. Sidky, J. Arthur, S. Bohner, *A disciplined approach to adopting agile practices: the agile adoption framework*, “Innovations in Systems and Software Engineering” 2007, vol. 3, no. 3, s. 203–2016.

zwinności, m.in. A. Sidky *et al.*, akcentują w szczególności potrzebę uporządkowane podejścia w zakresie badania i formułowania rekomendacji dotyczących: a) gotowości i podatności organizacji i przedsiębiorstw na zwinność, b) praktyk i metod pracy do wykorzystania w działaniach, c) spodziewanych trudności i barier we wdrożeniach oraz d) koniecznych przygotowań organizacji do wdrożenia²⁶⁶. Co ciekawe, niniejsza luka badawcza jest ciągle aktualnym problemem, co znajduje potwierdzenie w pracach np. K. Dikerta *et al.*²⁶⁷.

Uwzględniając powyższe założenia, celem niniejszego rozdziału jest prezentacja problemów, które napotykają współczesne organizacje w procesie opisanych zmian – omówienie zagadnień związanych ze zwinną transformacją w dużej organizacji, w szczególności barier we wdrożeniu zmian i metod ich przewyżczania.

4.2. Wyzwania zwinnej transformacji

W pierwszej kolejności warto rozpoznać powody, dla których zarządzający podejmują się działań zmierzających do uzwinnienia kierowanych przez nich organizacji. Wśród przytaczanych w badaniach empirycznych argumentów pojawiają się przede wszystkim przesłanki dotyczące usprawnienia działalności organizacji, w tym np. zoptymalizowanie procesu wytwórczego czy spłaszczenie organizacji²⁶⁸. Istotnym argumentem jest także podniesienie szybkości reagowania na zmiany, w szczególności zmiany oczekiwań odbiorców na rynku oraz wzrost elastyczności organizacji²⁶⁹. Liczne studia przypadków opisujące transformacje zwinne rzucają światło na zbiór oczekiwanych korzyści wobec wdrożeń²⁷⁰. Wartym przytoczenia w tym względzie źródłem jest artykuł M. Laanti *et al.* przedstawiający analizę opinii dotyczących transformacji zwinnej realizowanej na początku XXI w. w fińskiej Nokii. Badania wykazały, iż po wdrożeniu 60% ankietowanych pracowników zaakceptowało nowe sposoby pracy i nie wyrażało chęci powrotu do wcześniejszego, kaskadowego podejścia w zarządzaniu projektami. Przeciwnego zdania było 9% pracowników, zaś 31% – pozostało niezdecydowanych. Stosunkowo duży odsetek

²⁶⁶ Ibid.

²⁶⁷ K. Dikert, M. Paasivaara, C. Lassenius, *Challenges and success factors for large-scale agile transformations: A systematic literature review*, "Journal of Systems and Software" 2016, vol. 119, s. 87–108.

²⁶⁸ A. Qumer, B. Henderson-Sellers, *A framework to support the evaluation, adoption and improvement of agile methods in practice*, "Journal of Systems and Software" 2008, vol. 81, no. 11, s. 1899–1919.

²⁶⁹ L. Cao *et al.*, *A framework for adapting agile development methodologies*, "European Journal of Information Systems" 2009, vol. 18, s. 332–343; J. Highsmith, A. Cockburn, *Agile software development: the business of innovation*, "Computer" 2001, vol. 34, no. 9, s. 120–127, DOI: 10.1109/2.947100; B. Boehm, *Balancing agility and discipline: A guide for the perplexed*, Springer, Berlin – Heidelberg 2004; M. Lindvall *et al.*, *Agile software development in large organizations*, "Computer" 2004, vol. 37, no. 12, s. 26–34, DOI: 10.1109/MC.2004.231.

²⁷⁰ A. Sidky, J. Arthur, S. Bohner, *A disciplined approach to adopting agile practices...*

osób wahających się badacze tłumaczą tym, iż osoby te nie miały jeszcze dostatecznego kontaktu z nowymi praktykami²⁷¹.

Tabela 10. Postrzegane korzyści z wdrożenia podejścia zwinnego w firmie Nokia

Twierdzenia		Średnia ocen na skali 1–7
1	Podejście zwinne zwiększa przejrzystość rozwoju	5,13
2	Podejście zwinne zwiększa współpracę	5,04
3	Podejście zwinne zwiększa efektywność rozwoju	4,97
4	Podejście zwinne zwiększa autonomię zespołów programistycznych	4,86
5	Podejście zwinne umożliwia wcześniejsze wykrywanie błędów/defektów	4,77
6	Podejście zwinne podnosi jakość produktu	4,70
7	Podejście zwinne sprawia, że praca jest przyjemniejsza	4,61
8	Podejście zwinne sprawia, że praca jest lepiej zorganizowana/zaplanowana	4,50
9	Podejście zwinne sprawia, że praca jest mniej nerwowa/chaotyczna	3,64

Źródło: M. Laanti, O. Salo, P. Abrahamsson, *Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation*, "Information and Software Technology" 2011, vol. 53, no. 3, s. 276–290.

M. Laanti *et al.* zauważyli też, iż postawa wobec transformacji korelowała z długością wcześniejszego doświadczenia z metodykami zwinnymi i tradycyjnymi. Opinie wobec zwinności stawały się bardziej negatywne wraz z dłuższym doświadczeniem w realizacji projektów metodami tradycyjnymi w zakresie: efektywności, zorganizowania oraz wykrywania błędów. Odwrotny trend zaobserwowano w przypadku współpracy. Wraz ze wzrostem doświadczenia w podejściu tradycyjnym opinie o współpracy w podejściu zwinnym poprawiały się²⁷².

Interesującą obserwacją było również porównanie postrzeganych korzyści z wdrożenia zwinności między grupami zwolenników i przeciwników zmian (patrz tabela 11).

Tabela 11. Korzyści z wdrożenia zwinności w porównaniu między grupami zwolenników i przeciwników zmian

Grupa zwolenników	Grupa przeciwników
1. Widoczność i transparentność	
2. Zarządzanie wymaganiami / planowanie iteracyjne	
3. Produktywność / koncentracja / skuteczność / przewidywalność	3. Częsta dostawa / szybkość / reakcja na zmiany

Źródło: *ibid.*

²⁷¹ M. Laanti, O. Salo, P. Abrahamsson, *Agile methods rapidly...*, s. 276–290.

²⁷² *Ibid.*

W obydwu grupach panowała zgodność co do dwóch pierwszych benefitów. Przedstawiciele obu grup doceniali przede wszystkim transparentność i przejrzystość procesu oraz lepsze ich zdaniem zarządzanie wymaganiami wraz planowaniem iteracyjnym. Użyte wyniki mogą świadczyć o zgodności co do postrzegania efektów wdrożeń zwinnych i ich obiektywnym, dostrzeganym również przez przeciwników charakterze. Pozwalają także spojrzeć inny sposób na trudności i bariery we wdrożeniach. Być może należy poszukiwać ich źródeł poza samym zwinnym procesem wytwarzania, a raczej w postawach i zachowaniach organizacyjnych samych pracowników przedsiębiorstw i organizacji.

Istotnym obszarem badawczym dotyczącym adaptacji podejść zwinnych są trudności i bariery napotymane przez przedsiębiorstwa i menedżerów podczas ich wprowadzania do organizacji. Zgodnie z wcześniej przedstawionymi postulatami stanowią one krytyczny obszar luki nie tylko o charakterze naukowym i badawczym, ale też praktycznym. Zagadnienie to wydaje się dość złożone. Badacze problematyki wdrożeń, jak np. M. Lindvall *et al.*, podkreślają, iż w procesie transformacji wyzwanie stanowić będzie nie tylko opracowanie i przyjęcie samych technik oraz procesów, ale także konieczność zbudowania i zintegrowania środowiska projektowego na poziomie ogólnooorganizacyjnym²⁷³. Twierdzą oni, iż „większość trudności we wdrożeniu zwinnych praktyk pochodzi z otoczenia projektów, a nie z stosowania ich w samym projekcie”²⁷⁴. W szczególności podkreśla się rolę i znaczenie umiejętności zarządzania sytuacjami konfliktowymi oraz znajomość zachowań organizacyjnych dla zapewnienia sukcesu procesu wdrażania podejścia zwinnego²⁷⁵.

Wdrażanie metod zwinnych polaryzuje ludzi i interesariuszy na wspierających i „hamulcowych” względem ich użyteczności w organizacji. Krytyczna wobec tego staje się kwestia fundamentalnej zmiany filozofii i wykształcenie nowych zachowań²⁷⁶. Zjawiska te uwidaczniają się ze szczególną siłą, gdy uświadomimy sobie, iż w samym rdzeniu podejść zwinnych leży zmiana paradygmatu i stylu zarządzania. Metodyki zwinne odchodzą od stylu tradycyjnego opartego na wyznaczaniu zadań i ich kontroli na rzecz skupienia się na przywództwie służebnym (ang. *servant leadership*) i współpracy²⁷⁷.

Spośród źródeł o charakterze empirycznym zdecydowanie warto wyróżnić badania prowadzone od wielu lat przez firmę VersionOne Inc. Lektura corocznego raportu *State of the Agile* pozwala dogłębnie poznać stan praktykowania podejść zwinnych na świecie

²⁷³ M. Lindvall *et al.*, *Agile software development...*

²⁷⁴ *Ibid.*

²⁷⁵ B. Boehm, R. Turner, *Management challenges to implementing agile processes in traditional development organizations*, “IEEE Software” 2005, vol. 22, no. 5, s. 30–39, DOI: 10.1109/MS.2005.129.

²⁷⁶ D. Norton, *Five reasons organizations...*

²⁷⁷ S. Nerur, R. Mahapatra, G. Mangalaraj, *Challenges of migrating to Agile methodologies – Organizations must carefully assess their readiness before treading the path of agility*, “Communications of the ACM” 2005, vol. 48, no. 5, s. 72–78.

opracowany na podstawie głosów ponad 1200 respondentów, w niemal dwóch trzecich reprezentujących przedsiębiorstwa liczące ponad 1000 pracowników²⁷⁸.

Tabela 12. Różnice w założeniu i podejściu między tradycyjnym i zwinnym podejściem do zarządzania

	Tradycyjne	Zwinne
Podstawowe założenia	systemy są w pełni możliwe do określenia (specyfikowalne), przewidywalne i mogą być budowane poprzez szczegółowe i rozległe planowanie	adaptacyjne oprogramowanie wysokiej jakości może być tworzone przez małe zespoły przy użyciu zasad ciągłego doskonalenia i testowania w oparciu o szybkie informacje zwrotne i zmiany
Kontrola	zorientowana na proces	zorientowana na ludziach
Styl zarządzania	dowodzenie i kontrola	przywództwo i współpraca
Zarządzanie wiedzą	wiedza jawna	wiedza ukryta
Przypisanie roli	indywidualne – sprzyja specjalizacji	samoorganizujące się zespoły – zachęcają do wymienności ról
Komunikacja	formalna	nieformalna
Rola klienta	ważny	krytyczny
Cykl projektu	ustalony przez charakter zadań	prowadzony rozwojem cech produktu
Model rozwoju	model cyklu życia (kaskada, spiralny lub ich odmiana)	ewolucyjny model dostarczania
Pożądana forma/struktura organizacyjna	mechaniczny (biurokratyczny z wysoką formalizacją)	organiczne (elastyczne i partycypacyjne zachęcanie do wspólnych działań zespołowych)
Technologia	bez ograniczeń	sprzyja technologii zorientowanej obiektowo

Źródło: S. Nerur, R. Mahapatra, G. Mangalaraj, *Challenges of migrating to Agile methodologies – Organizations must carefully assess their readiness before treading the path of agility*, "Communications of the ACM" 2005, vol. 48, no. 5, s. 72–78.

Według wyników raportu kluczowymi wyzwaniami stojącymi naprzeciw transformacji zwinnej są czynniki nie inne niż te związane z kulturą organizacyjną i generalnym oporem przez zmianami – takie jak brak umiejętności i doświadczenia czy brak szkoleń pojawiają się dopiero na szóstej i siódmej pozycji rankingu. Co istotne, wyniki i układ rankingu pozostają spójne i relatywnie stabilne z wynikami zgromadzonymi w poprzednich kilkunastu edycjach badania²⁷⁹.

Warto w tym miejscu powrócić do przytaczanego już wcześniej przypadku firmy Nokia. Podobnie jak w przypadku korzyści, tak i odnośnie do barier M. Laanti *et al.* wyodrębnili opinie dwóch osobnych grup pracowników. Jednakże i tym razem uzyskany ranking trudności okazał się niemal identyczny. Co interesujące, w rankingu znalazł się

²⁷⁸ VersionOne, *14th annual...*

²⁷⁹ <https://stateofagile.com/#> (dostęp: 13.05.2021).

zbiór o innych kategoriach niż w powyżej przytoczonych badaniach. Na pierwszym miejscu znalazły się bowiem zagadnienia związane z samym wdrożeniem metod zwinnych, w szczególności do nieprzygotowanych zespołów, z niedostatecznym zapewnieniem poprawności stosowania czy niespójnymi praktykami²⁸⁰. Na drugiej pozycji znalazło się zarządzanie wymaganiami i planowanie iteracyjne, zaś na trzecim (w grupie „pozytywnych”) – zagadnienia współpracy, współzależności między projektami i „podzlecania” prac (w grupie „negatywnych”) – planowanie i zapewnienie zasobów ludzkich). Bariery te mają zdecydowanie bardziej techniczny, bliższy istocie wykonywanej pracy, charakter.

Tabela 13. Bariery wdrażania podejścia zwinnego w organizacjach wg VersionOne (w %)

1	Ogólny opór organizacji wobec zmian	48
2	Niedostateczne wsparcie liderów organizacji	46
3	Niespójne procesy i praktyki w zespołach	45
4	Kultura organizacyjna sprzeczna z wartościami zwinnymi	44
5	Niewystarczające wsparcie kierownictwa i sponsoring	43
6	Brak umiejętności / doświadczenia z metodami zwinnymi	41
7	Niewystarczające szkolenie i edukacja	39
8	Brak dostępności biznesu / klienta / właściciela produktu	36
9	Wszechobecność tradycyjnych metod rozwoju	30
10	Rozdrobnione narzędzia i dane / pomiary związane z projektem	29
11	Minimalna współpraca i dzielenie się wiedzą	22
12	Zgodność z przepisami / kwestie regulacyjne	16

Źródło: VersionOne, *14th Annual State of Agile Report*, <https://www.qagile.pl/wp-content/uploads/2020/06/14th-annual-state-of-agile-report.pdf> (dostęp: 24.11.2021).

Tabela 14. Badania w NOKII – bariery wdrożenia w podziale na grupy zwolenników i przeciwników adopcji podejścia zwinnego

Grupa zwolenników	Grupa przeciwników
1. Wdrażanie zwinnych metod	
2. Zarządzanie wymaganiami / planowanie iteracyjne	
3. Zależności / współpraca / podział pracy / podwykonawstwo	3. Zarządzanie zasobami / nakładem pracy

Źródło: M. Laanti, O. Salo, P. Abrahamsson, *Agile methods rapidly...*

W literaturze przedmiotu liczni autorzy wskazują na zidentyfikowane przez nich czynniki. I tak na przykład S. Nerur *et al.* wyróżniają ich cztery grupy²⁸¹:

²⁸⁰ M. Laanti, O. Salo, P. Abrahamsson, *Agile methods rapidly...*

²⁸¹ S. Nerur, R. Mahapatra, G. Mangalaraj, *Challenges of migrating to Agile methodologies...*

- 1) zarządzanie i organizację: kulturę organizacyjną, style kierowania, strukturę organizacyjną, zarządzanie wiedzą, system wynagradzania,
- 2) ludzie: umiejętność pracy zespołowej, wysoki poziom kompetencji, relacje z klientami,
- 3) procesy: zmianę z podejścia koncentrującego się na procesach do koncentracji na ludziach i funkcjonalnościach; krótkie, iteracyjne procesy wytwarzania oparte na adaptacyjności; zarządzanie dużymi, skalowanymi projektami; dobór właściwych metod zwinnych,
- 4) technologie (narzędzia i techniki): stosowność i adekwatność istniejącej technologii i narzędzi; nowy zestaw umiejętności pracowników.

Do podobnych wniosków doszli L. Cao *et al.*, przeprowadzając analizę studiów przypadków w organizacjach wdrażających metodykę Extreme Programming (XP)²⁸². Badacze wykazali kluczowe trudności, dzieląc je na cztery kategorie opisane w poniższej tabeli:

Tabela 15. Wyzwania i adekwatne do nich praktyki zwinnego zarządzania projektami

Odpowiednie praktyki i podjęte wyzwania			
Kategoria	Odpowiednie praktyki	Praktyki zwinne, które zostały zaadaptowane	Stawianie czoła wyzwaniom poprzez dostosowywanie zwinnych praktyk
Związane z procesem rozwoju	abstrakcja w projektowaniu architektury	prosta konstrukcja	te odpowiednie praktyki dotyczą rozwoju zagadnień procesowych poprzez skupienie się na równoważeniu zasady projektowania, które były znakiem rozpoznawczym zwinności. wprowadzają wymaganą formalizację i planowanie w praktyki zwinności
	refaktoryzacja w przód	refaktoryzacja	
	projektowanie poprzez sformalizowaną umowę	klient na miejscu, tylko tyle ile potrzeba dokumentacji (<i>just enough documentation</i>)	
	minimalna identyfikowalność	klient na miejscu, tylko tyle ile potrzeba dokumentacji (<i>just enough documentation</i>)	
	dokumentacja <i>post hoc</i>	gra w planowanie	
	minimalna dokumentacja	tylko tyle ile potrzeba dokumentacji (<i>just enough documentation</i>)	

²⁸² L. Cao *et al.*, *A framework for adapting agile development methodologies*, "European Journal of Information Systems" 2009, vol. 18, s. 332–343.

cd. tabeli 15

Odpowiednie praktyki i podjęte wyzwania			
Kategoria	Odpowiednie praktyki	Praktyki zwinne, które zostały zaadaptowane	Stawianie czoła wyzwaniom poprzez dostosowywanie zwinnych praktyk
Związane z programistami	parowanie nakładania się	programowanie w parach	te odpowiednie praktyki dotyczą kwestii związanych z programistami poprzez budowanie spójnego i wzmocnionego zespołu, i podkreślając autonomię programistów i współdzielenie przez nich odpowiedzialności
	upodmiotowienie poprzez wspólną wiedzę	wspólna własność i odpowiedzialność	
Związane z klientem	wspólne rozumienie specyfikacji	klient na miejscu	te odpowiednie praktyki dotyczą kwestii związanych z klientami i pozyskania zaangażowanego klienta wymaganego w podejściach winnych
	porozumienie w sprawie jakości	gra w planowanie	
Związane z organizacją/ zarządzaniem	oszacowanie z góry	gra w planowanie	te odpowiednie praktyki dotyczą organizacji/wyzwań związanych z zarządzaniem i dążeniem do osiągnięcia lepszego dopasowania kultury organizacyjnej do zwinnych praktyk
	zrównoważona formalizacja	procesy nieformalne	

Źródło: *ibid.*

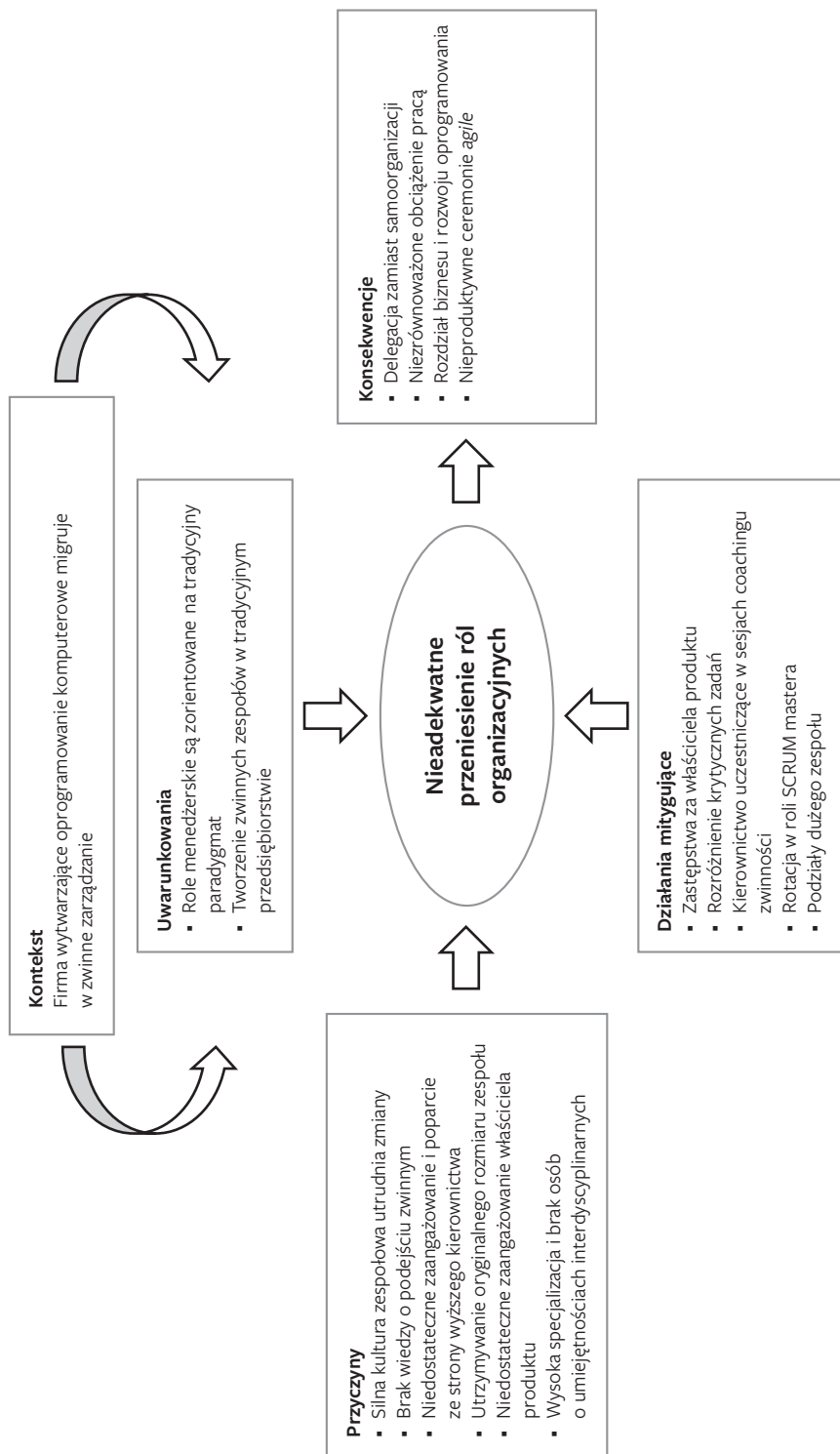
Interesującym wątkiem w dyskusji jest także praca M. Jovanovicia *et al.*, w której zespół badaczy skupił się na aspekcie ludzkim i organizacyjnym transformacji.²⁸³ Postawiona i udowodniona przez badaczy teza dotyczy istotności czynników związanych z nieadekwatną zmianą w zakresie transformacji ról organizacyjnych. Zmianie w zakresie sposobu wykonywania i organizowania pracy powinna ich zdaniem towarzyszyć również zmiana po stronie ról, stanowisk pracy oraz zachowań organizacyjnych. Efekty swoich prac przedstawili na zaprezentowanym poniżej modelu.

Analizując trudności i bariery towarzyszące wdrożeniu zwinnych metod zarządzania i zwinnym transformacjom, najcenniejszym źródłem w tym zakresie wydaje się praca autorstwa K. Dikerta, M. Paasivaary oraz C. Lasseniusa²⁸⁴. Badacze ci przeprowadzili kompleksową i systematyczną analizę 52 publikacji w zakresie studiów 46 przypadków zwinnej transformacji w dużej skali. Wysiłek szczegółowej analizy i podsumowania wniosków rzeszy zajmujących się tą problematyką specjalistów pozwolił im uzyskać całościowy obraz wyzwań zwinności.

²⁸³ M. Jovanović *et al.*, *Transition of organizational roles in Agile transformation process: A grounded theory approach*, "Journal of Systems and Software" 2017, vol. 133, s. 174–194.

²⁸⁴ K. Dikert, M. Paasivaara, C. Lassenius, *Challenges and success factors...*

Rysunek 44. Model nieodpowiedniego przejścia ról organizacyjnych w zwinnej transformacji



Źródło: M. Jovanović et al., *Transition of organizational roles in Agile transformation process: A grounded theory approach*, "Journal of Systems and Software" 2017, vol. 133, s. 174–194.

Po pierwsze, zaobserwowali, iż najbardziej rozpowszechnioną metodyką zwinną stosowaną w transformowanych organizacjach był zdecydowanie SCRUM. W 25 przypadkach była to jedyna wymieniona w artykułach metoda pracy. Drugą najczęściej spotykaną metodyką zwinną był Extreme Programming, samodzielnie występujący w czterech, zaś wraz ze SCRUM w dodatkowych pięciu przypadkach. W kolejnych sześciu artykułach pojawiły się informacje o Lean Software Development. Była to jednak metoda, której zawsze towarzyszył dodatkowo SCRUM. Według K. Dikerta *et al.* innymi przytaczanymi metodami zwinnymi były: Unified Process, Adaptive Development Methodology (ADM) i Rapid Application Development. W jednym przypadku brakowało informacji o stosowanej metodyce²⁸⁵.

W rozpatrywanym kontekście kluczową informacją płynącą z artykułu jest wyczerpująca lista 35 wyzwań towarzyszących zwinnym transformacjom, będąca podsumowaniem ww. systematycznej analizy literaturowej. Lista ta wraz z relatywnymi częstościami występowania została zaprezentowana w tabeli 16.

Tabela 16. Bariery w procesie zwinnej transformacji

1	Opór wobec zmiany (16 przypadków: 38%)	▪ generalny opór wobec zmian ▪ sceptycyzm wobec nowego sposobu pracy ▪ opór wobec odgórnych nakazów i rozporządzeń ▪ kierownictwo niechętnie do zmiany
2	Niedostateczne doinwestowanie transformacji (13 przypadków: 31%)	▪ brak coachingu ▪ brak szkolenia ▪ zbyt duże obciążenie pracą ▪ utrzymanie dotychczasowych zobowiązań ▪ trudności przy reorganizacji przestrzeni biurowych / pomieszczeń
3	Zwinność pozostaje trudna we wdrożeniu (20 przypadków: 48%)	▪ niezrozumienie koncepcji zwinnych ▪ niedostępność poradników i literatury dot. zwinnych praktyk ▪ nowe metody pracy niedostatecznie dopasowane do potrzeb organizacji ▪ nawrót do starego sposobu pracy ▪ przesadny, nadmierny entuzjazm
4	Wyzwania związane z koordynacją w środowisku wielozespołowym (13 przypadków: 31%)	▪ trudne kontakty i współpraca między zespołami ▪ model autonomicznego zespołu będący wyzwaniem dla personelu ▪ globalne wyzwania związane z dystrybucją ▪ osiągnięcie spójności technicznej
5	Pojawiające się różne podejścia w środowisku wielozespołowym (9 przypadków: 21%)	▪ interpretacja agile różni się między zespołami ▪ równoległe stosowanie starego i nowego podejścia
6	Hierarchiczne zarządzanie i granice organizacyjne (14 przypadków: 33%)	▪ rola menedżerów średniego szczebla w agile jest niejasna ▪ zarządzanie w podejściu kaskadowym ▪ utrzymanie starej biurokracji ▪ zachowane silosy wewnętrzne

²⁸⁵ *Ibid.*

7	Wyzwania związane z inżynierią wymagań (16 przypadków: 38%)	▪ niedostateczne adresowanie w podejściu zwinnym wysokopoziomowego zarządzania wymaganiami, ▪ wymagające praktyki rafinacji wymagań ▪ trudne tworzenie i szacowanie historyjek użytkownika ▪ luka między planowaniem długo- i krótkoterminowym
8	Problemy związane z zapewnieniem jakości (6 przypadków: 14%)	▪ uwzględnianie testów нефункциональных ▪ brak testów automatycznych ▪ niejednoznaczność wymagań wpływa na QA
9	Integracja funkcji innych niż produkcyjne (IT) (18 przypadków: 43%)	▪ inne funkcje nie chcą się zmieniać ▪ wyzwania związane z dostosowaniem się do przyrostowego modelu pracy ▪ wyzwania związane z dostosowywaniem działań związanych z wprowadzaniem produktów na rynek ▪ model nagradzania nieskupiający się na pracy zespołowej

Źródło: K. Dikert, M. Paasivaara, C. Lassenius, *Challenges and success factors for large-scale agile transformations: A systematic literature review*, "Journal of Systems and Software" 2016, vol. 119.

Kompleksowa metaanaliza przeprowadzona przez zespół badaczy pozwoliła opracować szeroki zakres czynników. Warto podkreślić, że to, iż trzy najczęstsze bariery to w kolejności: obiektywna trudność we wdrożeniu, integracja innych funkcji niż IT, a dopiero na trzecim miejscu *ex equo* z inżynierią wymagań pojawia się opór wobec zmiany. Obserwacja ta wydaje się nieco kontrować popularną opinię, iż jest to czynnik decydujący dzisiaj o powodzeniu wdrożeń. Wydaje się, że mimo koncentracji na postawach i zachowaniach ludzi nadal konieczne jest rozwijanie technicznych i organizacyjnych aspektów zwinnych wdrożeń i transformacji.

4.3. Modele zwinnej transformacji

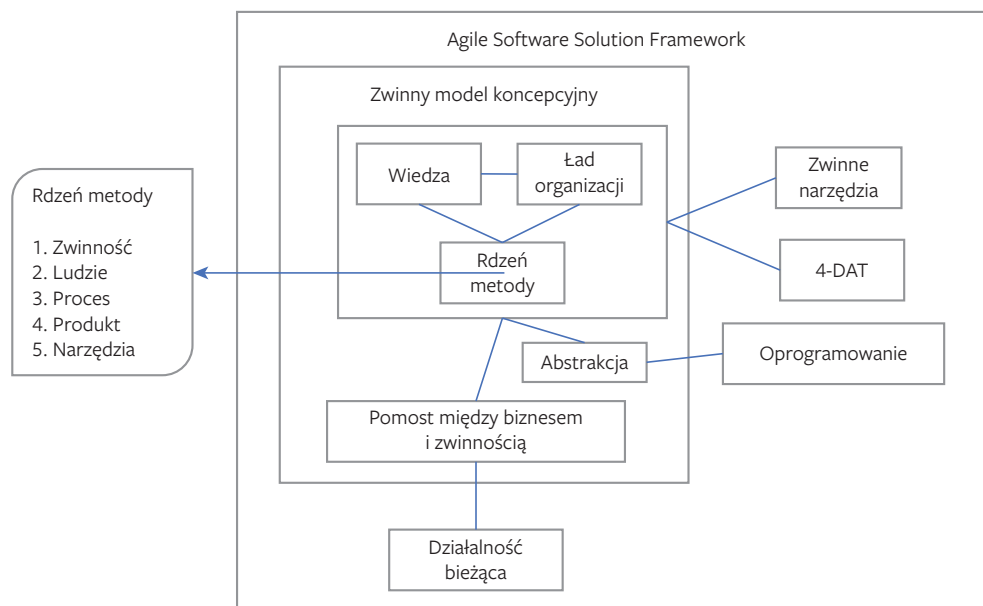
Ze względu na złożoność i wieloaspektowość zwinnych transformacji uzasadnione jest pytanie, czy oprócz mniej lub bardziej wycinkowych doświadczeń dostępne są kompleksowe modele mogące stanowić dla kierownictwa organizacji przewodnik w procesie zmian? Studia źródeł literaturowych pozwalają odnaleźć co najmniej kilka takich rozwiązań, a treść zawartych w nich zaleceń może stanowić cenny kierunkowskaz w tym procesie.

Jednym z takim modeli jest opracowana przez S. Soundararajana i J.D. Arthura rama nazwana OPP – Objective, Principles and Practices (OPP) Framework. Ma ona na celu ocenę wartości (ang. *goodness*) metod zwinnych z perspektywy przedsiębiorstw, adekwatności, potencjału samej organizacji oraz efektywności metody²⁸⁶.

²⁸⁶ S. Soundararajan, J.D. Arthur, *A structured framework for assessing the "goodness" of agile methods*, Conference: Engineering of Computer Based Systems (ECBS), IEEE, 2011.

Dość rozbudowanym rozwiązaniem jest model autorstwa A. Qumera i B. Henderson-Sellersa noszący nazwę Agile Software Solution Framework (ASSF)²⁸⁷. ASSF powstało w celu pomocy menedżerom w ocenie pożądanego poziomu zwinności oraz identyfikacji sposobów wdrożenia podejścia zwinnego w organizacji.

Rysunek 45. Kluczowe komponenty modelu Agile Software Solution Framework (ASSF)



Źródło: A. Qumer, B. Henderson-Sellers, *A framework to support the evaluation, adoption and improvement of agile methods in practice*, "Journal of Systems and Software" 2008, vol. 81, no. 11, s. 1899–1919.

Elementami proponowanego modelu są: **rdzeń metody** (ang. *method core*) reprezentujący sześć aspektów zwinnego wytwarzania oprogramowania; **wiedza** (ang. *knowledge*) związana z tworzeniem, dzieleniem i wykorzystywaniem kompetencji pracowników opracowujących nowe rozwiązania; **ład organizacyjny** (ang. *governance*) – w szczególności rozumiany jako ład IT (ang. *IT governance*) dostarczający zasady, role i odpowiedzialności wraz z modelem tworzenia wartości poprzez strategiczne związanie rozwoju IT z rozwojem biznesu; **narzędzia agile** (ang. *agile toolkit*) – zestaw metod i procesów pracy wykorzystywanych przy realizacji złożonych projektów tworzenia oprogramowania; **4DAT** (ang. *4-dimensional analytical tool*) – opracowana wcześniej przez autorów modelu metoda oceny metod zwinnych z czterech perspektyw: zakresu, charakteru zwinności, wartości zwinnych oraz specyfiki procesu programistycznego.

²⁸⁷ A. Qumer, B. Henderson-Sellers, *A framework to support...*

Modelem nieco mniej nastawionym na transformację, a bardziej na zrozumienie wielowymiarowości zwinności jest ten opracowany przez P. Kettunena i M. Laanti służący w ich intencji do wsparcia projektów zwinnego rozwoju nowych produktów (NPD)²⁸⁸. Autorzy oparli swoje rozwiązanie na relacjach między celami (ang. *goals*), które chcemy osiągnąć dzięki zwinności, środkami (ang. *means*) wdrażanymi w celu jej realizacji oraz katalizatorami (ang. *enablers*) – czynnikami wymaganymi przed zaistnieniem zwinności w organizacji. Wartościowym elementem modelu jest ocena podatności organizacji na uzwinnienie. Rozwiązanie to, zaprezentowane w tabeli 17, umożliwia trzeźwy osąd co do konieczności i wykonalności wdrożenia podejścia zwinnego do projektów opracowywania nowych produktów. Autorzy biorą w tym względzie pod uwagę m.in. model cenowy, modułowość rozwiązania, postawy załogi, możliwość kontroli procesu przez zespół itp.

Tabela 17. Model ewaluacji zwinnego rozwoju nowych produktów

Kluczowy obszar, czynnik	Poziom zwinności		
	mniej		więcej
Reakcja na zmiany rynkowe (nowe funkcje): czy dostosowanie ma miejsce przed czy po wykonaniu produktu?	wczesne zobowiązania (produkcja masowa)	<=>	późne decyzje (dopasowanie)
Wydajność i jakość (rentowność operacyjna): jak obsługiwane są aktualizacje (wbudowanego) oprogramowania i licencjonowanie?	stała cena		wybór oparty na funkcjonalnościach
Nowe innowacje produktowe (konkurencyjny i pożądany produkt): czy produkt jest innowacyjny czy funkcjonalny?	funkcjonalny		innowacyjny
Zwinne metody oprogramowania (z uwzględnieniem zmian): czy zespoły mają kontrolę nad procesem?	jeden standard organizacyjny (jeden rozmiar dla wszystkich)		zespoły projektowe samodzielnie określają swoje procesy
Elastyczna architektura oprogramowania, platformy programowe (elastyczność rozwojowa): czy architektura produktu została zaprojektowana z myślą o elastyczności?	monolityczna		modułowa, poddająca się rekonfiguracji
Wielowykwalifikowani, elastyczni pracownicy: czy ludzie mają odpowiednie nastawienie do pracy w dynamicznym i zmiennym środowisku?	opór przed zmianami		czujący się komfortowo ze zmianą
Doskonałość w technologii (warunki wstępne): czy jesteśmy w stanie zidentyfikować podstawowe trendy, aby przewidzieć przyszłe zmiany (uczenie się)?	reaktywność		innowacyjność (proaktywność)
Elastyczność organizacyjna: Co to jest „wskaźnik polubowności” organizacji?	tylko poprzez hierarchię (co muszę wiedzieć)		swobodny przepływ

Źródło: P. Kettunen, M. Laanti, *Combining agile software projects and large-scale organizational agility*, "Software Process Improvement and Practice" 2008, vol. 13, no. 2, s. 183–193.

²⁸⁸ P. Kettunen, M. Laanti, *Combining agile software projects and large-scale organizational agility*, "Software Process Improvement and Practice" 2008, vol. 13, no. 2, s. 183–193.

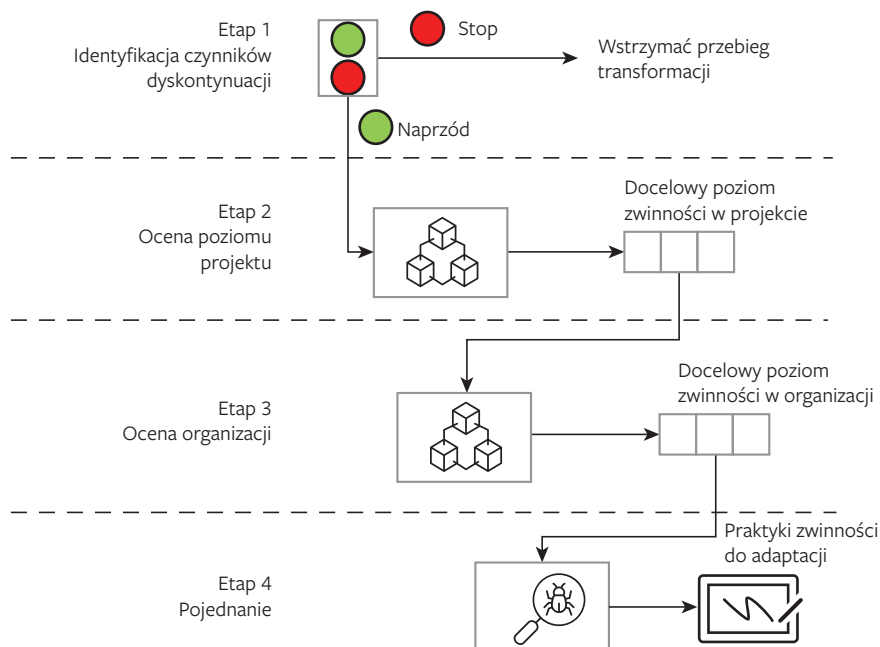
Tabela 18. Poziomy dojrzalości, pryncypia i praktyki podejścia zwinnego wg. A Sidky’ego

	Pryncypia zwinności				
	Zmierz się ze zmianą aby dostarczać wartość klientowi	Często planuj i dostarczaj oprogramowanie	Skupienie na ludziach	Doskonałość techniczna	Współpraca z klientem
Poziom 5. Obejmujący: ustanowienie żywego środowiska podtrzymującego zwinność	ceremonie niskiego procesu	zwinne szacowanie projektu	idealna, zwinna konfiguracja fizyczna	rozwój nakierowany na testy, programowanie w parach, minimalna liczba osób na poziomie – 1 lub 1b w zespole	częsta, bezpośrednia interakcja pomiędzy programistami i klientem (kolokacja)
Poziom 4. Adaptacyjny: odpowiedzi na zmiany poprzez wielopoziomową informację zwrotną	iteracje nakierowane na klienta, ciągła informacja zwrotna dotycząca satysfakcji klienta	mniej więcej i częstsze wydania (4-8 tygodni), adaptacyjne planowanie klienta		codzienne spotkania śledzenia postępów, zwinna dokumentacja, historie użytkownika	natychmiastowa dostępność klienta, umowa z klientem uwzględniająca deklarację współpracy i zaangażowania
Poziom 3. Efektywny: rozwijanie wysoko jakościowego, działającego oprogramowania i efektywny i skuteczny sposób		iteracje nakierowane na ryzyko; planowanie funkcjonalności nie zadań utrzymywanie listy wszystkich funkcjonalności i ich statusu (backlog)	samoorganizujące się zespoły, częsta bezpośrednia komunikacja	ciągła integracja, ciągłe doskonalenie (refaktoryzacja), testy jednostkowe: 30% osób z poziomu 2 i 3 w zespole	
Poziom 2. Ewolucyjny: dostarczanie oprogramowania często i stale	ewoluujące wymagania	ciągła dostawa, planowanie na wielu poziomach		zarządzanie konfiguracją oprogramowania, śledzenie postępów iteracji, brak dużego projektowania z góry (BDUF)	umowa z klientem uwzględniająca rozwój ewolucyjny
Poziom 1. Współpracujący: wzmocnienie komunikacji i współpracy	obserwacja i dopasowywanie procesu	wspólne planowanie	współpracujące zespoły; wzmocnione izmotywowane zespoły	standardy kodowania, narzędzia dzielenia wiedzy, zgłaszanie się do zadań na ochotnika	zobowiązanie klienta do współpracy z zespołami deweloperskimi

Źródło: A. Sidky, J. Arthur, S. Bohner, A. disciplined approach to adopting agile practices: the agile adoption framework, "Innovations in Systems and Software Engineering" 2007, vol. 3, no. 3, s. 203-2016.

Niedostatek rozwiązań w zakresie ustrukturyzowanego wdrażania podejść zwinnych został również dostrzeżony przez zespół badaczy pod kierunkiem A. Sidky'ego²⁸⁹. Stworzony przez nich model opiera się na dwóch kluczowych elementach. Pierwszym jest komponent oceny potencjału zwinności nazwany przez autorów Sidky Agile Measurement Index (SAMI). Drugi to czteroetapowy proces wykorzystujący wskaźnik do wskazania zakresu i konfiguracji praktyk podlegających wdrożeniu w organizacji.

Rysunek 46. Mapa drogowa wdrażania podejścia zwinnego w przedsiębiorstwie wg A. Sidky'ego



Źródło: *ibid.*

Zgodnie z wytycznymi modelu czynnikami brany pod uwagę w SAMI są:

- 1) poziomy zwinności – zestawy praktyk, których wdrożenie w organizacji przyczynia się do doskonalenia procesów tworzenia oprogramowania,
- 2) zwinne pryncypia – wytyczne konieczne do przestrzegania, jeśli zależy nam na zwinnej realizacji procesu,
- 3) zwinne praktyki i koncepcje – specyficzne działania oraz techniki wykorzystywane do opracowywania projektów programistycznych w sposób spójny ze zwinnymi pryncypiami,

²⁸⁹ A. Sidky, J. Arthur, S. Bohner, *A disciplined approach to adopting agile practices...*

- 4) wskaźniki – stosowane przez oceniającego pytania pozwalające na ocenę cech organizacji lub projektu z perspektywy gotowości do zaadaptowania podejścia zwinnego. Szczegółową prezentację modelu przedstawiono w tabeli 18 i na rysunku 46.

Wartym podkreślenia elementem opracowanego przez A. Sidky’ego modelu jest decyzja podejmowana w pierwszym etapie transformacji. Zgodnie z rekomendacją należy w tym momencie rozpatrzyć celowość i wykonalność planowanych zmian. Sidky nazywa ten etap identyfikacją „czynników dyskontynuacji” (ang. *discontinuing factors*). Autor modelu wskazał trzy kluczowe warunki niepowodzenia adaptacji *agile*:

- 1) niewłaściwą potrzebę zwinności: odnosi się to do sytuacji, w których z punktu widzenia biznesu lub rozwoju oprogramowania przyjęcie zwinności nie dodaje żadnej wartości,
- 2) brak wystarczających funduszy: w przypadku niedostępności lub niedostatku funduszy koniecznych do wsparcia działań związanych ze zwinną transformacją proces ten powinien być wstrzymany,
- 3) brak wsparcia najwyższego kierownictwa: w sytuacji braku wsparcia zaangażowanych sponsorów na poziomie najwyższego kierownictwa prawdopodobieństwo sukcesu jest niewielkie.

Jeśli pojawi się któryś z powyższych czynników, należy wstrzymać realizację zmiany lub odłożyć ją do czasu ich ustąpienia.

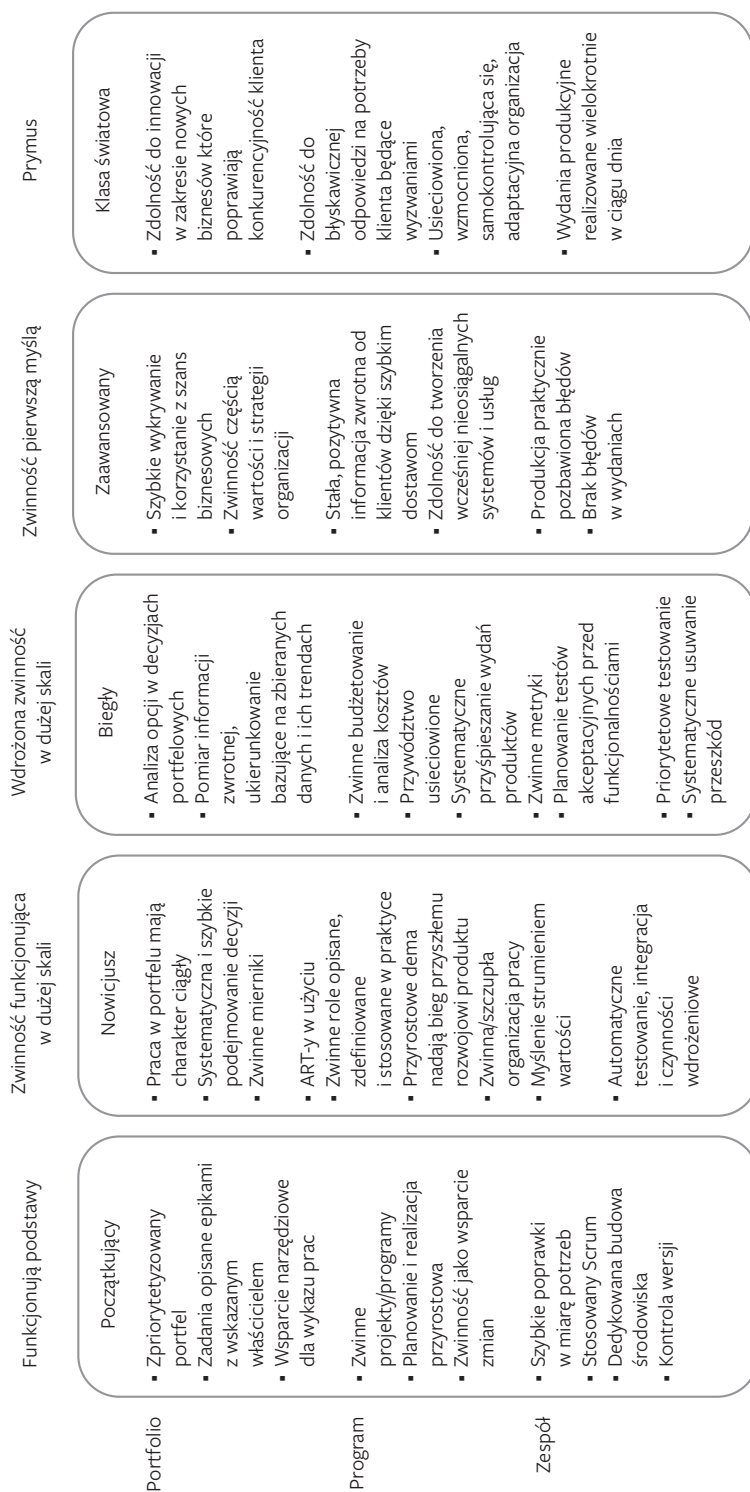
Zbliżoną do A. Sidky’ego koncepcję pomiaru zwinności dużych organizacji i śledzenia postępów ich transformacji biznesowych zaproponowała M. Laanti.²⁹⁰ Oparła ona swój model na pięciu poziomach, korespondujących ze stosowanymi szeroko w biznesie poziomami dojrzałości. Są to według niej pracownicy²⁹¹:

- 1) początkujący – posiadający bazowe praktyki, metody i umiejętności; rezygnujący z tradycyjnego modelu rozwoju na rzecz usprawnienia przepływu pracy,
- 2) nowicjusze – posiadający pełne zrozumienie korzyści z posługiwania się podejściem zwinnym i wykorzystujący skalowane rozwiązania *agile* w budowie swoich produktów,
- 3) biegli – w pełni korzystający z rozwiązań skalowania *agile* także na poziomie kultury, wartości i strategii organizacji,
- 4) zaawansowani – traktujący zwinne zarządzanie jako pierwszy wybór i domyślny sposób wykonywania pracy,
- 5) klasa światowa – liderzy rozwiązań stanowiący wzór do naśladowania dla innych, budujący własne, indywidualne podejścia zarządcze.

²⁹⁰ M. Laanti, *Agile transformation model for large software development organizations*, w: *Proceedings of the XP2017 Scientific Workshops (XP '17)*, Roberto Tonelli (ed.), Association for Computing Machinery, New York 2017, s. 1–5, DOI: 10.1145/3120459.3120479.

²⁹¹ *Ibid.*

Rysunek 47. Agile Transformation Model



Źródło: M. Laanti, *Agile transformation model for large software development organizations*, w: *Proceedings of the XP2017 Scientific Workshops (XP'17)*, Roberto Tonelli (ed.), Association for Computing Machinery, New York 2017, s. 1–5, DOI: 10.1145/3120459.3120479.

Układ modelu uwzględnia trzy poziomy zarządzania organizacją: poziom zespołu (produkcja i praktyki wytwórcze), poziom programu (organizacja i zarządzanie zwinnym przedsięwzięciem) oraz poziom portfela (zarządzanie strategiczne, decyzje inwestycyjne).

Szczegółową konstrukcję modelu zaprezentowano na rysunku 47.

Z informacji dostarczonych przez autorkę modelu, a zawartych w prezentującym go artykule, był on wykorzystany w praktyce przez kilka organizacji, w tym duże przedsiębiorstwo z branży bankowej i ubezpieczeniowej zatrudniające 12 tys. osób. Model zastosowany został równolegle do transformacji zwinnej realizowanej wg modelu SAFe. Poprzez wywiady ze 117 pracownikami udało się uzyskać lepsze zrozumienie stanu wyjściowego, stojących przed nimi wyzwań oraz wzrost zdolności do planowania kolejnych działań doskonalenia zwinności w organizacji. Zgodnie z deklaracjami autorki schemat ten ma być w najbliższym czasie rozwijane o kolejne funkcjonalności.

Jednym z bardziej interesujących rozwiązań prowadzących organizację przez proces zwinnej transformacji jest podręcznik i model opracowany przez brytyjską firmę Axelos. Axelos znany jest jako właściciel odpowiedzialny za rozwój metodyki PRINCE2 oraz innych rozwiązań (MSP, MoR, MoP, ITIL etc.) należących do grona najlepszych praktyk zarządzania w brytyjskiej administracji publicznej i biznesie. W uzupełnieniu do cieszącej się znaczną popularnością uniwersalnej, tradycyjnej metodyki zarządzania PRINCE2²⁹² oraz jej rozwinięcia w stronę podejścia zwinnego, tj. wydanego w 2015 r. podręcznika PRINCE2Agile²⁹³ Axelos opublikował w 2018 r. podręcznik pod nazwą *A Guide to AgileSHIFT*²⁹⁴. W zamierzeniu autorów podręcznik ma wspierać organizacje we wdrażaniu koniecznych zmian w kierunku uzyskania zwinności, prezentować korzyści z nowego kształtu kultury organizacyjnej, tworzyć całościowy ogłód na sposoby zwinnej pracy oraz dostarczać modelu adaptacji zwinności w organizacjach i zespołach. Całościowo model AgileSHIFT przedstawiony został na poniższym schemacie.

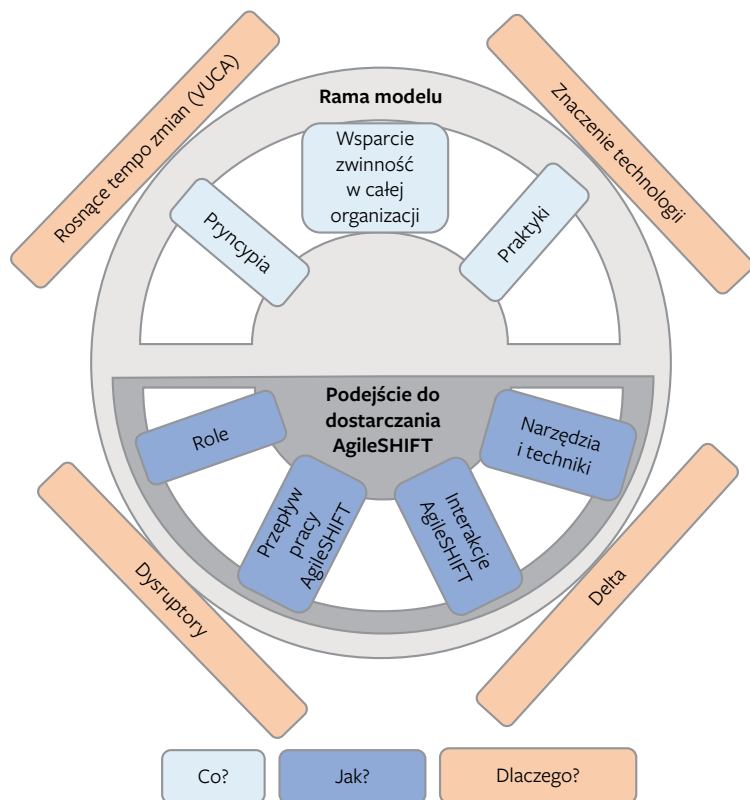
W pierwszej kolejności odnajdziemy w nim omówienie powodów i uzasadnienia dla przeprowadzenia zmian – *Why?* Autorzy wskazują przy tym powody związane z dynamiką otoczenia (VUCA), wzrostu roli technologii, pojawiania się przełomowych innowacji (ang. *disruptive innovation*) oraz tzw. delty, czyli koniecznością zmniejszania dystansu między aktualnym stanem organizacji a stanem pożądanym, docelowym. Axelos wskazuje również na wielowątkowy aspekt transformacji dotyczącej wszystkich obszarów biznesu. Wskazując, iż mogą one być realizowane w różnej formule, rekomenduje zastosowanie zwinnego podejścia do zarządzania dla skutecznej realizacji rozległych i złożonych transformacji biznesowych²⁹⁵.

²⁹² Axelos Ltd., *Managing Successful Projects with PRINCE2*, The Stationery Office, London 2018.

²⁹³ *Id.*, *PRINCE2 Agile...*

²⁹⁴ *Id.*, *A Guide to AgileSHIFT*, The Stationery Office, London 2018.

²⁹⁵ *Ibid.*, s. 20.

Rysunek 48. Model Axelos AgileSHIFT

Źródło: Axelos Ltd., *A Guide to AgileSHIFT*, The Stationery Office, Londyn 2018.

AgileSHIFT oparty jest na następujących elementach:

- 1) Czynniki wspierające zwinność w organizacji – model kładzie nacisk na współpracę wszystkich wewnątrz- i zewnątrzorganizacyjnych interesariuszy, działów i szczebli zarządzania w celu zbudowania strategicznego ukierunkowania działań na rzecz powodzenia transformacji.
- 2) Principia – podobnie jak w przypadku innych metodyk zarządzania opracowanych przez Axelos, tak i tutaj wprowadzono zasady/reguły nadające kierunek zmianom. Są to: a) kwestionuj *status quo*, b) rozwijaj środowisko, gdzie każdy wnosi wartość, c) skupiaj się na współtworzeniu wartości dla klienta, d) dostosuj swoje podejście, e) zmiana jest nieunikniona, przygotuj się na nią.
- 3) Praktyki – w AgileSHIFT są odpowiednikami „tematów” z PRINCE. Są to „aspekty zwinnej pracy, które powinny być stale brane pod uwagę”²⁹⁶. Podobnie jak w przypadku

²⁹⁶ Ibid., s. 44.

pryncypiów jest ich również pięć: a) angażuj interesariuszy, b) buduj współpracujące zespoły, c) planuj, aby być elastycznym, d) dostarczaj iteracyjnie i przyrostowo, e) mierz wartość.

- 4) Role – zgodnie z wytycznymi model przewiduje trzy role w zwinnym środowisku organizacji. Są to: zespół AgileSHIFT, trener AgileSHIFT oraz sponsor AgileSHIFT. Ich charakterystyka i opis zawarty w podręczniku jest zgodna z typowymi rolami w innych rozwiązaniach zwinnych (np. *development team*, *scrummaster*, *product owner* itp.).
- 5) Proces AgileSHIFT – znajdziemy w nim wskazówki dotyczące układu całego cyklu życia projektu opartego na pracy iteracyjnej z wyróżnieniem dwóch dodatkowych faz: 1) *start-up* – poświęconej organizacji przedsięwzięcia i jego zaplanowaniu oraz 2) *close-out*, która ma na celu w sposób zorganizowany zamknąć i rozliczyć projekt. Sam model pracy zbliżony jest do innych rozwiązań w tym zakresie.
- 6) Iteracje – szczegółowy opis działań podejmowanych przez role AgileSHIFT pokazuje znaczne inspirowanie się autorów rozwiązania praktykami pracy w metodyce SCRUM. Proponowane ceremonie charakteryzują się identycznym układem i przebiegiem jak w oryginale.
- 7) Narzędzia i techniki – w tym zakresie model dostarcza zwinnym zespołom zestawu wybranych, nielicznych narzędzi pomocnych w pracy zwinnej. Są to: a) epiki i historie użytkownika, b) szacowanie względne z wykorzystaniem *story points* c) lista zadań, będąca odpowiednikiem product backloga ze SCRUM, d) mapa drogowa, e) metoda pracy roju, gdy cały zespół skupia się nad jedną historyjką, aż ją skończy i zajmie się kolejną, f) tablica kanbanowa, g) tablice canvas, h) agendy pozwalające lepiej przygotować się do odbywających się w procesie ceremonii.

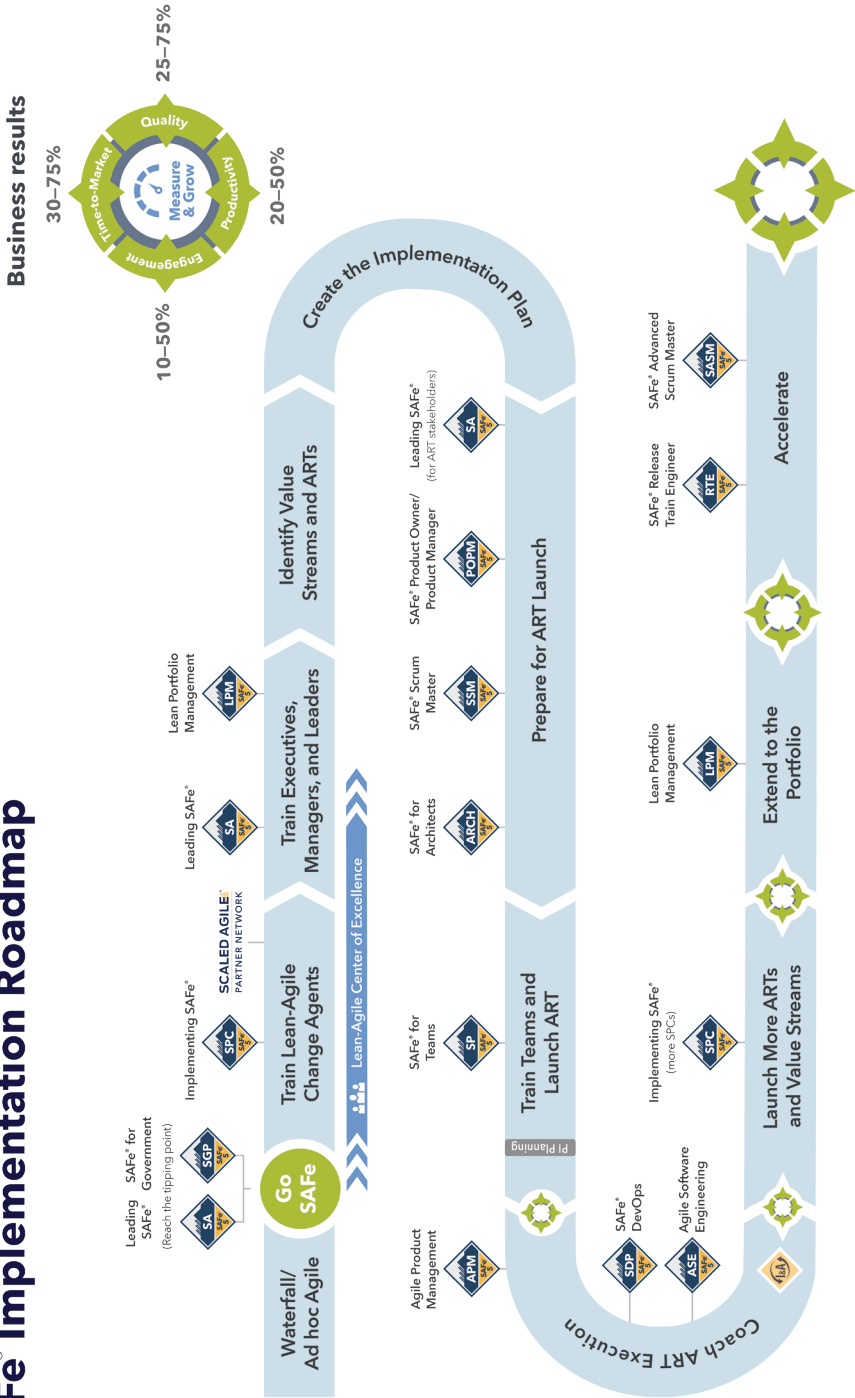
Omawiane wcześniej modele miały charakter naukowy i powstawały na bazie prowadzonych studiów i badań empirycznych. W przeciwieństwie do nich model AgileSHIFT jest rozwiązaniem o charakterze eksperckim. Pozycja Axelos w świecie zarządzania projektami jest znacząca, zaś opracowywane przez firmę podręczniki stanowią zbiór najlepszych praktyk rozpowszechnionych globalnie. AgileSHIFT i Axelos mają zatem potencjał do zaznaczenia swojej obecności w środowisku *agile*. Rozwiązanie to zdaje się jednak zbyt mocno koncentrować na pojedynczych projektach i powielać w tym zakresie zalecenia metodyk o już ugruntowanej pozycji. Oczekiwać można by zatem, że kolejne wersje, o ile zostaną wydane, dostarczą realnej mapy drogowej dla zwinnych transformacji.

To w czym zdaje się niedomagać AgileSHIFT, badacz zagadnień transformacji odnajdzie w najbardziej rozpowszechnionym modelu skalowania zwinności czyli modelu Scaled Agile Framework (SAFe)²⁹⁷.

²⁹⁷ SAFe jest zastrzeżonym znakiem handlowym Scaled Agile, Inc., <https://www.scaledagileframework.com/implementation-roadmap/> (dostęp: 8.06.2021).

Rysunek 49. Mapa drogowa zwinnej transformacji według modelu Scaled Agile Framework (SAFe)

SAFe® Implementation Roadmap



Źródło: <https://www.scaledagileframework.com/implementation-roadmap/> (dostęp: 24.04.2020).

Model ten (nazywany przez polskich specjalistów w skrócie jako „framework”) oferuje nie tylko szeroki zakres struktur, praktyk, metod i technik przeniesienia zwinności z poziomu zespołów na poziom projektów, programów portfeli i przedsiębiorstwa jako takiego. Jednym z elementów SAFe jest także mapa drogowa wdrożenia modelu do organizacji prezentująca główne etapy działania oraz kamienie milowe procesu zwinnej transformacji. Mapę tą przedstawiono na rysunku 49. Według oświadczenia autorów modelu umieszczonego na portalu SAFe została ona opracowana na podstawie doświadczeń „setek największych, światowych przedsiębiorstw, które przeszły już tą drogę”²⁹⁸.

Za punkt startowy obiera ona sytuację zarządzania projektami w sposób tradycyjny lub sporadyczny/nieuporządkowany (*ad hoc*). Decyzja o rozpoczęciu transformacji wyzwala działania związane z ustanowieniem „centrum doskonałości” (ang. *lean-agile center of excellence*) oraz rozpoczęcie szkoleń agentów zmian, zaś w dalszej kolejności poszczególnych szczebli kierowniczych. Dalsze działania to identyfikacja strumieni wartości i „pociągów wydań *agile*” (ang. *agile release trains*, ART), opracowanie szczegółowego planu wdrożenia oraz przygotowanie się do startu ART-ów. W momencie zakończenia przygotowań osoby zaangażowane w przyszłe projekty powinny posiadać już przewidziane w modelu certyfikaty – szczególności dotyczy to architektów, SCRUM masterów, właścicieli i kierowników produktów. W dalszej kolejności szkolenia obejmują zespoły wytwórcze, co konkluduje przygotowania i umożliwia rozpoczęcie planowania pierwszego przyrostu programu (program increment planning) i wystartowanie z pracami developerskimi w ramach ART-ów.

Dalsze etapy mapy polegają na wsparciu udzielanym wszystkim zaangażowanym interesariuszom w celu wzajemnej nauki nowych metod pracy, usprawnienia komunikacji oraz zapewnienia efektywnego przebiegu spotkań i ceremonii przewidzianych w modelu. Główną rolę, zgodnie z zamysłem autorów modelu odgrywają tu tzw. konsultanci programu (ang. *SAFe program consultants*), pełniący funkcję wewnętrznych trenerów, doradców i coachów zwinności. Po koniec tego etapu model przewiduje spotkanie podsumowujące i prezentujące osiągnięte postępy pracy w ramach ART-ów: tzw. przegląd i adaptację (ang. *inspect and adapt*, I&A). Kolejne kroki mapy drogowej to uruchamianie kolejnych przedsięwzięć produktowych, rozszerzanie portfela i pogłębianie zwinnej transformacji, aby nadać jej coraz doskonalszy kształt.

Mimo funkcjonowania wyżej przedstawionych, ale też wielu innych modeli i zaleceń dotyczących zwinnej transformacji, nadal poszukuje się konsensusu i szerzej akceptowanego podejścia do realizacji tego procesu²⁹⁹. Autorzy generalnie zgadzają się co do traktowania go jako istotnej i głębokiej zmiany organizacyjnej w rozwoju oprogramowania³⁰⁰,

²⁹⁸ <https://www.scaledagileframework.com/implementation-roadmap/> (data dostępu: 8.06.2021).

²⁹⁹ M. Jovanović et al., *Transition of organizational roles in Agile transformation process...*

³⁰⁰ M. Laanti, O. Salo, P. Abrahamsson, *Agile methods rapidly...*

jak również co do traktowania tej zmiany jako długotrwałego, wieloletniego procesu, a nie efektu osiąganego jedną decyzją czy deklaracją naczelnego kierownictwa³⁰¹. Co więcej, bardzo silnie akcentuje się aspekt dopasowania i „szycia na miarę” (ang. *tailoring*)³⁰². Każdorazowo zarówno zalecenia metodyk, jak i modeli ich adaptacji powinny zostać zweryfikowane pod kątem specyficznych uwarunkowań działalności konkretnej organizacji³⁰³. Autorzy dopuszczają w tym zakresie nawet daleko idącą adaptację metod wymagającą niejednokrotnie radykalnych zmian w celu dopasowania ich do kontekstu organizacji³⁰⁴. Wyróżnić w tym aspekcie można prace P. Kettunena i M. Laanti³⁰⁵, jak również opracowany przez Axelos na potrzeby metodyki Prince2 Agile tzw. Agilometr³⁰⁶ lub towarzyszący modelowi SAFe wykres radarowy, umożliwiający przeprowadzenie analizy zwinności biznesu (ang. *business agility assessment*). Dostępne są też rozwiązania firm doradczych, np. AgilityHealth Radars autorstwa AgilityHealth³⁰⁷.

4.4. Czynniki sukcesu zwinnej transformacji

Przedsiębiorstwa zainteresowane przeprowadzeniem zwinnej transformacji lub będące w jej trakcie z pewnością zainteresowane będą analizą czynników sukcesu i zbiorem dodatkowych zaleceń oceniających prawdopodobieństwo powodzenia tej zmiany. Z perspektywy badań ankietowych, gromadzących głosy praktyków, należy w tym momencie zacząć od wspomnianego wcześniej raportu firmy VersionOne – tym razem jednak z roku 2016³⁰⁸. Autorzy wskazują w nim pięć zaleceń skutecznego skalowania zwinności:

- 1) spójny proces i praktyki (43%),
- 2) wdrożenie wspólnego narzędzia w zespołach (40%),
- 3) zwinni konsultanci lub trenerzy (40%),
- 4) najwyższe kierownictwo działające jako sponsor (37%),

³⁰¹ A. Qumer, B. Henderson-Sellers, *A framework to support the evaluation...*

³⁰² A. Rohunen et al., *Approaches to agile adoption in large settings: a comparison of the results from a literature analysis and an industrial inventory*, w: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Ronan Nugent (ed.), Springer, Berlin – Heidelberg 2010, s. 77–91; A.S. Campanelli, F.S. Parreiras, *Agile methods tailoring – a systematic literature review*, “Journal of Systems and Software” 2015, vol. 110, s. 85–100, DOI: 10.1016/j.jss.2015.08.035; T. Dybå, T. Dingsøy, *What do we know about agile software development?* “IEEE Software” 2009, vol. 26, no. 5, s. 6–9, DOI: 10.1109/MS.2009.145.

³⁰³ M. Laanti, O. Salo, P. Abrahamsson, *Agile methods rapidly...*; M. Lindvall et al., *Agile software development...*; B. Fitzgerald, G. Hartnett, K. Conboy, *Customising agile methods to software practices at intel Shannon*, “European Journal of Information Systems” 2006, vol. 15, no. 2, s. 200–213.

³⁰⁴ Grenning J., *Launching extreme programming at a process-intensive company*, “IEEE Software” 2001, vol. 18, no. 6, s. 27–33.

³⁰⁵ P. Kettunen, M. Laanti, *Combining agile software projects...*

³⁰⁶ Axelos Ltd., *PRINCE2 Agile...*

³⁰⁷ <https://agilityhealthradar.com/radars/> (dostęp: 8.06.2021).

³⁰⁸ VersionOne, *14th annual...*

5) wewnętrzny zespół wsparcia zwinności (35%).

Kompleksowego zestawu czynników dostarcza nam również przywoływana już praca K. Dikerta *et al.*³⁰⁹ We wspomnianym artykule zespół dokonał nie tylko kompleksowej oceny przedstawionych wcześniej barier, ale również i czynników sukcesu wymienianych w przeanalizowanych studiach przypadków. Ich szczegółową listę przedstawia tabela 19.

Tabela 19. Czynniki sukcesu zwinnych transformacji

1	Wsparcie kierownictwa	▪ zapewnij wsparcie kierownictwa ▪ spraw, aby wsparcie kierownictwa było widoczne ▪ edukuj kierownictwo w zakresie zwinności
2	Zobowiązanie do zmiany	▪ poinformuj, że zmiana nie podlega negocjacjom ▪ okaż silne zaangażowanie
3	Przywództwo	▪ rozpoznaj znaczenie liderów zmian ▪ zaangażuj liderów zmian
4	Wybór i dostosowywanie podejścia zwinnego	▪ ostrożnie dostosuj podejście zwinne, dostosuj się do jednego podejścia ▪ zmapuj stare do nowego sposobu pracy w celu ułatwienia adaptacji ▪ zachowaj prostotę
5	Pilotaż	▪ zacznij od pilota, aby uzyskać akceptację ▪ zbierz doświadczenia i wnioski po pilocie
6	Szkolenia i coaching	▪ zapewnij szkolenie w zakresie metod zwinnych ▪ szkół i wspieraj zespoły, gdy uczą się w zwinnym działaniu
7	Angażowanie ludzi	▪ zacznij od osób już przekonanych do metodyk zwinnych ▪ uwzględnij osoby z wcześniejszym doświadczeniem zwinności ▪ zaangażuj wszystkich w organizację
8	Komunikacja i przejrzystość	▪ intensywnie komunikuj zmianę ▪ spraw, aby zmiana była transparentna ▪ na początku twórz i komunikuj pozytywne doświadczenia
9	Sposób myślenia i wspólne ukierunkowanie	▪ skoncentruj się na zwinnych wartościach ▪ organizuj wydarzenia społecznościowe ▪ pielęgnuj zwinne społeczności ▪ ukierunkuj organizację na zwinność
10	Autonomia zespołu	▪ pozwól zespołom na samoorganizację ▪ pozwól na oddolne wzmocnienie zespołów
11	Zarządzanie wymaganiami	▪ uznaj znaczenie roli właściciela produktu ▪ zainwestuj w naukę, aby udoskonalać wymagania

Źródło: K. Dikert, M. Paasivaara, C. Lassenius, *Challenges and success factors...*

Umieszczonej powyżej listy czynników sukcesu nie należy deprecjonować. Ich znajomość i chęć działania zgodnie z ich kształtem ma kluczowe znaczenie dla powodzenia zmian. Szczególnie istotne jest to, gdy uświadomimy sobie, iż liderzy zwinnych transformacji tacy jak choćby Spotify, Netflix, Google, ING, Allegro, mBank, PZU S.A. pozostają

³⁰⁹ K. Dikert, M. Paasivaara, C. Lassenius, *Challenges and success factors...*

nadal w awangardzie rynku³¹⁰ i że nawet mimo bycia liderem w ING Netherlands, według słów CIO Petera Jacobsa, przeprowadzona transformacja nie przyniosła zamierzonych efektów po stronie biznesu, który nie był w stanie w pełni wykorzystać nowego potencjału³¹¹. Według raportu VersionOne jedynie 5% respondentów przyznało, iż osiągnęli poziom rozwoju, na którym praktyki zwinne przekładają się na większą adaptacyjność³¹².

Jeśli spojrzeć szerzej na rynek, to według Jeffa Sutherlanda, bazującego na danych Standish Group oraz MIT Sloan Business Review, nawet 47% przeprowadzanych zwinnych transformacji kończy się porażką. Co więcej, z tej grupy niemal ⅔ przypadków są to porażki określone jako „definitywne” (ang. *terminal*)³¹³. Wyzwania świata VUCA, galopująca cyfryzacja gospodarek oraz perspektywa nawet szesnastoprocentowego wzrostu EBIDTA na przestrzeni ostatniej dekady organizacji przechodzących na zwinne metody zarządzania (według Accenture)³¹⁴ stanowią istotne zachęty dla przedsiębiorstw, aby mimo wszelkich, spodziewanych trudności podejmować przedsięwzięcia agile.

³¹⁰ M. Golba, P. Ścierski, *Analiza przypadków udanej transformacji zwinnej*, w: *Zwinne zarządzanie projektami w dużych organizacjach*, P. Wyróżewski (red.), Oficyna Wydawnicza SGH, Warszawa 2020.

³¹¹ <https://www.mckinsey.com/industries/financial-services/our-insights/ings-agile-transformation#>; <https://www.plutora.com/blog/agile-devops-failing-fortune-500-companies-wake-call-us> (dostęp: 9.06.2021).

³¹² VersionOne, *14th annual...*

³¹³ J. Sutherland, *Why 47% of Agile Transformations Fail*, <https://www.scruminc.com/why-47-of-agile-transformations-fail/> (dostęp: 9.06.2021).

³¹⁴ Accenture Strategy, *Organizational Analytics Research 2019*, <https://www.accenture.com/nl-en/insights/strategy/move-fast-thrive> (dostęp: 9.06.2021).

ZAKOŃCZENIE

Dwadzieścia lat od Manifestu zwinnego programowania upłynęło bardzo szybko. W nowoczesnych i dynamicznych branżach, do jakich należy IT, to cała epoka. Epoka naznaczona z perspektywy czasu wzrostem popularności i dominacją zwinności w zarządzaniu projektami, a obecnie także w zarządzaniu w ogóle. Na przestrzeni lat postęp pozwalał na przełamywanie kolejnych barier. Zgodnie z mottem ruchu olimpijskiego, sformułowanym przez Pierre’a de Coubertina – *Citius – Altius – Fortius* – w ślad za przyspieszającą gospodarką rosną oczekiwania wobec realizacji projektów: szybciej, wyżej, mocniej! Zwinność jako filozofia, jako podejście w zarządzaniu niezmiennie przysłużyła się realizacji tego hasła. Jak w teorii ograniczeń E. Goldratta zwinność pozwoliła przełamać powolność i ociążałość kaskady. Dzięki wejściu metod *agile* do głównego nurtu przemian rezultaty projektów pojawiają się dzisiaj szybciej, jak nigdy dotąd testując rynek, rozwijając innowacyjne produkty i usługi oraz dostarczając wartości zarówno klientom zewnętrznym, jak i wewnętrznym. Zwinność pozwoliła również wspinać się wyżej, wprowadzając systemy i rozwiązania o coraz większym zakresie, złożoności i skomplikowaniu. Metody skalowania *agile* umożliwiają przełamanie ograniczeń zasobowych. Pozwalają realizować rozległe przedsięwzięcia w strukturach gronowych, zsynchronizowanych zespołach, w SAFe-owych pociągach i większych strukturach zarządczych, nie tracąc przy tym pierwotnego ducha zwinnego programowania. Zwinne transformacje przekształcają organizacje, coraz mocniej i pełniej adaptujące nowe metody działania.

Dorobkiem *agile* w zarządzaniu jest jeszcze jedna, dająca się coraz częściej zaobserwować zmiana. Poszukując możliwości szybkiego dostarczania doskonałych jakościowo produktów, przedsiębiorstwa coraz częściej integrują działalność rozwojową z utrzymaniem. Rozłącznie dotychczas traktowane *run business* i *change business* integrowane są w połączone struktury zarządzane zgodnie z zaleceniami nowych koncepcji, takich jak DevOps³¹⁵

³¹⁵ M. Sánchez-Gordón, R. Colomo-Palacios, *Characterizing DevOps Culture: A Systematic Literature Review*, *International Conference on Software Process Improvement and Capability Determination*, <http://www.rcolomo.com/papers/327.pdf> (dostęp: 24.11.2021); A. Hermawan, L.P. Manik, *The Effect of DevOps Implementation on Teamwork Quality in Software Development*, <http://dx.doi.org/10.20473/jisebi.71.84-90> (dostęp: 24.11.2021).

i Infinite Flow³¹⁶. Według Sauce Labs w 2016 r. 88% badanych firm informatycznych zaadaptowało *agile*. Jednocześnie jednak tylko 21% osiągnęło wszystkich pięć obszarów dojrzałości zwinnego testowania, w tym: 23% natychmiast naprawia błędy, 24% iteruje małe, testowalne wymagania zamiast czekać na ukończenie funkcjonalności, 26% ma więcej testów automatycznych niż ręcznych, 77% zespołów ds. rozwoju i QA komunikuje się w czasie rzeczywistym, 86% deklaruje, iż zespoły ds. rozwoju i QA uważają się za partnerów³¹⁷. Niemal połowa z badanych firm (46%) zadeklarowało, że nadal jest zainteresowana szybszym wdrażaniem tworzonych rozwiązań. Dorobek dwudziestu lat zwinności jest niebywały, lecz wyzwania lat kolejnych mogą przynieść równie niezwykle osiągnięcia.

³¹⁶ J. Johnson, H. Mulder, *Endless Modernization: How Infinite Flow Keeps Software Fresh*, The Standish Group International, Inc., Centreville 2020.

³¹⁷ Sauce Labs, *Testing trends in 2016: a survey of software professionals*, https://saucelabs.com/assets/3vgjd9jz6Th-d4YPqhbOQgx/7d7e5147199a40ad9aeb17f1bd2838dd/Trends_in_Testing_2016_WP.pdf (dostęp: 24.11.2021).

BIBLIOGRAFIA

- Accenture Strategy, *Organizational Analytics Research 2019*, <https://www.accenture.com/nl-en/insights/strategy/move-fast-thrive> (dostęp: 9.06.2021).
- Afonasova M.A. et al., *Digitalization in economy and innovation: The effect on social and economic processes*, „Polish Journal of Management Studies” 2019, t. 19, nr 2.
- Aghina W., De Smet A., Weerda K., *Agility: It rhymes with stability*, “McKinsey Quarterly”, December 2015.
- Agile Business Consortium, *Agile PM. Agile Project Management Handbook v2*. Wydanie polskie, Agile Business Consortium, Ashford 2019.
- Agile Business Consortium, *Zwinne zarządzanie projektami – wersja 2*. Wydanie polskie, Agile Business Consortium, Warszawa 2016.
- Agile Enterprise Concepts and Some Results of Research*, R. Trzcieliński (red.), University of Technology and IEA Press, Poznań 2007.
- Alali E., *7 things wrong with Deloitte's Agile Tube Map*, <https://medium.com/tech-sojourner/7-things-wrong-with-deloittes-agile-tube-map-641192e20068> (dostęp: 17.06.2021).
- Alberts D.S., *The Agility Advantage*, DoD Command and Control Research Program, Washington 2011.
- Alberts D.S., Hayes R.E., *Power to the Edge*, Re Di Roma-Verlag, Washington 2009.
- Appelo J., *Management 3.0: Leading Agile Developers. Developing Agile Leaders*, Addison-Wesley, Boston 2010.
- Appelo J., *Zarządzanie 3.0, Kierowanie Zespołami z wykorzystaniem metodyk Agile*, Wydawnictwo Helion, Gliwice 2016.
- Asproni G., *Wstęp do SCRUM*, https://www.asprotunity.com/resources/articles/AnIntroductionToScrum_SDJ_06-2006_PL.pdf (dostęp: 24.11.2021).
- Atkinson S., Benefield G., *Software development: Why the traditional contract model is not fit for purpose*, w: *Proceedings of the Annual Hawaii International Conference on System Sciences*, Ralph H. Sprague, Jr. (ed.), Wailea 2013, s. 4842–4851.
- Axelos Ltd., *A Guide to AgileSHIFT*, The Stationery Office, London 2018.
- Axelos Ltd., *Managing Successful Projects with PRINCE2*, The Stationery Office, London 2018.
- Axelos Ltd., *PRINCE2 Agile*, The Stationery Office, Londyn 2018.
- Barroca L., Dingsøyr T., Mikalsen M., *Agile Transformation: A Summary and Research Agenda from the First International Workshop*, w: *Agile Processes in Software Engineering and Extreme Programming – Workshops. XP 2019*, R. Hoda (ed.), Lecture Notes in Business Information Processing, vol. 364, Springer, Cham, Montreal 2019, DOI: 10.1007/978-3-030-30126-2_1.
- BCG, *Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries*, https://www.bcg.com/publications/2015/engineered_products_project_business_industry_4_future_productivity_growth_manufacturing_industries (dostęp: 24.11.2021).
- Beck K., *Extreme Programming Explained: Embrace Change*, Addison-Wesley Professional, Boston 1999.

- Beck K., *Wydajne programowanie eXtreme Programming*, MIKOM, Warszawa 2001.
- Belz G., Barbasz A. (red.), *Research Papers. Management Forum 4*, Wydawnictwo Uniwersytetu Ekonomicznego we Wrocławiu, Wrocław 2014.
- Benefield G., *Rolling out Agile at large Enterprise*, w: J. Sutherland, K. Schwaber (eds.), *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*, https://www.academia.edu/43872765/The_Scrum_Papers_Nut_Bolts_and_Origins_of_an_Agile_Framework (dostęp: 20.10.2021).
- Benington H.D., *Production of large computer programs*. In *Proceedings, w: ONR Symposium on Advanced Programming Methods for Digital Computers*, F.D. Rigby (ed.), Washington 1956.
- Bessant J. et al., *Developing the agile enterprise*, "International Journal of Technology Management" 2002, vol. 24, no. 5/6.
- Birrell N.D., Ould M.A., *A practical handbook to software development*, Cambridge University Press, Cambridge 1988.
- Blank S., *Why The Lean Startup Changes Everything*, "Harvard Business Review", May 2013.
- Boehm B., *A spiral model of software development and enhancement*, "ACM SigSoft Software Engineering Notes" 1986, vol. 2, no. 4.
- Boehm B., *Balancing agility and discipline: A guide for the perplexed*, Springer, Berlin–Heidelberg 2004.
- Boehm B., Turner R., *Management challenges to implementing agile processes in traditional development organizations*, "IEEE Software" 2005, vol. 22, no. 5, s. 30–39, DOI: 10.1109/MS.2005.129.
- Brousseau D. et al., *The journey to an agile organization*, "McKinsey Insights", May 2019.
- Campanelli A.S., Parreiras F.S., *Agile methods tailoring – a systematic literature review*, "Journal of Systems and Software" 2015, vol. 110, s. 85–100, DOI: 10.1016/j.jss.2015.08.035.
- Cao L. et al., *A framework for adapting agile development methodologies*, "European Journal of Information Systems" 2009, vol. 18, s. 332–343.
- CHAOS Manifesto 2013. *Think Big, Act Small*, The Standish Group International, Incorporated, <http://versionone.com/assets/img/files/ChaosManifesto2013.pdf> (dostęp: 21.06.2020).
- Christensen C.M., *Przełomowe innowacje*, Wydawnictwa Profesjonalne PWN, Warszawa 2010.
- Cleff A., *Agile is not a noun*, <https://www.andycleff.com/2016/09/agile-is-not-a-noun/> (dostęp: 17.06.2021).
- Cockburn A., *Crystal Clear: A Human-Powered Methodology for Small Teams*, Addison-Wesley, London 2004.
- Conforto E.C. et al., *The agility construct on project management theory*, "International Journal of Project Management" 2016, vol. 34, no. 4, s. 660–674.
- Cottmeyer M., *Agile Transformation*, <https://www.leadingagile.com/whitepaper/> (dostęp: 23.04.2020).
- Cottmeyer M., *The Three Things You Need to Know to Transform Any Sized Organization into an Agile Enterprise*, <https://www.agilealliance.org/resources/sessions/three-things-you-must-know-to-transform-any-sized-organization-into-an-agile-enterprise/> (dostęp: 23.04.2020).
- Dame D., *Agile is a verb, not a noun*, <https://www.linkedin.com/pulse/agile-verb-noun-david-dame;> (dostęp: 17.06.2021).
- DeCarlo D., *eXtreme Project Management. Using Leadership, Principles, and Tools to Deliver Value in the Face of Volatility*, Jossey-Bass, New York 2004.
- Declaration of Interdependence*, <http://pmdoi.org/index.html> (dostęp: 07.02.2020).
- Deemer P. et al., *SCRUM Primer: An Introduction to Agile Project Management with Scrum*, ver. 1.2, <https://www.coursehero.com/file/49551637/scrumprimerpdf/> (dostęp: 10.08.2021).
- Denning S., *The Leader's Guide to Radical Management: Reinventing the Workplace for the 21st Century*, Jossey-Bass, New York 2010.

- Dikert K., Paasivaara M., Lassenius C., *Challenges and success factors for large-scale agile transformations: A systematic literature review*, "Journal of Systems and Software" 2016, vol. 119, s. 87–108.
- Disciplined Agile Delivery (DaD)*, <http://www.disciplinedagiledelivery.com/> (dostęp: 07.02.2020).
- DSDM Consortium, *Agile Project Management Handbook. Version 1.2*, DSDM Consortium 2013.
- DSDM Consortium, *The Agile PMO*, Agile Business Consortium, Kent 2016.
- Dybå T., Dingsøy T., *Empirical studies of agile software development: A systematic review*, "Information and Software Technology" 2008, vol. 50, no. 9–10, s. 833–859.
- Dybå T., Dingsøy T., *What do we know about agile software development?* "IEEE Software" 2009, vol. 26, no. 5, s. 6–9, DOI: 10.1109/MS.2009.145.
- Elssamadisy A., *Has SAFe cracked the large agile adoption nut?*, <http://www.infoq.com/news/2013/08/safe#> (dostęp: 24.11.2021).
- FAST Agile, strona główna, <http://www.fast-agile.com/> (dostęp: 24.11.2021).
- Fitzgerald, B., Hartnett, G., Conboy, K., *Customising agile methods to software practices at intel Shannon*, "European Journal of Information Systems" 2006, vol. 15, no. 2, s. 200–213.
- Flasiński M., *Zarządzanie projektami informatycznymi*, Wydawnictwo Naukowe PWN, Warszawa 2006.
- Forsberg K., Mooz H., *The relationship of system engineering to the project cycle*, <http://www.damiant-gordon.com/Courses/ISE/Papers/The%20Relationship%20of%20System%20Engineering%20to%20the%20Project%20Cycle.pdf> (dostęp: 12.04.2019).
- Frankfort-Nachmias C., Nachmias D., *Metody badawcze w naukach społecznych*, Zys i S-ka, Poznań 2001.
- Freedman R., *Back to the Roots: Where Agile Came From*, <https://dzone.com/articles/back-to-the-roots-where-agile-came-from> (dostęp: 24.11.2021).
- Ganguly A., Nilchiani R., Farr J. V., *Evaluating agility in corporate enterprises*, "International Journal of Production Economics" 2009, vol. 118, no. 2.
- Gilb T., *Software Metrics*, Little, Brown, and Co., New York 1976.
- Gilb T., Gilb K., *EVO. Evolutionary Project Management and Product Development*, <https://www.gilb.com/evo-book-preview-sign-up> (dostęp: 24.11.2021).
- Golba M., Ścierański P., *Analiza przypadków udanej transformacji zwinnej*, w: *Zwinne zarządzanie projektami w dużych organizacjach*, P. Wyróżewski (red.), Oficyna Wydawnicza SGH, Warszawa 2020.
- Goyal S., *Major Seminar On Feature Driven Development. Agile Techniques for Project Management and Software Engineering*, https://www.academia.edu/31885026/Major_Seminar_On_Feature_Driven_Development_Agile_Techniques_for_Project_Management_and_Software_Engineering (dostęp: 24.11.2021).
- Grenning J., *Launching extreme programming at a process-intensive company*, "IEEE Software" 2001, vol. 18, no. 6, s. 27–33.
- Gustavsson, T. *Benefits of agile project management in a non-software development context. A literature review*, w: *Project Management Development – Practice and Perspectives: Fifth International Scientific Conference on Project Management in the Baltic Countries April 14–15*, University of Latvia, Riga 2016, s. 67–82.
- Hermann M., Pentek T., Otto B., *Design Principles for Industrie 4.0 Scenarios: A Literature Review*, https://www.researchgate.net/publication/307864150_Design_Principles_for_Industrie_40_Scenarios_A_Literature_Review (dostęp: 24.11.2021).
- Hermawan A., Manik L. P., *The Effect of DevOps Implementation on Teamwork Quality in Software Development*, <http://dx.doi.org/10.20473/jisebi.7.1.84-90> (dostęp: 24.11.2021).
- Heusser M., *Comparing scaling agile frameworks*, <http://www.cio.com/article/2974436/agile-development/comparing-scaling-agile-frameworks.html> (dostęp: 24.11.2021).

- Heusser M., *Learning to (help others learn to) self-organise*, <https://agile2014.sched.com/event/1eE-82DI/learning-to-help-others-learn-to-self-organise-matthew-heusser> (dostęp: 24.04.2020).
- Heusser M., *Organize your team with an Agile scaling frameworks comparison*, <https://searchsoftwarequality.techtarget.com/photostory/252477744/Organize-your-team-with-an-Agile-scaling-frameworks-comparison/6/Apply-the-SCARE-framework-to-boost-Agile-productivity> (dostęp: 24.11.2021).
- Heusser M., *SaFe vs LeSS vs DaD vs LeadingAgile: Comparing scaling agile frameworks*, <https://www.cio.com/article/2974436/comparing-scaling-agile-frameworks.html> (dostęp: 23.04.2020).
- Heusser M., *Sustainable Cultural Agile Release in the Enterprise (SCARE)*, <https://techbeacon.com/app-dev-testing/whats-missing-large-scale-agile-frameworks-practical-transition-advice> (dostęp: 21.10.2021).
- Heusser M., *What's missing from the large-scale agile frameworks? Practical transition advice*, <https://techbeacon.com/app-dev-testing/whats-missing-large-scale-agile-frameworks-practical-transition-advice> (dostęp: 24.04.2020).
- Highsmith J., *Agile Project Management. Jak tworzyć innowacyjne produkty*, Mikom, Warszawa 2005.
- Highsmith J., *Agile Project Management: Creating Innovative Products*, Pearson Education/Addison-Wesley, New York 2004.
- Highsmith J., *Agile Software Development Ecosystems*, Addison-Wesley Professional, New York 2002.
- Highsmith J., *Messy, exciting, and anxiety-ridden: adaptive software development*, "American Programmer" 1997, vol. 10, no. 1.
- Highsmith J., Cockburn A., *Agile software development: the business of innovation*, "Computer" 2001, vol. 34, no. 9, s. 120–127, DOI: 10.1109/2.947100.
- Highsmith J., *biogram*, <http://www.dorsethouse.com/authors/highsmith.html> (dostęp: 24.11.2021).
- Hormozi, A.M. (2001), *Agile manufacturing: the next logical step*, "Benchmarking: An International Journal" 2001, vol. 8, no. 2, s. 132–143.
- Hundermark P., *Do Better Scrum*, https://moam.info/do-better-scrum-scrum-sense_59c59c471723dde092c9f7da.html (dostęp: 24.11.2021).
- IBM Business Consulting Services, *Companion guide to Your Turn: The Global CEO Study 2004*, <https://www.hospitalitynet.org/file/152001463.pdf> (dostęp: 24.11.2021).
- Jeffries R., *Issues with SAFe*, <https://ronjeffries.com/xprog/articles/issues-with-safe/> (dostęp: 22.04.2020).
- Johnson J., Mulder H., *Endless Modernization: How Infinite Flow Keeps Software Fresh*, The Standish Group International, Inc., Centreville 2020.
- Jovanović M. et al., *Transition of organizational roles in Agile transformation process: A grounded theory approach*, "Journal of Systems and Software" 2017, vol. 133, s. 174–194.
- Kaczor K., *Scrum i nie tylko. Teoria i praktyka w metodach agile*, Wydawnictwo Naukowe PWN, Warszawa 2014.
- Kettunen P., Laanti M., *Combining agile software projects and large-scale organizational agility*, "Software Process Improvement and Practice" 2008, vol. 13, no. 2, s. 183–193.
- Kisielnicki J., *Zarządzanie projektami badawczo-rozwojowymi*, Wolters-Kluwer, Warszawa 2013.
- Komus A. et al., *Studie Status Quo Agile 2014*, https://www.hs-koblenz.de/fileadmin/media/fb_wirtschaftswissenschaften/Forschung_Projekte/Forschungsprojekte/Status_Quo_Agile/Studie_2014/2014.07.23_Bericht_Interessenten_final.v.1.01.pdf (dostęp: 24.11.2021).
- Komus A. et al., *Study Status Quo (Scaled) Agile 2019/20*, <https://www.hs-koblenz.de/en/bpm-labor/status-quo-scaled-agile-2020> (dostęp: 14.06.2021).

- Komus A. et al., *Study Status Quo (Scaled) Agile 2020*, <https://www.hs-koblenz.de/en/bpm-labor/status-quo-scaled-agile-2020> (dostęp: 14.06.2021).
- Komus A., Kuberg M., *Result Report: Status Quo (Scaled) Agile 2019/20*, <https://www.process-and-project.net/studien/studien-anfordern/report-status-quo-scaled-agile-2020-en-request/> (dostęp: 9.06.2021).
- Kosinski T., *Trzy fale zwinności w zarządzaniu*, „Studia i Prace Kolegium Zarządzania i Finansów” 2019, nr 172, s. 143–154, DOI: 10.33119/SIP.2019.172.9.
- Kotarbiński T., *Traktat o dobrej robocie*, Wydawnictwo Uniwersytetu Łódzkiego, Łódź 2019.
- Kruchten P., *The Rational Unified Process: An Introduction*, Addison-Wesley Pearson Education, New York 2003.
- Laanti M., *Agile transformation model for large software development organizations*, w: *Proceedings of the XP2017 Scientific Workshops (XP '17)*, Roberto Tonelli (ed.), Association for Computing Machinery, New York 2017, s. 1–5, DOI: 10.1145/3120459.3120479.
- Laanti M., Salo O., Abrahamsson P., *Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation*, „Information and Software Technology” 2011, vol. 53, no. 3, s. 276–290.
- Larman C., *Agile and Iterative Development – A Manager's Guide*, Addison-Wesley, New Jersey 2003.
- Larman C., Basili V.R., *Iterative and Incremental Development: A Brief History*, „IEEE Computer”, June 2003.
- Larman C., Vodde B., *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*, Addison-Wesley Professional, Boston 2008.
- Leffingwell D., *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*, Addison-Wesley Professional, Boston 2011.
- Leffingwell D., *Scaling Software Agility: Best Practices for Large Enterprises*, Addison-Wesley Professional, Boston 2007.
- Lindvall M. et al., *Agile software development in large organizations*, „Computer” 2004, vol. 37, no. 12, s. 26–34, DOI: 10.1109/MC.2004.231.
- Marinova I., *Doing Agile vs. Being Agile. Why it matters?*, <https://kanbanize.com/blog/doing-agile-vs-being-agile/> (dostęp: 15.06.2021).
- Martin J., *Rapid Application Development*, Macmillan, New York 1991.
- McKinsey & Company, *Leading agile transformation: The new capabilities leaders need to build 21st-century organizations*, <https://www.mckinsey.com/~media/mckinsey/business%20functions/people%20and%20organizational%20performance/our%20insights/leading%20agile%20transformation%20the%20new%20capabilities%20leaders%20need%20to%20build/leading-agile-transformation-the-new-capabilities-leaders-need-to-build-21st-century-organizations.pdf> (dostęp: 24.11.2021).
- Meso P., Jain R., *Agile Software Development: Adaptive Systems Principle and Best Practices*, „Information Systems Management”, Summer 2006, s. 19.
- Midler Ch., „Projectification” of the Firm: The Renault Case, „Scandinavian Journal of Management” 1995, vol. 11, no. 4, s. 363–375.
- Milewski J., *Projektowy młyn*, „Computerworld” 2005, nr 29.
- Mink O.G. et al., *Change at Work: A Comprehensive Management Process for Transforming Organizations*, Jossey-Bass Publishers, San Francisco 1993, za: W.E. Whiteman, *Training and educating army officers for the 21st century: Implications for the United States Military Academy*, Defense Technical Information Center, Fort Belvoir 1998.

- Nerur S., Mahapatra R., Mangalaraj G., *Challenges of migrating to Agile methodologies – Organizations must carefully assess their readiness before treading the path of agility*, "Communications of the ACM" 2005, vol. 48, no. 5, s. 72–78.
- Nieto-Rodriguez A., *The Focused Organizations*, Gower, London 2012.
- Norton D., *Five reasons organizations fail to adopt agile methods*, "Gartner", 9 May 2008.
- O'Connor R., Ie R., Lepmets M., *Exploring the Use of the Cynefin Framework to Inform Software Development Approach Decisions*, w: *The International Conference on Software and System Process (ICSSP)*, Tallinn 2015.
- Palmer S.R., Felsing J.M., *A Practical Guide to Feature-Driven Development*, Prentice Hall, Hoboken 2002.
- Pawlak R., *Transformacja zwinna w dużych organizacjach*, w: *Zwinne zarządzanie projektami w dużych organizacjach*, P. Wyrozębki (red.), Oficyna Wydawnicza SGH, Warszawa 2020.
- Pawlak Z., Trocki M., *Metody organizatorskie*, Wydawnictwo SGPiS, Warszawa 1986.
- Paterek P., Kozarkiewicz A., *Zwinne zarządzanie zespołami projektowymi*, C.H. Beck, Warszawa 2020.
- Pieriegud J., *Cyfryzacja gospodarki i społeczeństwa – wymiar globalny, europejski i krajowy*, w: *Cyfryzacja gospodarki i społeczeństwa – szanse i wyzwania dla sektorów infrastrukturalnych*, J. Gajewski, W. Paprocki, J. Pieriegud (red.), Publikacja Europejskiego Kongresu Finansowego, Instytut Badań nad Gospodarką Rynkową – Gdańska Akademia Bankowa, Gdańsk 2016.
- Pieter J., *Metodologia pracy naukowej*, Wydawnictwo Wyższej Szkoły Pedagogicznej w Katowicach, Katowice 1965.
- PMI, Accenture, *Forging the Future: Evolving With Disruptive Technologies*, Project Management Institute, Thought Leadership Series 2018, <https://www.pmi.org/learning/thought-leadership/series> (dostęp: 20.01.2020).
- PMI, KPMG, *Achieving Greater Agility. The Vital Role of Culture and Commitment*, Project Management Institute, Thought Leadership Series 2017, <https://www.pmi.org/learning/thought-leadership/series> (dostęp: 20.01.2020).
- PMI, *Fact File*, „PMI Today”, October 2014, s. 4.
- PMI, *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, Fifth Edition, Project Management Institute 2013.
- PMI, *Construction Extension to the PMBOK® Guide*, Third Edition, Project Management Institute 2016.
- PMI, *Government Extension to the PMBOK® Guide*, Third Edition, Project Management Institute 2006.
- PMI, *Organizational Project Management Maturity Model (OPM3)*, Third Edition, Project Management Institute 2013.
- PMI, *PMI's Pulse of the Profession 11th Global Project Management Survey: The future of Work. Leading the Way with PMTQ*, <https://www.pmi.org/learning/thought-leadership/pulse> (dostęp: 20.01.2020).
- PMI, *Practice Standard for Earned Value Management*, Second Edition, Project Management Institute 2011.
- PMI, *Practice Standard for Project Configuration Management*, Project Management Institute 2007.
- PMI, *Practice Standard for Project Estimating*, Second Edition, Project Management Institute 2019.
- PMI, *Practice Standard for Project Risk Management*, Project Management Institute 2009.
- PMI, *Practice Standard for Scheduling*, Second Edition, Project Management Institute 2011.
- PMI, *Practice Standard for Work Breakdown Structures*, Second Edition, Project Management Institute 2006.
- PMI, *Project Manager Competency Development Framework*, Second Edition, Project Management Institute 2007.
- PMI, *Requirements Management: A Practice Guide*, Project Management Institute 2016.

- PMI, *Software Extension to the PMBOK® Guide*, Fifth Edition, Project Management Institute 2013.
- PMI, *The Standard for Portfolio Management*, Third Edition, Project Management Institute 2013.
- Podstawy metodologii badań w naukach o zarządzaniu, W. Czakon (red.), Wolters-Kluwer, Warszawa 2011.
- Portman H., *Will the PMO disappear in the scaling agile world?*, Konferencja Portfolio Experience, Warszawa 2017.
- Project Management Institute, *Ahead of the Curve: Forging a Future-Focused Culture. A New Way Forward*, "PMI Pulse of the Profession", February 2020, s. 1–8.
- Project Management Institute, *Capturing the value of project management*, <https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2015.pdf> (dostęp: 24.11.2021).
- Pszczółowski T., *Zasady sprawnego działania*, Wiedza Powszechna, Warszawa 1982.
- PWC, *Industry 4.0: Building the digital enterprise*, <https://www.pwc.com/gx/en/industries/industries-4.0/landing-page/industry-4.0-building-your-digital-enterprise-april-2016.pdf> (dostęp: 24.11.2021).
- Qumer A., Henderson-Sellers B., *A framework to support the evaluation, adoption and improvement of agile methods in practice*, "Journal of Systems and Software" 2008, vol. 81, no. 11.
- Raschke R., David J.S., *Business process agility*, w: *Proceedings of the 11th Americas Conference on Information Systems*, Omaha 2005, s. 355–360.
- Ries E., *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Crown Business, New York 2011.
- Rohunen, A. et al., *Approaches to agile adoption in large settings: a comparison of the results from a literature analysis and an industrial inventory*, w: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Ronan Nugent (ed.), Springer, Berlin–Heidelberg 2010, s. 77–91.
- Roland Berger, *The digital transformation of industry*, Federation of German Industries (BDI) & Roland Berger Strategy Consultants, Berlin 2015.
- Royce W.W., *Managing the development of large software systems: concepts and techniques*. In *Proceedings of the 9th international conference on Software Engineering (ICSE '87)*, IEEE Computer Society Press, Washington 1987, s. 328–338.
- Rudd C., *The Third Wave of Agile*, <https://www.solutionsiq.com/resource/white-paper/the-third-wave-of-agile-2/> (dostęp: 17.06.2021).
- Ruparelia N.B., *Software development lifecycle models*, "ACM SIGSOFT Software Engineering Notes" 2010, vol. 35, no. 3, s. 8–13.
- Sánchez-Gordón M., Colomo-Palacios R., *Characterizing DevOps Culture: A Systematic Literature Review*, *International Conference on Software Process Improvement and Capability Determination*, <http://www.rcolomo.com/papers/327.pdf> (dostęp: 24.11.2021).
- Sauce Labs, *Testing trends in 2016: a survey of software professionals*, https://saucelabs.com/assets/3vgld9jz6Thd4YPqhbOQgx/7d7e5147199a40ad9aeb17f1bd2838dd/Trends_in_Testing_2016_WP.pdf (dostęp: 24.11.2021).
- Saunders M., Lewis P., *Doing research in business and management*, Prentice Hall, Hoboken 2012.
- Scaled Agile Inc., *Achieving Business Agility with SAFe® 5.0*, https://issuu.com/scaledagile/docs/5.0_white_paper_digital (dostęp: 24.11.2021).
- Scaled Agile Inc., *Applying the Scaled Agile Framework in Outsourcing Context: Infogain Case Study*, <http://www.scaledagileframework.com/infogain-case-study> (dostęp: 22.04.2020).
- Scaled Agile Inc., *Epics*, <https://www.scaledagileframework.com/epic/> (dostęp: 22.04.2020).

- Scaled Agile Inc., *Value Streams*, <https://www.scaledagileframework.com/value-streams/> (dostęp: 22.04.2020).
- Schoper Y.-G. et al., *Projectification in Western Economies: A Comparative Study of Germany, Norway and Iceland*, "International Journal of Project Management" 2018, vol. 36, no. 1, s. 71–82.
- Schwaber K., *Agile Software Development with SCRUM*, Microsoft Press, Redmont 2004.
- Schwaber K., *Nexus – przewodnik*, <http://www.agile247.pl/wp-content/uploads/2016/02/Nexus-Guide-v1.1-Polish-nfv3.pdf> (dostęp: 24.04.2020).
- Schwaber, K., *The Enterprise and Scrum*, Microsoft Press 2007.
- Schwaber K., *unSAFE at any speed*, <https://kenschwaber.wordpress.com/2013/08/06/unsafe-at-any-speed/> (dostęp: 24.11.2021).
- Schwaber K., Beedle M., *Agile Software Development with SCRUM*, Prentice Hall, Hoboken 2002.
- Schwaber K., Sutherland J., *The Scrum Guide*, 2020, <https://scrumguides.org/scrum-guide.html> (dostęp: 9.06.2021).
- Scrum at Scale, *The Definitive Guide to Scrum at Scale*, <https://www.scruminc.com/scrum-scale-case-modularity/> (dostęp: 24.11.2021).
- SCRUMStudy, *A guide to the Scrum Body of Knowledge (SBOK Guide)*, SCRUM Study, Phoenix 2013.
- Sharifi H., Zhang Z., *A methodology for achieving agility in manufacturing organisations: An introduction*, "International Journal of Production Economics" 1999, no. 62, s. 7–22.
- Sheffield J., Lemétayer J., *Factors associated with the software development agility of successful projects*, "International Journal of Project Management" 2013, vol. 31, s. 459–472.
- Sidky A., Arthur J., Bohner S., *A disciplined approach to adopting agile practices: the agile adoption framework*, "Innovations in Systems and Software Engineering" 2007, vol. 3, no. 3, s. 203–206.
- Słownik Języka Polskiego PWN, hasło: zwinny, <https://sjp.pwn.pl/sjp/zwinny;2547753.html> (dostęp: 21.06.2021).
- Snowden D.J., Boone M.E., *A Leader's Framework for Decision Making*, <https://hbr.org/2007/11/a-leaders-framework-for-decision-making> (dostęp: 23.04.2020).
- Soundararajan S., Arthur J.D., *A structured framework for assessing the "goodness" of agile methods*, Conference: Engineering of Computer Based Systems (ECBS), IEEE, 2011.
- Standish Group, *CHAOS Report 2015*, https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf. (dostęp: 24.11.2021).
- Standish Group, *The Chaos Report (1994)*, https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf (dostęp: 1.01.2021).
- Stevens Award, *Stevens Lecture on Software Development Methods*, <http://reengineer.org/stevens/previous.htm> (dostęp: 2.03.2020).
- Sutherland J., *A Brief Introduction to SCRUM*, Scrum Alliance, Westminster 2006.
- Sutherland J., *Agile Development: Lessons Learned from the First SCRUM*, <https://www.torak.com/files/Lessons%20Learned%20From%20The%20First%20Scrum%20by%20Dr.%20Jeff%20Sutherland.pdf> (dostęp: 22.10.2021).
- Sutherland J., *Inventing and Reinventing SCRUM in Five Companies*, https://www.researchgate.net/publication/238252809_Inventing_and_Reinventing_SCRUM_in_Five_Companies (dostęp: 24.04.2020).
- Sutherland J., *Why 47% of Agile Transformations Fail. SCRUM Summit 2020*, <https://www.scruminc.com/why-47-of-agile-transformations-fail/> (dostęp: 9.06.2021).
- Sutherland J., Jacobson C., Johnson K., *Scrum and CMMI Level 5: A Magic Potion for Code Warriors*, <https://www.scruminc.com/wp-content/uploads/2014/05/Scrum-and-CMMI-Level-5-A-Magic-Potion-for-Code-Warriors.pdf> (dostęp: 24.11.2021).

- Sutherland J., Schwaber K., *The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process*, <https://www.qagile.pl/wp-content/uploads/2018/11/scrum-papers.pdf> (dostęp: 24.11.2021).
- TeamQuest, *XPrince*, <https://teamquest.pl/baza-wiedzy/project-management/xprince/137.html> (dostęp: 24.11.2021).
- Tekeuchi H., Noaka I., *The New New Product Development Game*, "Harvard Business Review", Jan.–Feb. 1986.
- Thomsett R., *Radical Project Management*, Prentice Hall, Upper Saddle River 2002.
- Trocki M., *Dwadzieścia lat zwinnego zarządzania – doświadczenia i wyzwania*, w: *Zwinne zarządzanie projektami w dużych organizacjach*, P. Wyróżewski (red.), Oficyna Wydawnicza SGH, Warszawa 2020.
- Trocki M., *Podstawy metodyczne zarządzania projektami*, w: *Metodyki zarządzania projektami*, M. Trocki (red.), Bizarre, Warszawa 2011.
- Trocki M., *Podstawy metodycznego wsparcia zarządzania projektami*, w: *Metodyki i standardy zarządzania projektami*, M. Trocki (red.), Polskie Wydawnictwo Ekonomiczne, Warszawa 2017.
- Trocki M., *Projekty w organizacji – problem i wyzwania strategiczne*, w: *Księga jubileuszowa dla Profesor Marii Romanowskiej*, Z. Dworzecki, G. Leśniak-Łebkowska (red.), Oficyna Wydawnicza SGH, Warszawa 2017, s. 195–213.
- Turetken O., Stojanov I., Trienekens J.J.M., *Assessing the adoption level of scaled agile development: a maturity model for Scaled Agile Framework*, "Journal of Software: Evolution and Process" 2017, vol. 29, no. 6, DOI: 10.1002/smr.1796.
- U.S. Army Heritage & Education Center, *Who first originated the term VUCA (Volatility, Uncertainty, Complexity and Ambiguity)?*, <http://usawc.libanswers.com/faq/84869> (dostęp: 31.01.2020).
- VersionOne, *13th Annual State of the Agile Report*, https://www.duxdiligns.com/wp-content/uploads/2019/09/13th-annual-state-of-agile-report_7_May_2019.pdf (dostęp: 24.11.2021).
- VersionOne, *14th Annual State of Agile Report*, <https://www.qagile.pl/wp-content/uploads/2020/06/14th-annual-state-of-agile-report.pdf> (dostęp: 24.11.2021).
- VersionOne, *The 10th Annual State of Agile Report*, <http://www.agile247.pl/wp-content/uploads/2016/04/VersionOne-10th-Annual-State-of-Agile-Report.pdf> (dostęp: 10.04.2017).
- Wald A. et al., *Towards a Measurement of "Projectification": A Study on the Share of Project-Work in the German Economy*, w: *Advanced Project Management: Flexibility and Innovative Capacity*, A. Wald et al. (eds.), Nürnberg 2015, s. 18–36.
- West D. et al., *Agile development: Mainstream adoption has changed agility*, "Forrester Research" 2020, vol. 2, no. 1.
- Wielki Słownik Języka Polskiego, hasło: zwinny, https://wsjp.pl/index.php?id_hasla=39341&id_znaczenia=4230797&l=29&ind=0 (dostęp: 24.11.2021).
- Williams S., *Getting Back to the Roots of Agile*, <https://www.pragmaticinstitute.com/resources/articles/product/getting-back-to-the-roots-of-agile/> (dostęp: 5.01.2021).
- Worrall C., *Agile – it's an adjective, not a noun*, <https://www.bcsconsulting.com/blog/agile-its-an-adjective-not-a-noun/> (dostęp: 17.06.2021).
- Wyróżewski P., *Elastyczne podejście do zarządzania projektami*, w: *Problemy współczesnej praktyki zarządzania*, S. Lachiewicz, M. Matejun (red.), Wydawnictwo Politechniki Łódzkiej, Łódź 2007.
- Wyróżewski P., *Metodyka SCRUM*, w: *Metodyki zarządzania projektami*, M. Trocki (red.), Bizarre, Warszawa 2011.
- Wyróżewski P., *PMO in Focus of Management Studies*, w: *Kommunikation in multikulturellen Projektteams*, S. Gruzca, J. Alnajjar (Hrsg.), Peter Lang Verlag, Frankfurt am Main 2015.
- Wyróżewski P., *Zarządzanie wiedzą projektową*, Difin, Warszawa 2014.

- Wyrozębski P., Zwinne koncepcje i modele planowania przebiegu projektów, w: *Planowanie przebiegu projektu*, M. Trocki, P. Wyrozębski (red.), Oficyna Wydawnicza SGH, Warszawa 2015.
- Wyrozębski P., Zwinne metodyki zarządzania projektami, w: *Metodyki zarządzania projektami*, M. Trocki (red.), Bizarre, Warszawa 2011.
- Wyrozębski P., Cichocki H., Zwinna transformacja w dużej skali, w: *Wybrane aspekty zarządzania procesami, projektami i ryzykiem w przedsiębiorstwach*, E. Sońta-Drączkowska, I. Bednarska-Wnuk (red.), Wydawnictwo Uniwersytetu Łódzkiego, Łódź 2020, s. 149–159.
- Wysocki R.K., *Efektywne zarządzanie projektami. Tradycyjne, zwinne, ekstremalne*, Helion, Gliwice 2013.
- Wysocki R.K., Lewis J.P., DeCarlo D., *The World Class Project Manager: A Professional Development Guide*, Basic Books 2001.
- Zaręba T., *Self aware networks – cechy techniczne i implikacje technologiczne*, w: *Cyfryzacja gospodarki i społeczeństwa – szanse i wyzwania dla sektorów infrastrukturalnych*, J. Gajewski, W. Paprocki, J. Pieriegud (red.), Gdańsk 2016.
- Zieleniewski J., *Organizacja i zarządzanie*, PWN, Warszawa 1969.
- Zwinne zarządzanie projektami w dużych organizacjach, P. Wyrozębski (red.), Oficyna Wydawnicza SGH, Warszawa 2020.
- Žárský J., Sobisek O., Bayer J., *Sustainable Cultural Agile Release in the Enterprise (SCARE)*, https://spicenter.vse.cz/wp-content/uploads/2018/08/semestralni_prace/18_05/Sustainable_Cultural_Agile_Release_in_the_Enterprise.pdf (dostęp: 24.04.2020).

Źródła internetowe

- <http://agilemanifesto.org/> (dostęp: 24.11.2021).
- <http://agilemanifesto.org/iso/pl/manifesto.html> (dostęp: 24.11.2021).
- <http://blogs.agilefaqs.com/2009/01/29/10-evolutionary-project-management-evo-principles/> (dostęp: 24.11.2021).
- <http://less.works/> (dostęp: 24.11.2021).
- <http://scrumalliance.com> (dostęp: 24.11.2021).
- <http://usawc.libanswers.com/faq/84869> (dostęp: 31.01.2020).
- <http://www.agilealliance.org/> (dostęp: 24.11.2021).
- <http://www.agileleadershipnetwork.org/> (dostęp: 17.06.2021).
- <http://www.extremeprogramming.org> (dostęp: 24.11.2021).
- <http://www.gilb.com/> (dostęp: 24.11.2021).
- <http://www.growingagile.co.za/2013/08/slim-an-agile-scaling-pattern/> (dostęp: 24.11.2021).
- <http://www.opengroup.org/togaf/> (dostęp: 24.11.2021).
- <http://www.scaledagileframework.com/> (dostęp: 24.11.2021).
- <http://www.scrumstudy.com/blog/what-is-crystal/> (dostęp: 24.11.2021).
- <https://agilemanifesto.org/> (dostęp: 24.11.2021).
- <https://agilemanifesto.org/history.html> (dostęp: 24.11.2021).
- https://agile-mercurial.com/2019/02/06/agile-frameworks-fact-sheet/#S_Agile (dostęp: 24.04.2020).
- <https://agilityhealthradar.com/radars/> (dostęp: 8.06.2021).
- <https://dictionary.cambridge.org/pl/dictionary/english-polish/agile> (dostęp: 15.06.2021).
- <https://kendis.io/framework-for-scaling-scrum/companies-scaling-scrum-with-nexus/> (dostęp: 24.04.2020).
- <https://pl.wiktionary.org/wiki/agere> (dostęp: 15.06.2021).
- <https://web.archive.org/web/20140207224318/http://www.xprince.net/> (dostęp: 24.11.2021).

- <https://wolski.pro/2008/03/xp-prince2-xprince/> (dostęp: 24.11.2021).
- <https://www.agilest.org/scaled-agile/scrum-of-scrums//> (dostęp: 28.04.2020).
- <https://www.knowledgehut.com/tutorials/scrum-tutorial/scrum-of-scrums> (dostęp: 28.04.2020).
- <https://www.leadingagile.com/> (dostęp: 23.04.2020).
- <https://www.macmillandictionaryblog.com/agile> (dostęp: 15.06.2021).
- <https://www.mckinsey.com/industries/financial-services/our-insights/ings-agile-transformation#> (dostęp: 24.11.2021).
- <https://www.mendix.com/agile-framework/> (dostęp: 24.11.2021).
- <https://www.plutora.com/blog/agile-devops-failing-fortune-500-companies-wake-call-us> (dostęp: 9.06.2021).
- <https://www.scaledagile.com/customer-stories/> (dostęp: 22.04.2020)
- <https://www.scaledagileframework.com/about/> (dostęp: 5.01.2021).
- <https://www.scaledagileframework.com/implementation-roadmap/> (dostęp: 8.06.2021).
- <https://www.scrum.org/Resources/The-Nexus-Guide> (dostęp: 24.11.2021).

