



Studium Magisterskie

Kierunek E-biznes

Imię i nazwisko autora
Miłosz Truszkowski
Nr albumu 131551

Czy kierownik projektu informatycznego powinien być specjalistą do spraw inżynierii oprogramowania?

Praca magisterska
pod kierunkiem naukowym
Dr Hab Bartosza Gruczy
Instytut
Infrastruktury, Transportu i Mobilności

Warszawa 2025

Spis treści

1. Wstęp	5
2. Charakterystyka projektów	7
2.1. Definicja projektu	7
2.2. Żelazny trójkąt projektu	9
2.3. Projektizacja	11
2.4. Rezultaty projektów	13
2.5. Projekty informatyczne w kontekście rosnącej liczby projektów	14
3. Specyfika projektów informatycznych	17
3.1. Niechęć do projektów informatycznych	18
3.2. Cykl życia w projektach informatycznych	19
3.2.1. Cykl życia projektu	19
3.2.2. Cykl życia produktu	20
3.2.3. Integracja PLC i SDLC w projektach informatycznych	21
3.3. Modele SDLC	22
3.3.1. Model kaskadowy	24
3.3.2. Agile	25
3.4. Różnorodne technologie	28
3.5. Specjaliści IT	29
3.6. Ryzyka w projektach IT	29
3.6.1. Czasowy model ryzyk	32
3.7. Ewaluacja projektów informatycznych	34
3.8. Ostatnie trendy w projektach informatycznych	36
3.8.1. Globalizacja	36
3.8.2. Outsourcing	37
3.8.3. Zespoły zdalne	38
4. Kierownik projektu	39
4.1. Kierownik projektu	39
4.2. Role i zadania kierownika projektu	39
4.3. Kompetencje kierownika projektu	42
4.3.1. Komunikacja	44

4.3.2. Przywództwo	45
4.3.3. Zarządzanie sobą (ang. Self-management)	45
4.3.4. Umiejętności zarządcze	46
4.3.5. Umiejętności techniczne	46
4.4. Znaczenie wiedzy z zakresu inżynierii oprogramowania dla kierownika projektu	47
5. Badanie ankietowe wśród pracowników branży IT	51
5.1. Cel badania	51
5.2. Metodologia oraz opis badania	51
5.3. Charakterystyka próby	52
5.4. Wyniki badania	52
5.4.1. Porównanie średnich ocen kompetencji	52
5.4.2. Ocena wymaganego doświadczenia technicznego kierownika projektu	55
5.5. Wpływ wiedzy z zakresu inżynierii oprogramowania na zarządzanie projektem informatycznym	56
5.6. Negatywny wpływ braku wiedzy technicznej na zarządzanie projektem informatycznym	59
5.7. Zagrożenia wynikające z nadmiernego skupienia się kierownika projektu na aspektach technicznych	60
5.8. Wnioski	62
6. Podsumowanie i wnioski końcowe	63
6.1. Złożoność roli kierownika projektu IT	63
6.2. Najważniejsze wyniki badania empirycznego	64
6.3. Odpowiedź na pytanie badawcze	65
6.4. Rekomendacje i znaczenie praktyczne	66
6.5. Zakończenie	66
Bibliografia	67
Spis rysunków	72
Spis tabel	73
Spis załączników	73

1. Wstęp

W świecie, w którym technologia informatyczna przenika niemal każdy aspekt działalności biznesowej, umiejętne zarządzanie projektami IT staje się kluczowe do osiągnięcia sukcesu. Złożoność tych projektów, obejmujących różnorodne technologie, narzędzia i zespoły specjalistów, stawia przed kierownikami projektów wysokie wymagania kompetencyjne. W tym kontekście zasadne jest pytanie, czy wiedza specjalistyczna z dziedziny inżynierii oprogramowania jest niezbędnym elementem w zestawie umiejętności efektywnego kierownika projektu informatycznego.

Literatura z zakresu zarządzania projektami IT wskazuje na wielowymiarowość kompetencji wymaganych od kierowników projektów. Kluczowe znaczenie mają zarówno umiejętności techniczne, jak i kompetencje miękkie. Do umiejętności technicznych zaliczamy między innymi znajomość szerokiego spektrum technologii IT, doświadczenie w pracy z systemami informatycznymi oraz umiejętność zarządzania zespołem technicznym.¹² Natomiast kompetencje miękkie, takie jak komunikacja, negocjacje, budowanie relacji i zarządzanie konfliktami, są niezbędne do efektywnej współpracy z zespołem projektowym, klientami i dostawcami.

Wiedza specjalistyczna z dziedziny inżynierii oprogramowania może być niewątpliwie atutem w pracy kierownika projektu informatycznego. Umożliwia mu ona lepsze zrozumienie technicznych aspektów projektu, trafniejszy wybór odpowiednich narzędzi i technologii, a także efektywniejszą komunikację z zespołem programistów. Hobbs i inni zauważają jednak, że po osiągnięciu pewnego podstawowego poziomu wiedzy, jej dalsze pogłębianie nie przekłada się automatycznie na wzrost kompetencji kierownika projektu.³ Kluczem do sukcesu w zarządzaniu projektami IT może być zatem znalezienie optymalnej równowagi między umiejętnościami technicznymi a miękkimi, dostosowanej do specyfiki danego projektu i kontekstu organizacyjnego.

Należy również wziąć pod uwagę zróżnicowanie typów projektów IT. W zależności od rodzaju projektu, na przykład rozwoju oprogramowania, wdrożenia systemu ERP czy utrzymania infrastruktury IT, znaczenie poszczególnych kompetencji może się różnić. Ba-

¹ N. Haggerty. *Understanding the Link between IT Project Manager Skills and Project Success*. 2000.

² N. Langer, S. Slaughter i T. Mukhopadhyay. *How do Project Managers' Skills Affect Project Success in IT Outsourcing?* 2008.

³ L. Crawford, J. B. Hobbs i J. R. Turner. *Project Categorizations Systems: Aligning Capability with Strategy for Better Results*. 2005.

dania wskazują, że w przypadku projektów o dużej złożoności, kluczowe znaczenie mają kompetencje miękkie, takie jak umiejętność zarządzania zespołem, rozwiązywania konfliktów i budowania relacji. W projektach o mniejszej złożoności większą rolę mogą odgrywać umiejętności techniczne, w tym wiedza programistyczna.⁴⁵

Pytanie o niezbędność wiedzy technicznej dla kierownika projektu IT jest niezwykle istotne, ponieważ ma bezpośredni wpływ na dobór kandydatów na to stanowisko, a co za tym idzie, na efektywność i sukces realizowanych projektów. Do weryfikacji prac poruszających tą tematykę zastosowano metodykę przeglądu zakresu literatury, czyli przegląd literatury, który ma na celu zidentyfikowanie zakresu dostępnych badań na dany temat, a nie syntezę wyników badań.⁶ Przegląd ten wykazał, że w dostępnej literaturze brakuje prac, które odpowiadałyby w sposób bezpośredni na pytanie stawiane w tytule tej pracy. Dostępne źródła skupiają się najczęściej na ogólnych umiejętnościach kierownika projektu, nie zgłębiając zbytnio tematu inżynierii oprogramowania, lub jedynie na pewnej gałęzi projektów informatycznych.⁷ Brak jasnej odpowiedzi na to pytanie generuje wiele dyskusji i kontrowersji w środowisku zarządzania projektami.

Celem niniejszej pracy magisterskiej jest dogłębna analiza argumentów za i przeciw tezie, że kierownik projektu informatycznego powinien posiadać wiedzę z zakresu inżynierii oprogramowania. W pracy zostaną przeanalizowane role i zadania kierownika projektu w kontekście specyfiki projektów IT, ze szczególnym uwzględnieniem unikalnych wyzwań i technologii stosowanych w tej dziedzinie. Zostaną omówione popularne metodyki zarządzania projektami informatycznymi, takie jak tradycyjne metody kaskadowe oraz zwinne metodyki Agile i Scrum, aby ocenić, w jaki sposób wpływają one na rolę i kompetencje kierownika projektu. Przeprowadzone zostanie badanie ankietowe wśród pracowników z branży IT, aby poznać ich opinie na temat niezbędnego wykształcenia i doświadczenia dla kierowników projektów informatycznych. Na podstawie analizy literatury przedmiotu, studiów przypadków oraz wyników badania ankietowego, praca sformułuje wnioski dotyczące znaczenia wiedzy technicznej dla kierownika projektu informatycznego.

⁴ M. Podgórska i M. Pichlak. *Analysis of project managers' leadership competencies Project success relation: what are the competencies of polish project leaders?* 2018.

⁵ J. J. Jiang, G. Klein i S. Margulis. *Important Behavioral Skills for IS Project Managers: The Judgments of Experienced IS Professionals*. 1998.

⁶ M. Ćwiklicki. *Metodyka przeglądu zakresu literatury (scoping review)*. 2020.

⁷ K. Umaji i W. Paireekreng. *A Study of Project Manager Skills for Enterprise Data Platform Project*. 2021.

2. Charakterystyka projektów

W celu lepszego zrozumienia roli kierownika projektu warto najpierw przedstawić i przypomnieć podstawowe informacje na temat projektów oraz zarządzania nimi. Rozdział ten pozwoli na ponowne spojrzenie na projekt w ramach teoretycznych, w oderwaniu od emocji związanych z doświadczeniami prywatnymi, które mogą wpływać na zrozumienie całości tematu. Oprócz podstawowej definicji projektu przytoczone zostaną szczegółowe statystyki dotyczące rosnącego znaczenia projektów oraz ich rezultatów.

2.1. Definicja projektu

Projekt to tymczasowe przedsięwzięcie podejmowane w celu stworzenia unikalnego produktu, usługi lub rezultatu. Tymczasowy charakter projektu oznacza określony początek i koniec. Koniec projektu następuje, gdy jego cele zostaną osiągnięte lub gdy projekt zostaje zakończony, ponieważ jego cele nie mogą zostać zrealizowane lub potrzeba projektu przestaje istnieć. Tymczasowość niekoniecznie oznacza krótki czas trwania. Tymczasowość zazwyczaj nie dotyczy produktu, usługi lub rezultatu stworzonego przez projekt; większość projektów jest podejmowana w celu osiągnięcia trwałego rezultatu.⁸ Turner⁹ zwraca szczególną uwagę w swojej definicji projektu na to, że projekt ma za zadanie dostarczyć unikatową zmianę. Zmiana ta może być rozumiana jako nowy produkt, np. budynek, ale również jako coś bardziej abstrakcyjnego i niematerialnego, np. nowa struktura organizacyjna w firmie. Turner podkreśla również, że zmiana ta powinna przynosić organizacji korzyści. Projekt nie może mieć na celu generowania strat, a w perspektywie długoterminowej powinien wpisywać się w cele strategiczne organizacji. Według Turnera projekt nie jest jedynie wykonywaniem określonej pracy w ramach ustalonego czasu, kosztów i jakości. Ograniczenia te są istotą wszystkich działań, zarówno biznesowych, jak i prywatnych. Projekt definiują również złożoność i powtarzalność wymaganych prac. Charakterystyka projektu w porównaniu z procesami i funkcjami w organizacji została przedstawiona na rysunku 2.1. Projekt to działanie o dużej złożoności i niskiej powtarzalności, a warunki projektu są zawsze wyjątkowe, co wymaga dostosowania działań do konkretnych okoliczności. Kerzner¹⁰ oraz Turner¹¹ za istotę projektu

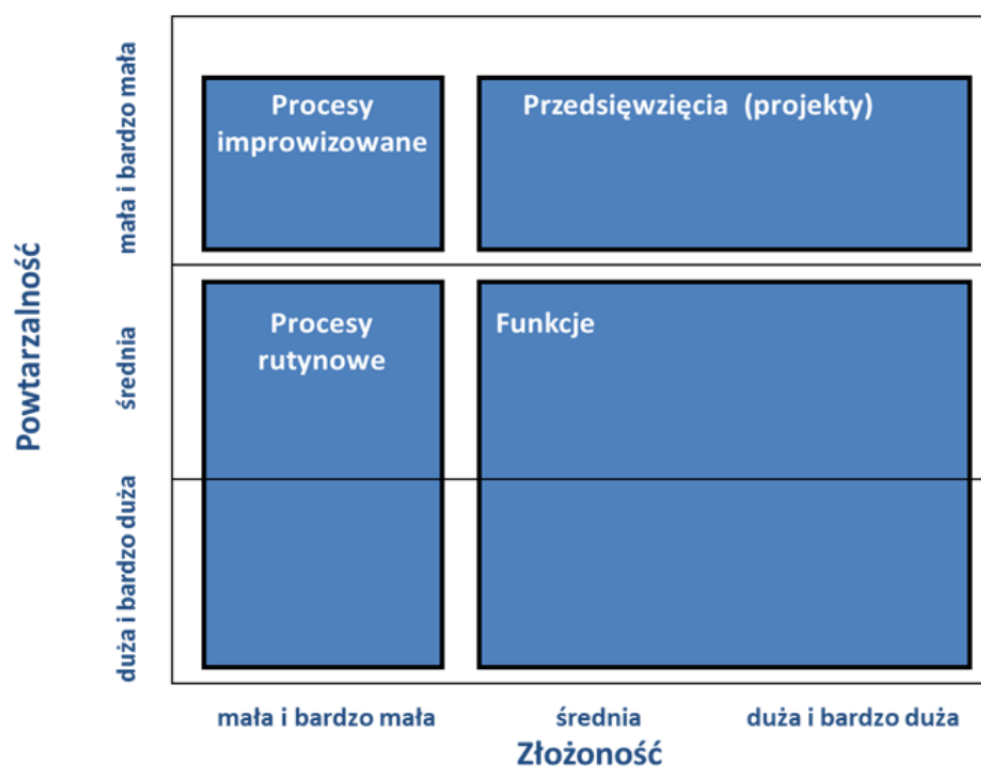
⁸ Project Management Institute. *PMBOK Guide – Fourth Edition*. 2009.

⁹ J. R. Turner. *Gower Handbook of Project Management*. Routledge, 2016.

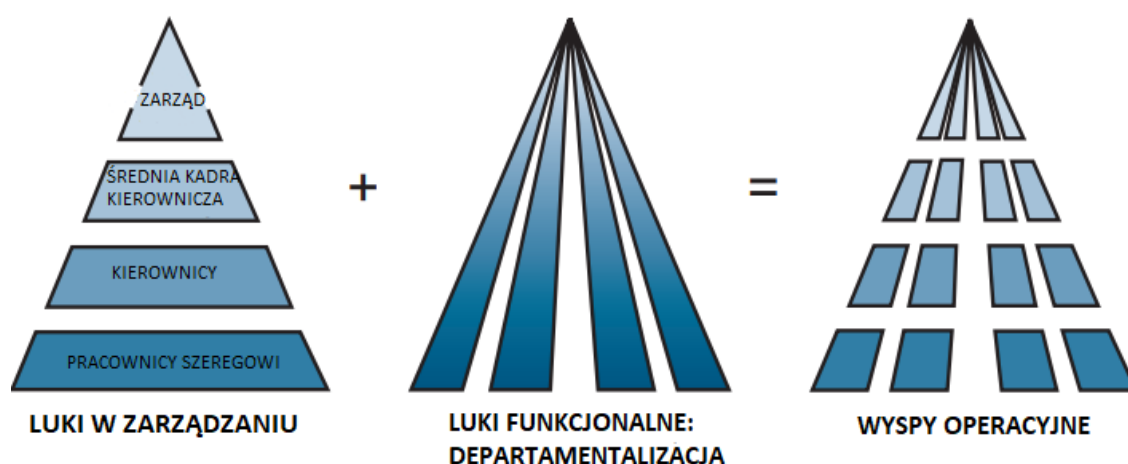
¹⁰ H. Kerzner. *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. John Wiley & Sons, 2017.

¹¹ Turner, *Gower Handbook of Project Management*.

Rysunek 2.1. Porównanie projektów i funkcji



Źródło: M. Trocki, B. Gucza, K. Ogonek, *Zarządzanie projektami*, PWE, Warszawa 2003, s. 14

Rysunek 2.2. Struktura w organizacjach

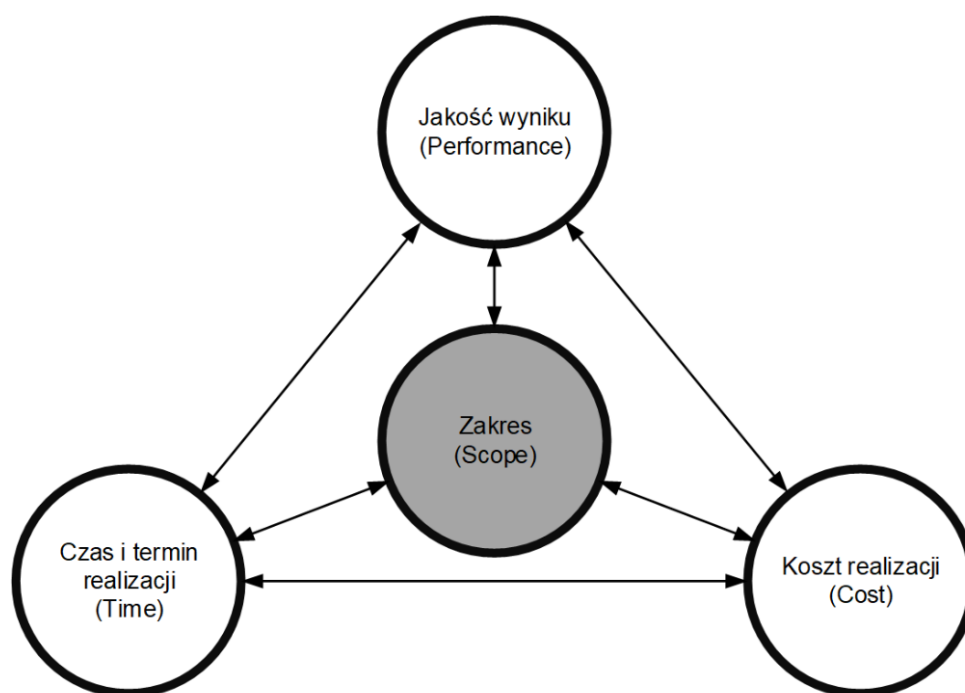
Źródło: H. Kerzner. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. John Wiley & Sons, 2017

uznają również to, że projekt wymaga integracji zarówno materialnych, jak i niematerialnych zasobów organizacji. Kerzner posługuje się grafiką, która została przedstawiona na rysunku 2.2. Przedstawia on zasoby organizacji jako samotne wyspy powstałe w wyniku podziału organizacji na pionowy funkcyjny oraz warstwy zarządcze. Wyspy te w ramach projektu łączone są w tymczasową organizację mającą wspólny cel. Jednak osiągnięcie współpracy między tymi samodzielnymi jednostkami nie jest łatwe, dlatego odpowiednia koordynacja staje się kluczowym elementem projektu. Turner natomiast w kontekście wykorzystania zasobów w projekcie podkreśla, że podział organizacji na mniejsze jednostki, czyli zespoły projektowe, otwiera przed nią nowe możliwości, które wcześniej były nieosiągalne. Autor porównuje tę sytuację do tankowca i floty małych jachtów. Gdy przed obiema jednostkami postawi się zadanie przewożenia ropy, wykorzystanie tankowca będzie bardziej efektywne. Jednak tankowiec nie jest w stanie wykonywać ostrych zwrotów ani przepłynąć przez wszystkie śluzy. Zespoły projektowe pozwalają osiągnąć zwinność małych jachtów, która jest niezbędna w działaniach charakteryzujących się dużą niepewnością, takich jak projekty.

2.2. Żelazny trójkąt projektu

W celu ułatwienia zrozumienia kluczowych ograniczeń projektu opracowano koncepcję żelaznego trójkąta projektu. Ukazuje ona wzajemne zależności między czterema kluczowymi

Rysunek 2.3. Żelazny trójkąt projektu



Źródło: Wykład *Podstawy Zarządzania Projektami*, Mateusz Juchniewicz

ograniczeniami projektu: zakresem, czasem, jakością i kosztem.¹²¹³ Zakres definiuje wszystkie prace niezbędne do stworzenia finalnego produktu, usługi lub rezultatu projektu. Obejmuje on cele, zadania, funkcje i cechy projektu.¹⁴ Czas to okres trwania projektu, od początku do końca. Obejmuje harmonogram zadań, kamienie milowe i ostateczny termin zakończenia projektu. Jakość odnosi się do stopnia, w jakim produkt, usługa lub rezultat projektu spełniają określone wymagania i standardy. Obejmuje zarówno aspekty funkcjonalne, jak i нефunkcjonalne, a także oczekiwania interesariuszy. Koszt to całkowita kwota pieniędzy niezbędna do sfinansowania projektu. Uwzględnia on zasoby, materiały, wynagrodzenia, narzędzia i inne wydatki. Główna idea trójkąta projektu polega na tym, że zmiana jednego z tych ograniczeń nieuchronnie wpływa na co najmniej jedno z pozostałych. Na przykład:

- Zwiększenie zakresu prac projektu zazwyczaj wymaga wydłużenia czasu trwania projektu lub zwiększenia budżetu.

¹² J. Pollack, J. Helm i D. Adler. „What is the Iron Triangle, and how has it changed?” W: *International Journal of Managing Projects in Business* (2018). DOI: 10.1108/IJMPB-09-2017-0107.

¹³ P. Wyrozębski i in. *Project Management - Challenges and Research Results*. 2016. DOI: 10.6084/m9.figshare.4502792.

¹⁴ C. Van Wyngaard, J. H. Pretorius i L. Pretorius. *Theory of the triple constraint — A conceptual review*. 2012. DOI: 10.1109/IEEM.2012.6838095.

- Skrócenie czasu realizacji projektu może wiązać się z koniecznością zwiększenia kosztów (np. poprzez zatrudnienie dodatkowych zasobów) lub obniżenia jakości.
- Podniesienie standardów jakości często wymaga więcej czasu i pieniędzy.

Trójkąt projektu jest jednak jedynie uproszczonym modelem, który nie uwzględnia wszystkich aspektów projektu. Nie uwzględnia on na przykład ryzyka, zasobów ludzkich, zarządzania zakłóceniami czy satysfakcji klienta. Należy o tym pamiętać, interpretując wyniki analizy opartej na tym modelu. Mimo to trójkąt projektu pozostaje cennym narzędziem w zarządzaniu projektami. Pomaga on lepiej zrozumieć podstawowe zależności między zakresem, czasem, jakością i kosztem. Trójkąt projektu wspiera również komunikację tych zależności interesariuszom projektu oraz pozwala na podejmowanie bardziej świadomych decyzji dotyczących kompromisów i alokacji zasobów.

2.3. Projektyzacja

W ostatnich latach rosnące znaczenie projektów w działalności organizacji zaowocowało pojawieniem się zjawiska projektyzacji, które polega na zastępowaniu tradycyjnych, powtarzalnych działań unikatowymi przedsięwzięciami realizowanymi jako projekty.¹⁵ Proces ten wpływa zarówno na struktury organizacyjne, jak i na podejście do zarządzania w sektorach publicznym i prywatnym.¹⁶ Pojęcie projektyzacji zostało wprowadzone już w latach 90. XX wieku. Kluczowe badania przeprowadzone przez Midlera oraz Maylora i współpracowników¹⁷ wskazują, że transformacja organizacyjna polega nie tylko na zwiększeniu liczby realizowanych projektów, ale przede wszystkim na zmianie sposobu, w jaki organizacje planują i wdrażają swoje działania. Proces projektyzacji niesie ze sobą zarówno korzyści, jak i pewne zagrożenia. Z jednej strony umożliwia on zwiększenie elastyczności organizacyjnej i szybką reakcję na zmieniające się warunki rynkowe.¹⁸ Z drugiej strony jednak, ciągła intensyfikacja realizowanych projektów może prowadzić do nadmiernego obciążenia pracowników, wzrostu presji i stresu, a także do fragmentacji działań organizacyjnych.¹⁹

¹⁵ M. Juchniewicz. „Projektyzacja – koncepcja, zakres, konsekwencje”. W: *Studia i Prace Kolegium Zarządzania i Finansów SGH* (2018).

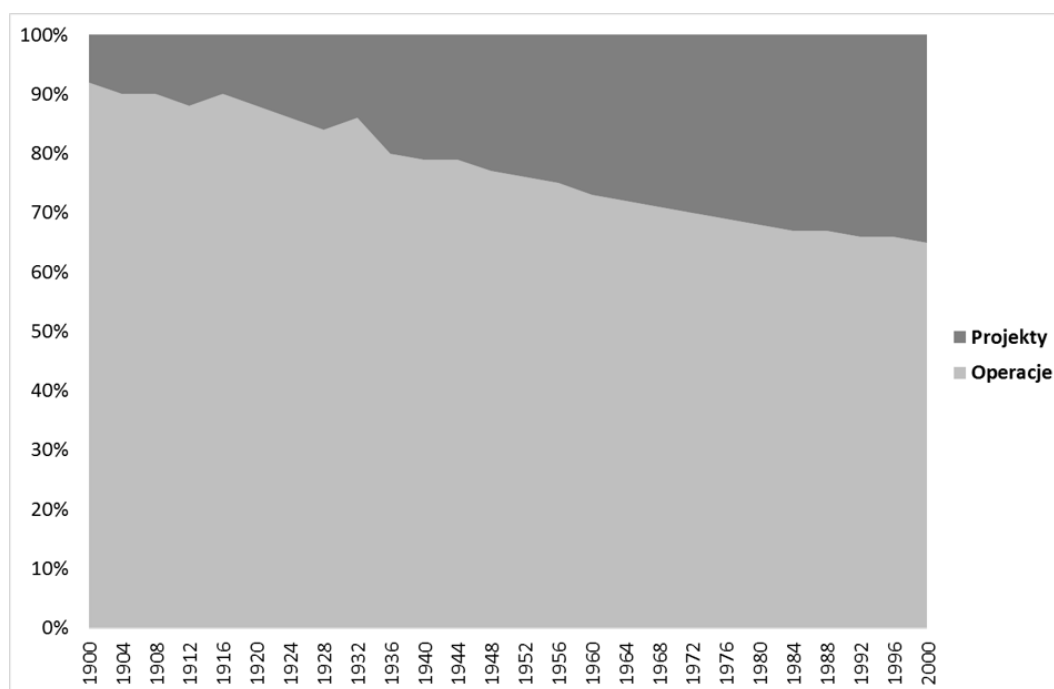
¹⁶ C. Midler. „Projectification of the Firm: The Renault Case”. W: *Scandinavian Journal of Management* (1995).

¹⁷ H. Maylor i in. „From Projectification to Programmification”. W: *International Journal of Project Management* (2006).

¹⁸ G. Praweńska-Skrzypek. *Zarządzanie projektami jako szansa i wyzwanie dla administracji publicznej*. Wydawnictwo SGH, 2011.

¹⁹ B. Jałocha. „Projectification of the European Union and its Implications for Public Labour Market Organisations in Poland”. W: *Journal of Project, Program and Portfolio Management* (2012).

Rysunek 2.4. Stosunek projektów do operacji w organizacjach



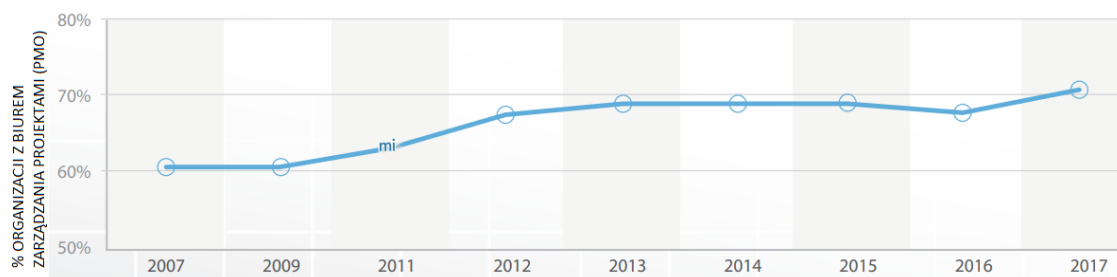
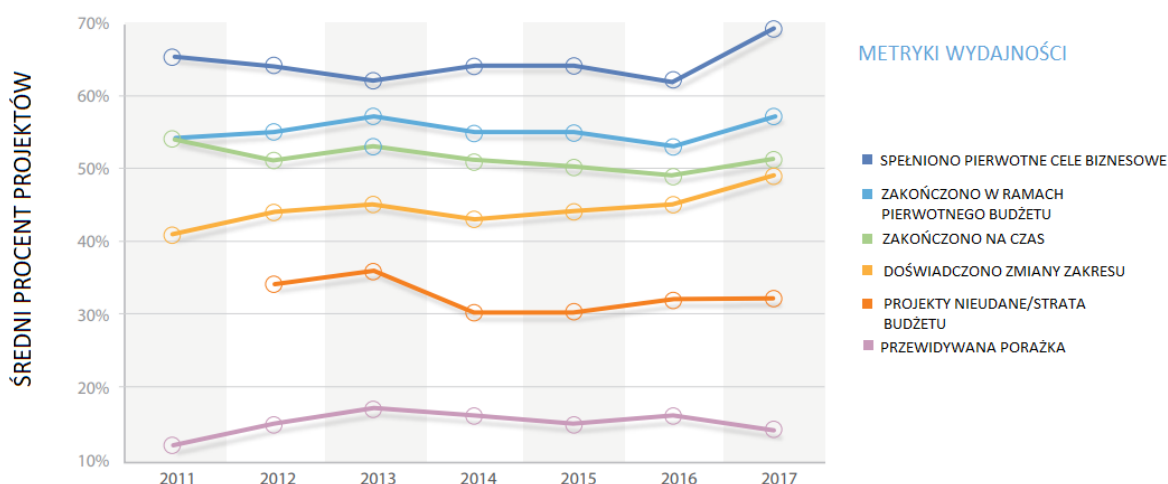
Źródło: Wykład *Podstawy Zarządzania Projektami*, M. Juchniewicz

Aktualne dane statystyczne pokazują, że projekty stanowią coraz większy udział w działalności organizacji. PriceWaterhouseCoopers w swoim raporcie podaje, że globalne wydatki na publiczne projekty inwestycyjne wzrosną z poziomu 4 bln USD w 2012 r. do ok. 9 bln USD w 2025 r.²⁰ Dodatkowo A. Nieto-Rodriguez stwierdza, że udział projektów jako źródła globalnego produktu brutto będzie ciągle rósł. Szacuje się, że do ok. 2025 r. wyniesie on ok. 35%.²¹ Rysunek 2.4 przedstawia szacunkowe dane obrazujące to zjawisko. Potwierdzeniem rosnącego znaczenia projektów są również raporty PMI, w których zauważono znaczący wzrost liczby biur projektów w organizacjach. Rysunek 2.5 przedstawia wspomniane statystyki. Według tego raportu odsetek organizacji posiadających biuro projektowe wzrósł z 61% w 2007 r. do 70% w 2017 r. PMI podał również, że liczba stanowisk związanych z zarządzaniem projektami osiągnęła w 2017 roku prawie 66 milionów, a popyt wciąż rośnie. „Do 2027 roku pracodawcy będą potrzebować 87,7 miliona osób na stanowiskach związanych z zarządzaniem projektami.”²²

²⁰ PWC. *Capital Projects and Infrastructure Spending, Outlook to 2025*. 2014.

²¹ A. Nieto-Rodriguez. *The Focused Organizations*. Gower, 2017.

²² Project Management Institute. *Project Management Job Growth and Talent Gap*. 2017.

Rysunek 2.5. Procent organizacji z biurem zarządzania projektamiŹródło: *Pulse of the Profession*, PMI 2017**Rysunek 2.6.** Rezultaty projektówŹródło: *Pulse of the Profession*, PMI 2017

Kolejnym przykładem projektyzacji może być polski sektor publiczny, w którym projekty w wielu organizacjach stanowią już około 60-80% wszystkich działań.²³

2.4. Rezultaty projektów

Warte wspomnienia są również statystyki dotyczące rezultatów realizowanych projektów. Według raportu PMI jedynie 70% projektów osiąga pierwotnie założone cele, a około 15% jest uznawanych za całkowitą porażkę. Rysunek 2.6 przedstawia te statystyki w sposób bardziej szczegółowy. Raport Arras People jest jeszcze bardziej pesymistyczny – aż 41%

²³ Prawelska-Skrzypek, *Zarządzanie projektami jako szansa i wyzwanie dla administracji publicznej*.

respondentów zgłasza, że projekty, w których uczestniczyli, zakończyły się porażką.²⁴ Takie dane sugerują, że mimo rosnącej liczby realizowanych projektów, wiele z nich nie przynosi oczekiwanych rezultatów, co stanowi istotne wyzwanie dla współczesnych organizacji.

2.5. Projekty informatyczne w kontekście rosnącej liczby projektów

Projekty informatyczne stanowią istotny segment realizowanych przedsięwzięć, zwłaszcza w kontekście cyfryzacji i dynamicznych zmian technologicznych. Według Gartner światowe wydatki na IT wyniosły 3,5 biliona dolarów w 2017 roku, co oznacza wzrost o 2,4% w porównaniu z rokiem 2016.²⁵ Mimo rosnącej liczby projektów informatycznych, wyniki ich realizacji często pozostawiają wiele do życzenia i wypadają one znacznie gorzej na tle innych typów przedsięwzięć. Według raportu Standish Group jedynie 30% projektów informatycznych osiąga pełne cele, natomiast 50% wykazuje problemy operacyjne, takie jak opóźnienia czy przekroczenia budżetowe. Aż 20% analizowanych projektów informatycznych zakończyło się porażką

Tabela 2.1. Rezultaty projektów w latach 2011-2015

	2011	2012	2013	2014	2015
Sukces	29%	27%	31%	28%	29%
Wyzwania	49%	56%	50%	55%	52%
Niepowodzenie	22%	17%	19%	17%	19%

Źródło: *CHAOS Report*, Standish Group 2015

Podobne dane raportuje również World Bank Group. Według ich statystyk jedynie 13% projektów informatycznych z sektora dużych przedsiębiorstw kończy się sukcesem. Dodatkowo przedstawili oni porównanie projektów informatycznych do wszystkich projektów, z którego wynika, że te związane z teleinformatyką wyraźnie odnotowują gorsze rezultaty. Obie te statystyki zostały przedstawione na rysunkach 2.7. i 2.8. Według ankiety przeprowadzonej wśród specjalistów i kadry kierowniczej z branży technologicznej, zarówno w sektorze publicznym, jak i prywatnym, powodem wysokiego odsetka niepowodzeń projektów

²⁴ Arras People. *Project Management Benchmark Report*. 2010.

²⁵ Inc. Gartner. *Gartner Says Worldwide IT Spending Forecast to Grow 2.4 Percent in 2017*. 2014.

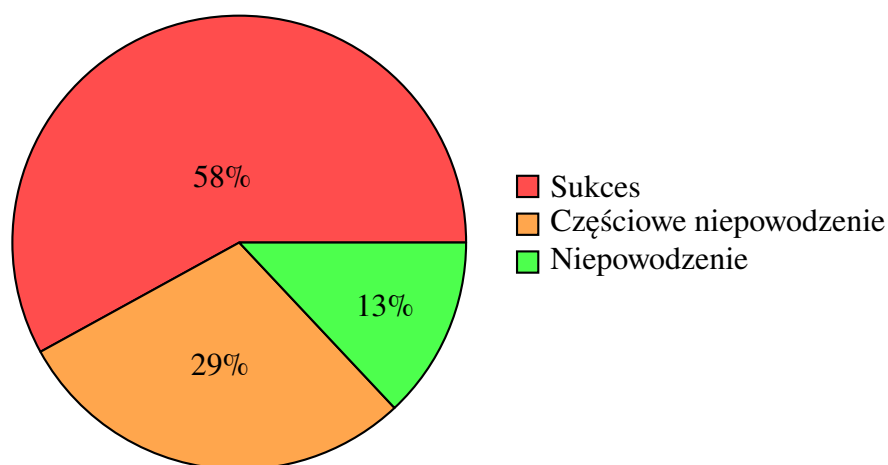
Tabela 2.2. Podsumowanie wyników badania CHAOS

	Duże	Średnie	Małe
Średni koszt rozwoju	\$2,322,000	\$1,331,000	\$434,000
Średnie przekroczenie kosztów	189%	178%	112%
Średnie przekroczenie harmonogramu	222%	220%	239%
Zachowane oryginalne funkcje	42%	65%	74%
Projekty zakończone sukcesem	9.6%	16.2%	28.0%
Projekty z wyzwaniami	61.5%	46.7%	46.7%
Projekty nieudane	29.5%	37.1%	25.3%

Źródło: CHAOS Report, Standish Group 2015

informatycznych jest słabe zaangażowanie użytkowników, brak wsparcia kierownictwa oraz nieprecyzyjne określenie wymagań projektu.²⁶

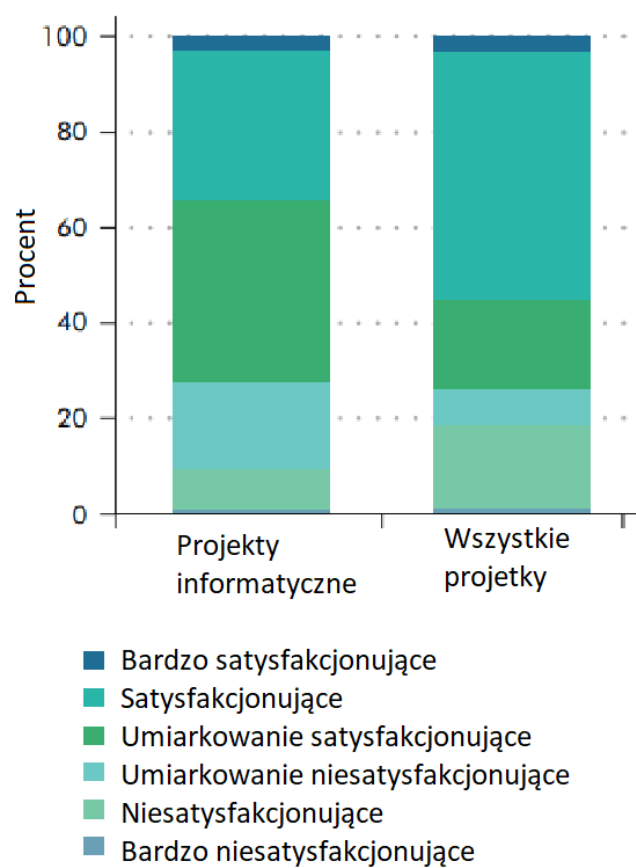
Rysunek 2.7. Wskaźnik sukcesu dużych projektów informatycznych w sektorze publicznym



Źródło: World Development Report, World Bank Group 2016

²⁶ World Bank Group. *World Development Report*. 2016.

Rysunek 2.8. Porównanie sukcesu dla projektów informatycznych i wszystkich projektów



Źródło: *World Development Report*, World Bank Group 2016

3. Specyfika projektów informatycznych

Skuteczne dostarczanie oczekiwanych korzyści z projektów informatycznych stanowi wyzwanie dla wielu organizacji.²⁷ Badania branżowe²⁸ oraz raporty rządowe^{29,30} wzywają do poprawy efektywności realizacji projektów IT. Połączenie rosnącej zależności od systemów informatycznych oraz wzrastających kosztów wdrażania takich projektów sugeruje, że wydajność projektów IT jest kluczową kwestią organizacyjną.³¹ Jednak w związku ze specyficzną naturą projektów informatycznych są one wyzwaniem. Projekty informatyczne cechują się wyjątkową heterogenicznością w porównaniu z inicjatywami w innych branżach. Część projektów ogranicza się jedynie do wdrażania gotowych rozwiązań sprzętowych i oprogramowania przez stosunkowo małe grupy specjalistów. Inne projekty wymagają zaś zaangażowania setek osób w analizę procesów biznesowych wielu odrębnych podmiotów, a następnie współtworzenie wraz z użytkownikami dopasowanych systemów informatycznych. Ważnym aspektem projektów informatycznych jest różnorodność dostępnych technologii. Nawet w przypadku prostych projektów sprzętowych występuje szerokie spektrum urządzeń – od komputerów stacjonarnych i mainframe'ów po infrastrukturę sieciową, terminale mobilne czy urządzenia IoT. Łączność w takich projektach może opierać się na technologiach bezprzewodowych, światłowodowych, komórkowych lub też satelitarnych. Jest to jednak stosunkowo mała złożoność w porównaniu do projektów programistycznych, gdzie ilość dostępnych rozwiązań i technologii jest wielokrotnie większa. Same projekty programistyczne również mogą być skrajnie różne - od tworzenia podstawowych narzędzi w środowisku Excel po rozwój międzynarodowych platform handlu elektronicznego, wykorzystujących zaawansowane frameworki i działających w środowisku wieloplatformowym.

Projekty informatyczne obejmują praktycznie wszystkie dziedziny gospodarki i funkcje biznesowe. Jednak kompetencje niezbędne do zarządzania projektem informatycznym dla studia animacji różnią się od tych wymaganych przy modernizacji rządowego systemu podatkowego czy wdrażaniu nowej sieci telekomunikacyjnych w regionie. Ta różnorodność,

²⁷ K. Lyytinen i D. Robey. „Learning failure in information systems development.” W: *Information Systems Journal* (1999).

²⁸ D. Rubinstein. „Standish Group report: There's less development chaos today”. W: *Software Development Times* (2007).

²⁹ Report of the Auditor General of Canada. *Systems under development: Managing the risks*. 1995.

³⁰ Parliamentary Office of Science Report Summary no. 200 i Technology. *Government IT projects*. 2003.

³¹ A. Gemino, B. Reich i C. Sauer. „A Temporal Model of Information Technology Project Performance”. W: *J. of Management Information Systems* 24 (sty. 2008), s. 9–44. DOI: 10.2753/MIS0742-1222240301.

połączona z dynamicznym rozwojem branży i mnogością technologii, powoduje, że projekty informatyczne znacząco się wyróżniają na tle reszty dziedzin.³²

3.1. Niechęć do projektów informatycznych

Kwestia projektowania i wdrażania systemów informatycznych bywa często postrzegana jako abstrakcyjna nawet przez osoby regularnie korzystające z technologii w życiu zawodowym, jak również prywatnym. Użytkownicy nie mają świadomości związku między tym, co widzą na ekranie, a stojącym za tym złożonym procesem tworzenia oprogramowania. Niestety, problem ten sięga głębiej, często pomija się zależność między funkcjonalnością systemu a koniecznością przeprowadzenia rzetelnego projektu w celu jej wprowadzenia. Co więcej, próby zaangażowania przyszłych użytkowników w definiowanie wymagań spotykają się z oporem, gdyż w społecznym odbiorze wystarczające wydaje się wykorzystanie już istniejących narzędzi. W przypadku skomplikowanych projektów innowacyjnych ujawnia się kolejne stereotypowe przekonanie, że proces adaptacji technologii w organizacji postrzegany jest jako uciążliwy oraz zakłócający codzienne obowiązki.

Mimo powszechnego już korzystania z oprogramowania, obserwuje się niedostatek wiedzy o wzajemnych powiązaniach poszczególnych etapów tworzenia systemów oraz o ich ciągłej ewolucji. W ostatnich latach nastąpiła jednak znacząca zmiana. Współczesne podejście do wytwarzania oprogramowania kładzie nacisk na dostosowanie rozwiązań do realnych potrzeb użytkowników końcowych oraz zwiększenie efektywności ekonomicznej projektów. Ta orientacja kontrastuje wyraźnie z bezrefleksyjną implementacją technologii za wszelką cenę z początku stulecia i stanowi istotny postęp.³³ Thomas i Hunt w swojej książce zawarli ważne zdanie na ten temat: "programiści pomagają użytkownikom zrozumieć, czego użytkownicy chcą".³⁴ To stwierdzenie uwypukla kluczową rolę zrozumienia i realizacji potrzeb użytkowników w projektach informatycznych.

Możemy zbudować system, który jest sukcesem technicznym, lecz porażką organizacyjną. Systemy informatyczne – produkty projektów IT – są planowaną zmianą organizacyjną. Technologia informatyczna jest czynnikiem umożliwiającym tworzenie nowych produktów,

³² K. Schwalbe. *Information Technology Project management*. Cengage, 2018.

³³ W. Chmielarz. *Information Technology Project management*. Wydawnictwo Naukowe Wydziału Zarządzania, 2015.

³⁴ D. Thomas i A. Hunt. *Pragmatyczny programista Od czeladnika do mistrza*. Helion, 2021.

usług i procesów, które mogą zmieniać istniejące relacje między organizacją a jej klientami lub dostawcami, a także między osobami w ramach samej organizacji.

Ta zmiana może stanowić zagrożenie dla wielu grup. Dlatego ludzie mogą nie zawsze być przychylni nowemu systemowi informatycznemu – niezależnie od tego, jak dobrze został zbudowany lub jak nowoczesne są zastosowane technologie, narzędzia i techniki. Z drugiej strony, osoby w organizacji mogą słusznie opierać się systemowi, który nie działa prawidłowo lub nie spełnia ich oczekiwań.³⁵

Opisane wyzwania dotyczą niestety także części specjalistów IT, którzy niekiedy ograniczają się do stosowania sprawdzonych, lecz nie zawsze adekwatnych schematów. Przekonanie, że „w informatyce nie ma już nic nowego” czy „wszystkie innowacje zostały wdrożone” bywa pułapką utrudniającą odpowiedź na zmieniającą się rzeczywistość – zarówno wirtualną, jak i realną.³⁶

Konieczne jest podejście, które nie stawia strony technicznej ponad organizacyjną ani odwrotnie. Aby dostarczyć udany projekt, należy zachować równowagę między aspektami technicznymi a organizacyjnymi projektów IT.³⁷

3.2. Cykl życia w projektach informatycznych

W inżynierii oprogramowania wyróżnia się różne rodzaje projektów (np. rozwój produktu, usługi outsourcingowe, konserwacja oprogramowania, tworzenie usług itp.). W cyklu życia produktu programistycznego może być wymaganych wiele projektów. Przykładowo, na etapie koncepcji produktu może zostać przeprowadzony projekt w celu określenia potrzeb klientów i wymagań rynkowych lub podczas konserwacji może zostać zrealizowany projekt mający na celu opracowanie kolejnej wersji produktu.³⁸

3.2.1. Cykl życia projektu

Podobnie jak wszystkie żywe istoty, projekty mają cykle życia, w których rodzą się, rosną, osiągają szczyt, upadają, a następnie kończą się.³⁹ Chociaż cykle życia projektów mogą się różnić w zależności od branży lub projektu, wszystkie cykle życia projektów będą

³⁵ J. Marchewka. *Information Technology Project management*. Northern Illinois University, 2003.

³⁶ Chmielarz, *Information Technology Project management*.

³⁷ Marchewka, *Information Technology Project management*.

³⁸ H. Washizaki. *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*. Waseda University Japan, 2024.

³⁹ J. Gido i J. P. Clements. „Successful Project Management”. W: *South Western Hill* (1999).

miały początek, środek i koniec.⁴⁰ Cykl życia projektu (PLC) jest to ciąg faz, przez które projekt przechodzi od rozpoczęcia po zakończenie.⁴¹ Faza projektu zaś to zbiór logicznie powiązanych działań, które kończą się wytworzeniem jednego lub więcej rezultatów (np. dokumentacji, prototypu, gotowego modułu). Każda faza ma określone ramy czasowe – rozpoczyna się i kończy w punkcie kontrolnym (ang. phase gate). Według PMBOK® Guide, podstawowy cykl obejmuje cztery etapy:

- rozpoczęcie projektu,
- organizacja i przygotowanie,
- realizacja prac,
- zakończenie projektu.

Ogólnie rzecz biorąc, cykle życia projektu określają, jaka praca zostanie wykonana w każdej fazie, jakie produkty zostaną wyprodukowane i kiedy, kto jest zaangażowany w każdą fazę oraz w jaki sposób kierownictwo będzie kontrolować i zatwierdzać pracę wyprodukowaną w każdej fazie. Produktem może być faktycznie jakiś produkt lub usługa, taka jak na przykład raport techniczny czy sesja szkoleniowa.⁴²

3.2.2. Cykl życia produktu

Podobnie jak projekty, produkty również mają swój cykl życia. Cykl życia produktu to seria faz, które reprezentują ewolucję produktu, od koncepcji, poprzez dostawę, wzrost, dojrzałość, aż do wycofania.⁴³

Ta definicja nie jest specyficzna dla systemów programistycznych, lecz ma zastosowanie ogólnie do wszystkich produktów. Podobnie koncepcja cyklu życia, powiązana z pojęciem produktu, nie jest wyłączną domeną inżynierii oprogramowania.

Systemy programistyczne zawierają jednostki oprogramowania (ang. software units), które są „podstawowymi komponentami architektury oprogramowania, podlegającymi niezależnemu testowaniu”. Cykl życia systemu programistycznego (uwzględniający interdyscyplinarne podejście inżynierii oprogramowania) obejmuje wszystkie procesy, działania i zadania – od powstania koncepcji systemu aż do jego wycofania, włączając w to produkcję, eksploatację,

⁴⁰ J. Rosenau. *Successful Project Management*. John Wiley & Sons Australia, Limited, 1998.

⁴¹ Project Management Institute. *PMBOK Guide – Seventh Edition*. 2021.

⁴² Schwalbe, *Information Technology Project management*.

⁴³ Project Management Institute. *PMBOK Guide – Sixth Edition*. 2017.

ewolucję, a także – w razie potrzeby – pozyskanie i dostarczenie. Analogicznie można rozpatrywać cykl życia pojedynczego elementu systemu (np. jednostki oprogramowania).⁴⁴

Większość specjalistów IT zna koncepcję cyklu życia produktu, szczególnie w przypadku rozwoju oprogramowania.⁴⁵ Choć projekty podlegają cyklowi życia projektu (PLC), rozwój systemów informatycznych podlega cyklowi życia produktu. Najpopularniejszym cyklem w branży IT jest Cykl Życia Oprogramowania (SDLC), który obejmuje sekwencyjne fazy, przez które przechodzi system informatyczny w trakcie swojego istnienia.⁴⁶ SDLC określa logiczną kolejność etapów rozwoju systemu oraz decyduje o przejściu od jednej czynności do kolejnej.⁴⁷

Podstawowe fazy SDLC to

- planowanie,
- analiza,
- projektowanie,
- wdrożenie,
- utrzymanie i wsparcie.⁴⁸

3.2.3. Integracja PLC i SDLC w projektach informatycznych

Kluczowa różnica między cyklem życia projektu a cyklem życia oprogramowania polega na tym, że PLC koncentruje się na zarządzaniu procesem realizacji projektu, podczas gdy SDLC dotyczy tworzenia i implementacji produktu – systemu informatycznego. W projektach IT oba cykle często się przenikają – np. fazy SDLC (projektowanie) realizowane są w ramach etapów PLC. Połączenie zarządzania projektem z cyklem życia oprogramowania jest cechą charakterystyczną projektów IT. Jak pokazuje Rysunek 4.1, SDLC stanowi część PLC, ponieważ większość działań związanych z rozwojem systemu odbywa się w fazie realizacji projektu. Etapy zamknięcia i ewaluacji projektu następują dopiero po wdrożeniu systemu.⁴⁹

⁴⁴ Washizaki, *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*.

⁴⁵ Schwalbe, *Information Technology Project management*.

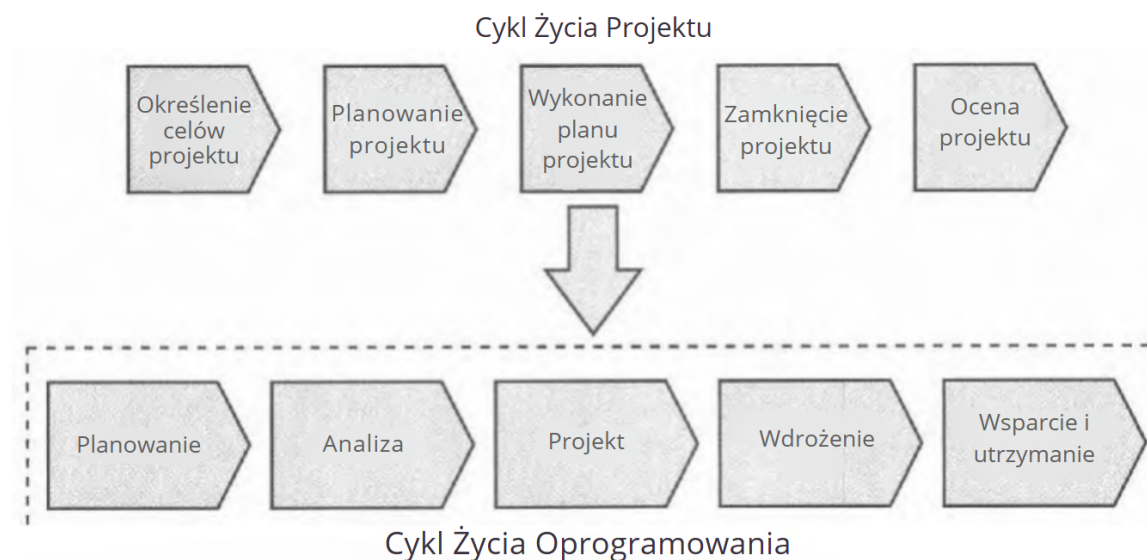
⁴⁶ Marchewka, *Information Technology Project management*.

⁴⁷ S. McConnell. *Rapid Development: Taming Wild Software Schedules*. Microsoft Press, 1996.

⁴⁸ Marchewka, *Information Technology Project management*.

⁴⁹ Tamże.

Rysunek 3.1. Porównanie cyklu życia oprogramowania do cyklu życia projektu



J. T. Marchewka. Information Technology Project management. Northern Illinois University, 2003

3.3. Modele SDLC

W inżynierii oprogramowania koncepcja SDLC stanowi podstawę wielu modeli cyklu życia oprogramowania. Modele te tworzą ramy planowania i kontroli procesu tworzenia systemu informatycznego.⁵⁰ Wszystkie sugerowane modele SDLC dzielą podstawowe cechy: składają się z sekwencji faz lub kroków, które muszą być przestrzegane i zakończone przez programistów oraz projektantów systemów, aby osiągnąć rozwinięte systemy i dostarczyć wymagane produkty.⁵¹ Każdy system oprogramowania posiada unikalne cechy odzwierciedlające potrzeby interesariuszy – zarówno biznesowe, jak i techniczne. Odpowiedni cykl życia powinien uwzględniać wszystkie te wymagania.⁵² Wybór modelu SDLC powinien być dokonywany na podstawie wymagań projektu.⁵³

PMBOK® krótko opisuje pięć cykli życia produktu lub rozwoju. Dwa czynniki są istotne przy wyborze odpowiedniego cyklu życia: stopień zmienności wymagań oraz częstotliwość

⁵⁰ Washizaki, *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*.

⁵¹ A. Alshamrani i A. Bahattab. „A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model”. W: *International Journal of Computer Science Issues (IJCSI)* 12.1 (2015), s. 106.

⁵² Washizaki, *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*.

⁵³ R. Arora i N. Arora. „Analysis of SDLC models”. W: *International Journal of Current Engineering and Technology* 6.1 (2016), s. 268–272.

dostarczania użytecznych wyników. Na przykład, dla produktu o niskim stopniu zmienności wymagań i niskiej częstotliwości dostarczania, odpowiedni będzie cykl życia predykcyjny.⁵⁴⁵⁵

- **Cykl życia predykcyjny:** Zakres, harmonogram i koszty są określone na wczesnym etapie, a zmiany w zakresie są ściśle kontrolowane. PMI określa również cykle predykcyjne jako model kaskadowy (waterfall).⁵⁶⁵⁷
- **Cykl życia iteracyjny:** Zakres jest określany na wczesnym etapie, ale szacowania czasu i kosztów są modyfikowane w miarę wzrostu wiedzy zespołu o produkcie. Iteracje służą do rozwijania produktu poprzez powtarzające się cykle, które dodają kolejne funkcjonalności.⁵⁸⁵⁹⁶⁰ Długość iteracji definiuje się dla każdego projektu.⁶¹ To podejście najlepiej sprawdza się w przypadku wysokiego stopnia zmienności i niskiej częstotliwości dostarczania.⁶²
- **Cykl życia przyrostowy:** Produkt końcowy jest dostarczany poprzez serię iteracji stopniowo dodających funkcjonalność w określonym przedziale czasowym. Produkt osiąga pełną funkcjonalność dopiero po ostatniej iteracji.⁶³⁶⁴⁶⁵ To podejście najlepiej sprawdza się w przypadku niskiego stopnia zmienności i wysokiej częstotliwości dostarczania.⁶⁶
- **Cykl życia adaptacyjny:** Interesariusze definiują i zatwierdzają szczegółowy zakres przed rozpoczęciem iteracji, dostarczając użyteczny produkt na końcu każdej iteracji. PMI określa również cykle adaptacyjne jako zwinne (agile) lub sterowane zmianą. To podejście najlepiej sprawdza się w przypadku wysokiego stopnia zmienności i wysokiej częstotliwości dostarczania.⁶⁷⁶⁸
- **Cykl życia hybrydowy:** Jest to kombinacja podejścia adaptacyjnego i predykcyjnego.

⁵⁴ Institute, *PMBOK Guide – Sixth Edition*.

⁵⁵ Schwalbe, *Information Technology Project management*.

⁵⁶ Institute, *PMBOK Guide – Fourth Edition*.

⁵⁷ Schwalbe, *Information Technology Project management*.

⁵⁸ I. Sommerville. *Software Engineering*. 10th. Pearson, 2015.

⁵⁹ D. Farley. *Modern Software Engineering: Doing What Works to Build Better Software Faster*. Addison-Wesley Professional, 2021.

⁶⁰ Institute, *PMBOK Guide – Sixth Edition*.

⁶¹ Washizaki, *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*.

⁶² Schwalbe, *Information Technology Project management*.

⁶³ Sommerville, *Software Engineering*.

⁶⁴ Farley, *Modern Software Engineering: Doing What Works to Build Better Software Faster*.

⁶⁵ Institute, *PMBOK Guide – Sixth Edition*.

⁶⁶ Schwalbe, *Information Technology Project management*.

⁶⁷ Institute, *PMBOK Guide – Sixth Edition*.

⁶⁸ Schwalbe, *Information Technology Project management*.

Oznacza to, że niektóre elementy pochodzą z podejścia predykcyjnego, a inne z podejścia adaptacyjnego. Takie podejście do rozwoju jest przydatne, gdy istnieje niepewność lub ryzyko związane z wymaganiami. Hybrydowe podejście sprawdza się również, gdy dostarczane elementy mogą być modularne lub gdy różne zespoły projektowe mogą opracowywać różne części produktu. Podejście hybrydowe jest bardziej adaptacyjne niż podejście predykcyjne, ale mniej niż podejście w pełni adaptacyjne.⁶⁹

Kilka modeli cyklu życia stało się dobrze znanych wraz z rozwojem inżynierii oprogramowania od jej początków, dwa z nich zostaną omówione w kolejnych podrozdziałach.

3.3.1. Model kaskadowy

Jednym z modeli, który zdobył popularność we wczesnej historii dyscypliny inżynierii oprogramowania, jest model kaskadowy.⁷⁰ Model kaskadowy, zaliczany do modeli predykcyjnych, zakłada określenie zakresu, harmonogramu i kosztów już na wczesnym etapie, a wszelkie zmiany są ściśle kontrolowane. Podejście to obejmuje fazy takie jak: zbieranie wymagań, projekt wstępny, projekt szczegółowy, kodowanie oraz testowanie, przy czym kolejna faza nie może się rozpocząć, dopóki poprzednia nie zostanie zakończona.⁷¹ Po zakończeniu fazy nie wracamy już do niej, dlatego model ten jest zazwyczaj planowany przy użyciu wykresu Gantta. Podstawowe założenie metody kaskadowej polega na zapewnieniu stabilności wymagań. Wymagania określają potrzebne informacje, funkcje, zachowanie, wydajność oraz interfejsy.⁷²

Postęp w modelu kaskadowym przebiega z góry na dół, przypominając wodospad. Model ten wywodzi się z przemysłu produkcyjnego oraz budownictwa, czyli silnie ustrukturyzowanych środowisk fizycznych, w których wprowadzanie zmian po fakcie jest niezwykle kosztowne, a często wręcz niemożliwe. Ponieważ w tamtym czasie nie istniały formalne metodologie tworzenia oprogramowania, ten produkcyjny model został po prostu zaadaptowany do potrzeb rozwoju oprogramowania.⁷³

Pierwszy formalny opis modelu kaskadowego często przypisuje się artykułowi opubliko-

⁶⁹ Institute, *PMBOK Guide – Seventh Edition*.

⁷⁰ Sommerville, *Software Engineering*.

⁷¹ Washizaki, *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*.

⁷² Arora i Arora, „Analysis of SDLC models”.

⁷³ V. Suburayan, P. Davachelvan i S. Kayalvizhi, „Evolving A New Model (SDLC Model2010) For Software Development Life Cycle (SDLC)”. W: *International Journal of Computer Science and Network Security* (2010).

wanemu w 1970 roku przez Winstona W. Royce'a (1929–1995),⁷⁴ chociaż w swoim artykule nie użył on terminu „waterfall” (kaskadowy). Royce przedstawił ten model jako przykład wadliwego, nieskutecznego podejścia. W rzeczywistości właśnie w ten sposób termin ten był najczęściej używany w literaturze dotyczącej inżynierii oprogramowania – jako sposób na krytykę powszechnie stosowanej praktyki wytwarzania oprogramowania.⁷⁵

Model kaskadowy był jednak użyteczny, gdyż wprowadził systematyzację w rozwoju systemów oprogramowania, czyli inżynierskie podejście do tworzenia produktów programistycznych.⁷⁶

Zalety podejścia kaskadowego

- Proste i łatwe w użyciu oraz zrozumieniu.
- Fazy nie nakładają się na siebie, co pozwala na efektywne zarządzanie projektem.
- Bardzo dobrze sprawdza się w małych projektach.

Wady podejścia kaskadowego

- Nie można uzyskać akceptacji użytkownika aż do zakończenia całego procesu.
- Sukces projektu jest niepewny aż do jego zakończenia.

Kiedy stosować metodę kaskadową

- Model najlepiej sprawdza się, gdy wszystkie wymagania są jasne, stałe i dobrze określone.
- Definicja produktu i wykorzystywana technologia są dobrze znane.
- Brak niejednoznacznych wymagań.
- Czas trwania projektu jest krótki.⁷⁷

3.3.2. Agile

Kolejnym popularnym modelem jest agile. Jest to model iteracyjny, w którym fazy projektu przebiegają równocześnie, tj. równolegle. Wymagania mogą ewoluować na każdym etapie i w każdej fazie.⁷⁸

⁷⁴ W. W. Royce. „Managing the Development of Large Software Systems”. W: *Proceedings of IEEE WESCON* (1970), s. 1–9.

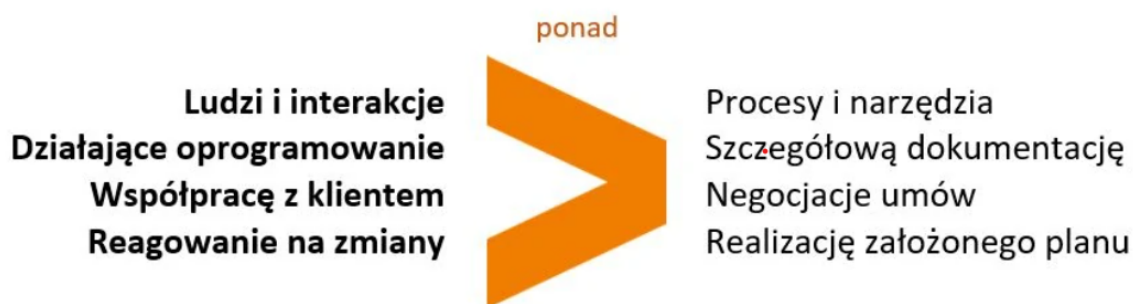
⁷⁵ Suburayan, Davachelvan i Kayalvizhi, „Evolving A New Model (SDLC Model2010) For Software Development Life Cycle (SDLC)”.

⁷⁶ Washizaki, *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*.

⁷⁷ Arora i Arora, „Analysis of SDLC models”.

⁷⁸ Tamże.

Rysunek 3.2. Wartości Agile



Źródło: <https://www.intellect.pl/blog/dlaczego-agile/>

Manifest Agile⁷⁹ wywołał rewolucję w społeczności inżynierii oprogramowania, wprowadzając przekonanie, że proces powinien być otwarty na zmiany – wymagania mogą być modyfikowane na każdym etapie, jeśli zmieniają się potrzeby użytkowników. Kluczowe stały się komunikacja i wzajemne zaufanie między zespołem a klientem. Sygnatariusze Manifestu podkreślali, że choć komunikacja (często twarzą w twarz) i interakcje z klientem są niezwykle istotne, dokumentacja (np. do definiowania wymagań) pozostaje konieczna. Postulowali także małe, przyrostowe dostawy oprogramowania, w przeciwieństwie do modelu kaskadowego, gdzie całość produktu dostarczana jest dopiero na zakończenie projektu.⁸⁰

Agile wyraźnie rozróżnia wartości i zasady (np. ciągłe dostarczanie wartości czy zobowiązanie do doskonałości technicznej) od praktyk (takich jak programowanie w parach, planowanie sprintów czy retrospektywy). Mentalność Agile, oparta na zasadach takich jak otwartość na zmiany i nacisk na komunikację, różni się od podejścia predykcyjnego, które skupia się na realizacji ustalonych specyfikacji wymagań. Agile pomaga również radzić sobie ze złożonością systemów.⁸¹

Możemy postrzegać metody zwinne, takie jak Extreme Programming (XP) i Scrum, jako reakcję na modele predykcyjne, które kładą nacisk na "zracjonalizowane, inżynierskie podejście",⁸² obejmujące szczegółowe planowanie, sformalizowane procesy oraz rygorystyczne wykorzystanie komponentów.⁸³ W przeciwieństwie do nich, metody zwinne odpowiadają na

⁷⁹ Agile Alliance. *Manifesto for Agile Software Development*. 2001. URL: <http://agilemanifesto.org/>.

⁸⁰ Washizaki, *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*.

⁸¹ Tamże.

⁸² T. Dybå. „Improvisation in Small Software Organizations”. W: *IEEE Software* 17.5 (2000), s. 82–87.

⁸³ B. Boehm. „Get Ready for Agile Methods, with Care”. W: *Computer* 35.1 (2002), s. 64–69.

wyzwania nieprzewidywalnego świata, podkreślając wartość, jaką ludzie i ich relacje wnoszą w rozwój oprogramowania.⁸⁴⁸⁵

Wokół Agile narosło wiele nieporozumień. Niektórzy myślą, że Agile stanowi kompletną metodę – jednak nie jest ona metodą samą w sobie, lecz zbiorem wartości, zasad i praktyk. Inne błędne przekonanie mówi, że Agile jest szybszy niż model kaskadowy, gdyż nie wymaga tworzenia dokumentacji. Kolejnym mitem jest pogląd, że Agile opiera się na ograniczonym zbiorze metod. Mimo rosnącej popularności modelu Agile, skalowanie go na duże projekty i portfele pozostaje wyzwaniem. Obecnie uważa się, że Manifest Agile przyniósł znaczące zmiany, jednak po 20 latach niektórzy autorzy sugerują konieczność aktualizacji pewnych jego zasad H. Washizaki. *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*. Waseda University Japan, 2024.

Agile to nie tylko zbiór technik programistycznych, ale "ruch, ideologia, filozofia". Idąc jeszcze dalej, jeden z twórców Scruma stwierdza: „Agile to emocja”.⁸⁶ Zastosowanie praktyk Agile wykracza poza sam proces inżynierii oprogramowania – terminy takie jak „zwinność biznesowa” oraz „zwinne organizacje” stały się powszechne. Z perspektywy inżynierii oprogramowania Agile umożliwił ponowną inżynierię i lepsze dopasowanie procesów technicznych do strategicznych celów biznesowych. Wykorzystanie podejścia Agile w procesach biznesowych znajduje odzwierciedlenie między innymi w zasadach DevOps oraz w procesach oceny i doskonalenia jakościowych.⁸⁷

Zalety podejścia Agile

- Zmiany można wprowadzać bardzo łatwo.
- Zmiany można obsłużyć bez dużych kosztów ponownego planowania.
- Stałe zaangażowanie zespołu.

Wady podejścia Agile

- Częściej pojawiają się problemy ze strukturą zespołu niż problemy z planowaniem.
- Niepewne terminy planowania.

Kiedy stosować podejście Agile

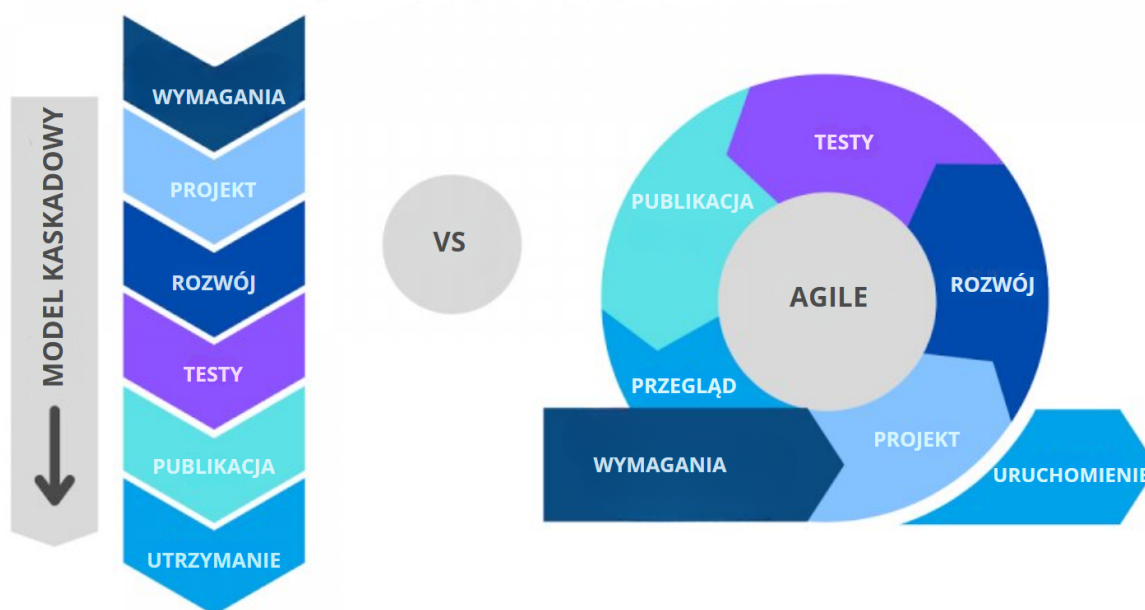
⁸⁴ T. Dybå i T. Dingsøyr. „What do we know about agile software development?” W: *IEEE software* 26.5 (2009), s. 6–9.

⁸⁵ S. Nerur i V. Balijepally. „Theoretical Reflections on Agile Development Methodologies”. W: *Communications of the ACM* 50.3 (2007), s. 79–83.

⁸⁶ B. Meyer. „Agile”. W: *The good, the hype and the ugly*. Switzerland: Springer International Publishing (2014).

⁸⁷ Washizaki, *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*.

Rysunek 3.3. Porównanie modelu kaskadowego z Agile



Źródło: <https://www.linkedin.com/pulse/waterfall-vs-agile-which-better-you-why-datacademy-cloud/>

- Gdy istnieje ciągła potrzeba wprowadzania zmian.
- Gdy zarówno programiści, jak i inni interesariusze mają swobodę sugerowania zmian.⁸⁸

3.4. Różnorodne technologie

Różnorodność technologii w branży IT jest bardzo duża i kompetencje oraz wiedza specjalistów mogą się bardzo różnić pomiędzy zajmowanymi stanowiskami. Różnice te często znacznie utrudniają komunikację nawet wewnątrz zespołu projektowego. Na przykład:

- Specjaliści od sprzętu mogą nie rozumieć języka analityków baz danych i odwrotnie.
- Eksperti ds. bezpieczeństwa mogą mieć trudności w dialogu z analitykami biznesowymi.

Nawet osoby pracujące w tej samej dziedzinie IT czasem się nie rozumieją, jeśli korzystają z innych technologii. Przykładowo, jeśli programista jest specjalistą w języku Java, to może mieć problem ze zrozumieniem pewnych zagadnień charakterystycznych dla języka Python, co ponownie może powodować zakłócenia w komunikacji.

Kolejnym wyzwaniem związanym z różnorodnością technologii jest ich szybka zmienność. Zespół może być bliski ukończenia projektu, gdy odkryje nowe rozwiązanie, które znacząco

⁸⁸ Arora i Arora, „Analysis of SDLC models”.

ulepszy produkt i lepiej zaspokoi długoterminowe potrzeby biznesowe. Nowe technologie skróciły też czas, jaki firmy mają na opracowanie, wdrożenie i dystrybucję produktów czy usług. To dynamiczne środowisko wymaga równie szybkich procesów zarządzania projektami IT.⁸⁹

3.5. Specjaliści IT

Inwestycje w technologie informatyczne są ściśle związane z rosnącym zapotrzebowaniem na wykwalifikowanych pracowników. Innymi słowy, jednym z głównych czynników sukcesu projektów IT są umiejętności techniczne zespołu. Według Tohidi specjaliści IT są zazwyczaj silni w dziedzinie podejmowania decyzji, jednak mają oni często duże braki komunikacyjne. Większość z nich jest introwertykami i brakuje im umiejętności, takich jak:

- Umiejętności wsparcia: Przyjazne i taktowne podejście, cierpliwość, empatia wobec osób zestresowanych lub mających obawy, wysłuchiwanie skarg i problemów oraz uwzględnianie potrzeb innych.
- Szkolenie i doradztwo: Dostarczanie szkoleń i praktycznych porad, tworzenie kontekstu ułatwiającego rozwój umiejętności, wspieranie przedsiębiorczości i optymalizacja procesów biznesowych.
- Zarządzanie postawami i budowanie zespołu: Rozwiązywanie konfliktów, motywowanie do współpracy, identyfikowanie kluczowych obszarów biznesowych.
- Networking: Nieformalna komunikacja z osobami posiadającymi wartościowe informacje, utrzymywanie regularnych kontaktów (spotkania, rozmowy telefoniczne, korespondencja) oraz uczestnictwo w wydarzeniach społecznych.

Według badań komunikacja ma zaś bezpośrednie przełożenie na sukces projektu.⁹⁰ Nawet technicznie doskonałe inicjatywy często upadają z powodu jej braku.⁹¹

3.6. Ryzyka w projektach IT

Wszystkie projekty są ryzykowne, ponieważ stanowią unikalne przedsięwzięcia o różnym stopniu złożoności, których celem jest dostarczanie korzyści. Realizowane są one w kontekście

⁸⁹ Schwalbe, *Information Technology Project management*.

⁹⁰ M. Celestin, H. Jean Paul i N. Clementine. „Analysis of Effective Communication and Project Success: Survey on Electricity Access Roll Out Project at EDCL -EARP”. W: *International Journal of Computational Research and Development* (2023).

⁹¹ H. Tohidi. „Human resources management main role in information technology project management”. W: *Procedia CS* 3 (grud. 2011), s. 925–929. doi: 10.1016/j.procs.2010.12.151.

ograniczeń i założeń, przy jednoczesnym uwzględnianiu oczekiwań interesariuszy, które mogą być sprzeczne i zmienne. Organizacje powinny podejmować ryzyko projektowe w sposób kontrolowany i świadomy, aby tworzyć wartość, równoważąc ryzyko i zysk.⁹²

Niepewność to brak zrozumienia i świadomości w zakresie problemów, zdarzeń, ścieżek działania lub rozwiązań do wdrożenia. Ryzyko zaś jest aspektem niepewności. Ryzyko definiuje się jako niepewne zdarzenie lub warunek, które w przypadku wystąpienia może wywrzeć pozytywny jak i negatywny wpływ na jeden lub więcej celów projektu. Ryzyka negatywne nazywane są zagrożeniami, zaś pozytywne – szansami. Wszystkie projekty obarczone są ryzykiem, ponieważ są to unikalne przedsięwzięcia charakteryzujące się różnym stopniem niepewności.⁹³

Projekty informatyczne niosą za sobą pewne specyficzne dla nich aspekty ryzyka, takie jak skłonność inżynierów do dodawania niepotrzebnych funkcji (ang. feature creep) czy ryzyka związane z niematerialnym charakterem oprogramowania. Szczególną uwagę należy poświęcić ryzykom związanym z wymaganiami jakościowymi oprogramowania, takimi jak bezpieczeństwo funkcjonalne (ang. safety) czy bezpieczeństwo IT (ang. security).^{94,95}

Poniżej została przedstawiona przykładowa propozycja kwestionariusza ryzyk dla projektu informatycznego, ryzyka zostały podzielone na pięć kategorii:⁹⁶

- Ryzyko rynkowe:
 - Czy projekt IT, który ma stworzyć nowy produkt lub usługę, będzie przydatny dla organizacji lub nadający się do sprzedaży?
 - Czy użytkownicy zaakceptują i będą korzystać z produktu/usługi?
 - Czy konkurencja nie opracuje lepszego rozwiązania szybciej, czyniąc projekt stratą czasu i pieniędzy?
- Ryzyko finansowe:
 - Czy organizacja może sobie pozwolić na realizację projektu?
 - Jak bardzo interesariusze ufają prognozom finansowym?

⁹² Institute, *PMBOK Guide – Sixth Edition*.

⁹³ Institute, *PMBOK Guide – Seventh Edition*.

⁹⁴ K. Nidiffer, C. Woody i T. A. Chick. *Program Manager's Guidebook for Software Assurance*. Software Solutions i CERT Divisions, Software Engineering Institute, 2018.

⁹⁵ Washizaki, *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*.

⁹⁶ Schwalbe, *Information Technology Project management*.

- Czy projekt spełni szacunki dotyczące wartości bieżącej netto (NPV), zwrotu z inwestycji (ROI) i okresu zwrotu?
- Jeśli nie, czy organizacja będzie w stanie kontynuować projekt?
- Czy ten projekt to najlepsze wykorzystanie środków finansowych organizacji?
- Ryzyko technologiczne:
 - Czy projekt jest wykonalny technicznie?
 - Czy wykorzysta technologie dojrzałe, nowoczesne czy eksperymentalne?
 - Kiedy zapadną decyzje dotyczące wyboru technologii?
 - Czy sprzęt, oprogramowanie i sieci będą działać prawidłowo?
 - Czy technologia będzie dostępna na czas, aby osiągnąć cele projektu?
 - Czy technologia zdezaktualizuje się, zanim powstanie użyteczny produkt?
- Ryzyko kadrowe:
 - Czy organizacja ma pracowników z odpowiednimi kompetencjami, aby pomyślnie zrealizować projekt?
 - Jeśli nie, czy można znaleźć takie osoby?
 - Czy zespół posiada umiejętności menedżerskie i techniczne?
 - Czy członkowie zespołu mają wystarczające doświadczenie?
 - Czy kierownictwo wyższego szczebla wspiera projekt?
 - Czy istnieje rzecznik projektu (ang. project champion)?
 - Czy organizacja zna sponsora/klienta projektu?
 - Jak dobra jest relacja ze sponsorem/klientem?
- Ryzyko strukturalne/procesowe:
 - Na jakim poziomie zmiany wprowadzi projekt w obszarach użytkowników i procedurach biznesowych?
 - Ile różnych grup użytkowników musi zadowolić projekt?
 - Z iloma innymi systemami musi współdziałać nowy projekt/system?
 - Czy organizacja ma wypracowane procesy umożliwiające sukces projektu?

3.6.1. Czasowy model ryzyk

Ciekawa interpretacja i zrozumienie ryzyk w projektach informatycznych zostało zawarte w pracy Gemino.⁹⁷ Zwraca on uwagę na czasowy charakter projektów (rozumiany jako przejście od warunków początkowych, przez przebieg projektu, aż do jego rezultatów). Dzięki czemu proponuje inną formę kategoryzacji ryzyk w projektach informatycznych, uwzględniającą różnice między ryzykami obecnymi w momencie definiowania projektu a tymi, które pojawiają się lub są ujawniane podczas jego realizacji.

Ryzyka a priori

Niektóre cechy projektu, takie jak budżet, czas trwania, złożoność techniczna, niepewność wymagań, brak doświadczenia zespołu oraz niewystarczająca wiedza sponsora projektu, mogą być oszacowane przed rozpoczęciem projektu. Są to ryzyka a priori. Można wyróżnić dwie kategorie ryzyk a priori:

- Ryzyka związane z elementami strukturalnymi projektu (np. czas trwania, budżet, nakład pracy, złożoność techniczna)
- Ryzyka wynikające z wiedzy dostępnej dla zespołu (np. kompetencje kierownika projektu, sponsora, członków zespołu)

Elementy strukturalne, takie jak wielkość projektu i złożoność techniczna, są uznawane za istotne ryzyka.⁹⁸⁹⁹ Są one znane a przynajmniej potencjalnie możliwe do określenia na początku projektu. Zarządzanie nimi wymaga stosowania tradycyjnych ("twardych") technik zarządzania projektami, takich jak podział pracy, estymacja, harmonogramowanie oraz budżetowanie.

Brak zasobów wiedzy (np. niedoświadczony kierownik projektu lub zespół) jest również ważną kategorią ryzyk w projektach informatycznych.¹⁰⁰¹⁰¹ Ryzyka te wymagają zastosowania technik zarządzania, które często określa się mianem „umiejętności miękkich”, takich jak komunikacja, budowanie zespołu, uczenie się i koordynacja wiedzy.

Ryzyka wyłaniające się

⁹⁷ Gemino, Reich i Sauer, „A Temporal Model of Information Technology Project Performance”.

⁹⁸ F. W. McFarlan. „Portfolio approach to information systems.” W: *Harvard Business Review*, (1981).

⁹⁹ R. W. Zmud. „Management of large software development efforts”. W: *MIS Quarterly* (1980).

¹⁰⁰ McFarlan, „Portfolio approach to information systems.”

¹⁰¹ H. Barki, S. Rivard i J. Talbot. „Toward an assessment of software development risk.” W: *Journal of Management Information Systems* (1993).

Zanim projekt się rozpocznie, kierownik projektu wyrabia sobie oczekiwania co do poziomu wsparcia ze strony najwyższego kierownictwa oraz użytkowników. Jednakże oczekiwany poziom wsparcia lub uczestnictwa nie wpływa bezpośrednio na wyniki projektu. To właśnie rzeczywisty poziom wsparcia, ujawniony przez zachowania najwyższego kierownictwa i użytkowników, ma najbardziej bezpośredni wpływ na efektywność.¹⁰² Na przykład, choć kierownik projektu może wierzyć, że otrzyma wysoki poziom uczestnictwa użytkowników, to rzeczywisty poziom uczestnictwa ujawnia się dopiero poprzez zachowania użytkowników w trakcie realizacji projektu. Kierownik wyższego szczebla może obiecywać wysoki poziom wsparcia, ale nie dostarczyć go w praktyce. Dlatego te ryzyka można określić jako ryzyka wyłaniające się (ang. Emergent Risks).

Można wyróżnić dwie kategorie ryzyk wyłaniających się:

- Ryzyka związane z niedoborami wsparcia organizacyjnego (np. brak wsparcia sponsora, klienta lub użytkownika)
- Ryzyka związane ze zmianami, które zachodzą (np. zmiany celów, członków zespołu oraz szerszego otoczenia)

103

Pierwszą z tych kategorii to ryzyko wsparcia organizacyjnego. Istnieją trzy kluczowe obszary wsparcia organizacyjnego — wsparcie sponsora wykonawczego, zaangażowanie menedżera klienta oraz uczestnictwo użytkowników. Wsparcie organizacyjne jest uznawane za kluczowe dla sukcesu projektu. Gdy poziom wsparcia organizacyjnego jest niższy niż oczekiwano, określa się to jako ryzyko (np. brak wsparcia ze strony kierownictwa). Natomiast gdy wsparcie organizacyjne spełnia lub przewyższa oczekiwania, traktuje się je jako istotny zasób dla kierowników projektów.¹⁰⁴¹⁰⁵¹⁰⁶

Drugą kategorię ryzyk stanowią zmiany wpływające na projekty, w tym zmiany celów projektu, kluczowych osób oraz warunków zewnętrznych, z którymi projekt musi się mierzyć.

¹⁰² Standish Group. *CHAOS Report*. 2015.

¹⁰³ Gemino, Reich i Sauer, „A Temporal Model of Information Technology Project Performance”.

¹⁰⁴ M. Keil i in. „A framework for identifying software project risks.” W: *Communications of the ACM* (1998).

¹⁰⁵ R. Sharma i P. Yetton. „The contingent effects of management support and task interdependence on successful information systems implementation.” W: *MIS Quarterly* (2003).

¹⁰⁶ P. Yetton i in. „A model of information systems development project performance.” W: *Information Systems Journal* (2000).

Są to ryzyka zmienności projektu.¹⁰⁷ Kategoria ta jest podobna do kategorii ryzyka „środowiskowego” zidentyfikowanej przez Keil.¹⁰⁸ Ta zmienność często pozostaje poza kontrolą członków zespołu projektowego, ale może mieć znaczący wpływ na wyniki projektu.¹⁰⁹

Omówione kategorie ryzyk zostały przedstawione w tabelach 4.1. oraz 4.2.

3.7. Ewaluacja projektów informatycznych

Szybkie tempo zmian w technologii IT powoduje poważne trudności już na etapie rozpoczynania każdej dużej inwestycji. Każdy długoterminowy, sztywny projekt staje się niemal przestarzały, zanim w ogóle się rozpocznie, a w momencie pełnej instalacji jest już z pewnością nieaktualny. Nie oznacza to jednak, że nie należy dokonywać oceny projektów. Ocenianie lub uzasadnianie inwestycji w technologię informacyjną jest problematyczne. Inwestycje w technologię informacyjną są trudniejsze niż wiele innych decyzji inwestycyjnych, ponieważ koszty i korzyści trudno jest zidentyfikować i zmierzyć, a występujące czynniki niematerialne mogą mieć istotne znaczenie.¹¹⁰

Czasami wartość projektu informatycznego nie jest od razu znana w momencie wdrożenia systemu. Na przykład, celem projektu mającego na celu stworzenie witryny e-commerce powinno być generowanie zysków – a nie budowanie lub instalacja sprzętu, oprogramowania i stron internetowych na określonej platformie serwerowej. Technologia i jej późniejsze wdrożenie są jedynie środkiem do osiągnięcia celu. W związku z tym celem witryny e-commerce może być wygenerowanie 250 000 dolarów w ciągu sześciu miesięcy. W rezultacie ocena, czy projekt osiągnął swój cel, może nastąpić dopiero po wdrożeniu systemu.¹¹¹

Komitety oceniające w organizacjach często stają przed wyborem, w którym konkurencyjne technologie nie dają się łatwo ze sobą porównać. Wynika to z faktu, że dwa projekty mogą zaspokajać różne, a nawet konkurencyjne wymagania. Ostateczna pozycja w rankingu technologii zależy od tego, w jakim stopniu każda z nich odpowiada na poszczególne wymagania. Jednak poza przypadkami, w których jedna technologia wyraźnie dominuje

¹⁰⁷ C. Sauer, A. Gemino i B. H. Reich. „IT project performance: The impact of size and volatility”. W: *Communications of the ACM* (2007).

¹⁰⁸ Keil i in., „A framework for identifying software project risks.”

¹⁰⁹ Gemino, Reich i Sauer, „A Temporal Model of Information Technology Project Performance”.

¹¹⁰ P. Powell. „Information technology evaluation: is it different?” W: *Journal of the operational research society* (1992).

¹¹¹ Marchewka, *Information Technology Project management*.

Tabela 3.1. Kategoryzacja ryzyk według modelu czasowego

Proponowane kategorie czynników ryzyka			
Ryzyka a priori		Ryzyka wywołujące się	
Wiedza	Struktura	Wsparcie organizacyjne	Zmienność
Stopień nowości	Złożoność technologiczna	Brak zaangażowania użytkowników	Zmiany personalne
Brak doświadczenia z technologią	Rozmiar projektu	Ryzyka środowiska organizacyjnego	Ryzyka polityczne – np. konflikty między działami
Brak eksperyty, tempo przyrostu wiedzy (nowe oprogramowanie/sprzęt)	Ryzyka planowania	Ryzyka zarządzania zasobami	Zmiany technologiczne
Nieuzupełnienie wymagań	Ryzyka funkcjonalności systemu		Zmiana zakresu
Zarządzanie oczekiwaniami użytkowników			Ryzyka harmonogramowe

Źródło: Gemino, Reich i Sauer, „A Temporal Model of Information Technology Project Performance”

pozostałe, pojawia się problem, gdy dwie alternatywy zaspokajają odmienne i konkurencyjne potrzeby. Rozwiązanie takich konfliktów jest praktycznie niemożliwe bez udziału wiedzy eksperckiej w procesie podejmowania decyzji.¹¹²

Currie określa jednak uzasadnianie inwestycji w IT jako „rytuał legitymizacji”, a nie proces rzeczywistej oceny korzyści.¹¹³ Jak zauważył Williams już pół wieku temu: „W biznesie nie chodzi o to, by tworzyć dobre prognozy, lecz by sprawiać, aby się one spełniały”.¹¹⁴

3.8. Ostatnie trendy w projektach informatycznych

Ostatnie trendy, takie jak rosnąca globalizacja, outsourcing oraz zespoły wirtualne, stwarzają nowe wyzwania i możliwości dla kierowników projektów informatycznych oraz ich zespołów.¹¹⁵

3.8.1. Globalizacja

Globalizacja to termin powszechnie używany, lecz o niejasnym znaczeniu, nawet wśród tych, którzy się nim posługują.¹¹⁶ Anthony McGrew twierdzi, że globalizacja stanowi sieć powiązań i połączeń wykraczających poza państwa narodowe (a tym samym społeczeństwa), które tworzą współczesny system światowy. To proces, w ramach którego wydarzenia, decyzje i działania w jednej części świata mogą mieć znaczące konsekwencje dla jednostek i społeczności w odległych regionach globu.¹¹⁷

Globalizacja znacząco wpłynęła na sektor IT. Mimo że główne firmy technologiczne, takie jak Microsoft czy IBM, powstały w Stanach Zjednoczonych, większość ich działalności ma charakter globalny – przedsiębiorstwa i osoby z całego świata przyczyniają się do rozwoju technologii informacyjnych oraz współpracują przy różnych projektach IT.¹¹⁸

Ma to również znaczący wpływ na pracę kierowników projektów, gdyż muszą oni uwzględnić następujące zagadnienia:

¹¹² N. S. Thomaidis, N. Nikitakos i G. D. Dounias. „The evaluation of information technology projects: A fuzzy multicriteria decision-making approach”. W: *International Journal of Information Technology & Decision Making* (2006).

¹¹³ W. Currie. „The art of justifying new technology to top management”. W: *Omega* (1989).

¹¹⁴ B. R. Williams. *Technology, Investment and Growth*. London: Chapman & Hall, 1967.

¹¹⁵ Schwalbe, *Information Technology Project management*.

¹¹⁶ R. Robertson i K. E. White. „What is globalization?” W: *The Blackwell companion to globalization* (2007), s. 54–66.

¹¹⁷ A. McGrew. „A Global Society”. W: *Modernity and Its Futures*. Red. S. Hall, D. Held i A. McGrew. Cambridge: Polity Press, 1990.

¹¹⁸ Schwalbe, *Information Technology Project management*.

- **Komunikacja:** Praca w różnych strefach czasowych, różne języki, pochodzenie kulturowe i święta wymagają efektywnego planowania komunikacji.
- **Zaufanie:** Zaufanie jest kluczowe dla wszystkich zespołów, zwłaszcza globalnych. Należy je budować od początku, szanując różnice i doceniając wartości, jakie członkowie wnoszą do projektu.
- **Wspólne praktyki pracy:** Należy ujednolicić procesy pracy i wypracować praktyki akceptowane przez wszystkich.¹¹⁹

3.8.2. Outsourcing

Outsourcing oznacza pozyskiwanie dóbr i usług przez organizację z zewnętrznych źródeł. Termin offshoring odnosi się do outsourcingu realizowanego w innym kraju. Offshoring jest naturalną konsekwencją globalizacji. Projekty IT coraz częściej opierają się na outsourcingu, zarówno w granicach kraju, jak i poza nimi. Niektóre organizacje utrzymują konkurencyjność, strategicznie wykorzystując outsourcing. Wielu firmom udało się obniżyć koszty dzięki tej praktyce, mimo że bywa ona niepopularna w ich krajach macierzystych.¹²⁰ Słynnym przykładem outsourcingu jest decyzja Eastmana Kodaka o przekazaniu całego centrum danych firmie IBM, operacji mikrokomputerowych Businessland oraz sieci telekomunikacyjnych i danych Digital Equipment Corporation oraz IBM.

Outsourcing IT zależy od wielu czynników na różnych poziomach. Na poziomie gospodarczym trendy i cykle ekonomiczne mogą motywować firmy do racjonalizacji zarządzania infrastrukturą IT poprzez outsourcing. Na poziomie branżowym presja konkurencyjna skłania firmy do budowania „partnerskich relacji” z kluczowymi dostawcami IT. Na poziomie firmy poszukiwanie przewagi konkurencyjnej jest kluczowym impulsem do decyzji o outsourcingu. Wewnątrz firmy decyzja o outsourcingu może zależeć od czynników menedżerskich, takich jak dążenie do kontroli zasobów lub postrzeganie informacji jako źródła władzy.^{121 122 123}

W związku z rosnącym wykorzystaniem outsourcingu w projektach IT, kierownicy

¹¹⁹ Tamże.

¹²⁰ Tamże.

¹²¹ L. Loh i N. Venkatraman. „Determinants of information technology outsourcing: A cross-sectional analysis”. W: *Journal of management information systems* 9.1 (1992), s. 7–24.

¹²² R. M. Kanter. „Power Failure in Management Circuits”. W: *Harvard Business Review* 57.4 (1979), s. 65–75.

¹²³ H. Mintzberg. *Power In and Around Organizations*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

projektów muszą lepiej zrozumieć globalne i zamówieniowe aspekty, w tym pracę z zespołami zdalnymi.¹²⁴

3.8.3. Zespoły zdalne

Globalizacja projektów zwiększyła zapotrzebowanie na zespoły zdalne,¹²⁵ czyli grupy osób współpracujących pomimo granic czasowych i przestrzennych, wykorzystujące technologie komunikacyjne. Członkowie zespołu mogą pracować dla tej samej firmy w jednym kraju lub obejmować pracowników, niezależnych konsultantów, dostawców, a nawet wolontariuszy z całego świata.¹²⁶

W konkurencyjnym rynku zespoły zdalne stanowią coraz częstsze rozwiązanie odpowiadające na potrzebę skrócenia czasu wprowadzenia produktu na rynek, obniżenia kosztów oraz szybkiego rozwiązywania złożonych problemów organizacyjnych. Zespoły wirtualne umożliwiają organizacjom łączenie talentów i ekspertyzy pracowników oraz osób z zewnątrz poprzez eliminację barier czasowych i przestrzennych. Obecnie firmy intensywnie inwestują w zespoły zdalne, aby podnieść swoją wydajność i konkurencyjność.¹²⁷

Zalety zespołów wirtualnych

- **Obniżenie kosztów** – wielu pracowników zdalnych nie wymaga biur ani dodatkowego wsparcia.
- **Dostęp do ekspertyzy i elastyczność** – globalny zasięg umożliwia pracę przez całą dobę, zwiększając konkurencyjność.
- **Lepsza równowaga praca–życie** – brak sztywnych godzin pracy i dojazdów.

Wady zespołów wirtualnych

- **Izolacja członków** – trudności w adaptacji do środowiska zdalnego.
- **Problemy komunikacyjne** – brak języka ciała i komunikacji niewerbalnej utrudnia budowanie relacji.
- **Ograniczenie nieformalnej wymiany informacji** – trudności w nawiązywaniu kontaktów.
- **Zależność od technologii** – awarie mogą sparaliżować pracę.

¹²⁴ Schwalbe, *Information Technology Project management*.

¹²⁵ Institute, *PMBOK Guide – Sixth Edition*.

¹²⁶ Schwalbe, *Information Technology Project management*.

¹²⁷ N. A. Ebrahim, S. Ahmed i Z. Taha. „Virtual teams: A literature review”. W: *Australian journal of basic and applied sciences* 3.3 (2009), s. 2653–2669.

4. Kierownik projektu

Jako rozwinięcie części teoretycznej zarządzania projektami kluczowym jest zrozumienie roli oraz zadań kierownika projektu. Dodatkowo w literaturze przedmiotu jest dostępnych wiele badań oraz prac omawiających cechy oraz kompetencje kierownika projektu. Warto przytoczyć te teksty w celu udzielenia trafniejszej odpowiedzi na postawione w pracy pytanie.

4.1. Kierownik projektu

Najprostsza definicja za pomocą, której można wyjaśnić kim jest kierownik projektu to: osoba, która wyznacza proces dostarczenia zmiany i jest odpowiedzialna za zarządzanie tym procesem.¹²⁸ Bardziej powszechne wydaje się jednak rozumienie kierownika jako podmiotu odpowiedzialnego za realizację projektu w sposób bezpieczny, zgodny z harmonogramem i budżetem oraz osiągający wymagane standardy wydajności lub jakości określone przez klienta.¹²⁹ Nicholas i Steyn podają w swojej książce jednak bardziej metaforyczne rozumienie kierownika projektu jako swoistego spoiwa scalającego wszystkie elementy projektu oraz napędzającego jego rozwój. W ramach tej funkcji pełnione są liczne zadania, między innymi integracja, komunikacja, podejmowanie decyzji, motywowanie, propagowanie idei, działalność przedsiębiorcza oraz inicjowanie zmian.¹³⁰ Z taką definicją kierownika projektu zgadza się również Blackburn i Sarah. W swoim artykule określili kierownika projektu jako byt łączący sieć aktorów w celu realizacji wspólnego celu¹³¹

4.2. Role i zadania kierownika projektu

Zgodnie z przytoczonymi definicjami kierownika projektu, jego głównym zadaniem jest zapewnienie dostarczenia końcowego produktu w ramach przyjętego budżetu i ustalonych terminów, zgodnie z określonymi wymaganiami technicznymi. Pozostałe obowiązki ulegają modyfikacji w zależności od umiejętności kierownika, etapu realizacji przedsięwzięcia, rozmiaru i charakteru projektu oraz zakresu odpowiedzialności przekazanej przez wyższe kierownictwo. Mimo zróżnicowania kompetencji, zakres obowiązków zazwyczaj obejmuje:

¹²⁸ Turner, *Gower Handbook of Project Management*.

¹²⁹ J. Sommerville, N. Craig i J. Hendry. „The role of the project manager: All things to all people?” W: *Structural Survey* (2010). DOI: 10.1108/02630801011044235.

¹³⁰ J. M. Nicholas i H. Steyn. *Project Management for Business, Engineering, and Technology*. Elsevier, 2012.

¹³¹ S. Blackburn. „The project manager and the project-network”. W: *International Journal of Project Management* (2002). DOI: 10.1016/S0263-7863(01)00069-2.

- Opracowanie planu działań projektowych, zadań oraz końcowych rezultatów, co obejmuje sporządzenie struktury podziału pracy, ustalenie harmonogramu i budżetu, a także koordynację zadań oraz dystrybucję zasobów. Polega to na łączeniu różnorodnych działań i rozproszonych elementów w sposób umożliwiający osiągnięcie wyznaczonych celów czasowych, budżetowych i jakościowych. Centralną funkcją kierownika projektu jest zapewnienie spójności wszystkich składników przedsięwzięcia, co umożliwia ich harmonijną współpracę zgodnie z ustalonym planem.
- Dobór oraz organizacja zespołu odpowiedzialnego za realizację przedsięwzięcia. Nawet przy ograniczonych formalnych uprawnieniach do podejmowania decyzji kadrowych, rola ta umożliwia wywieranie znaczącego wpływu na dokonywane wybory osób dysponujących odpowiednimi kompetencjami.
- Nawiązywanie kontaktów z interesariuszami oraz oddziaływanie na ich postawy. Na styku licznych kanałów informacyjnych znajduje się rola kierownika, odpowiedzialnego za zbieranie oraz dystrybucję danych – obejmujących raporty, zgłoszenia, notatki służbowe czy skargi. Przekazywane informacje są starannie opracowywane i poddawane analizie, aby wszyscy interesariusze byli rzetelnie poinformowani o założeniach dotyczących polityki, celów, budżetów, harmonogramów, wymagań, postępów prac oraz zmian w projekcie.
- Prowadzenie negocjacji oraz scalanie działań kierowników działów, wykonawców, użytkowników i kadry zarządzającej. Istotnym aspektem działalności kierownika jest również propagowanie idei projektu, co wiąże się z kreowaniem przekonania o jego wartości i wykonalności. W początkowej fazie koncepcyjnej to właśnie kierownik często dysponuje pełnym obrazem przedsięwzięcia, a jego zdolność do zdobycia poparcia kluczowych interesariuszy może przesądzić o uzyskaniu niezbędnego finansowania
- Zapewnienie kanału komunikacji z klientem. Znaczenie tej komunikacji podkreślają w swojej pracy Turner i Müller.¹³² Przypominają oni, że kierownik projektu działa tak naprawdę w imieniu i na rzecz klienta, a więc projekt powinien być zgodny z jego potrzebami.
- Nadzór nad postępem realizacji projektu. Ocena rzeczywistych osiągnięć technicznych w porównaniu z planowanymi, przegląd i weryfikacja zasadności celów technicznych,

¹³² R. Turner i R. Müller. „Communication and Co-operation on Projects Between the Project Owner As Principal and the Project Manager as Agent”. W: *European Management Journal* (2004). doi: 10.1016/j.emj.2004.04.010.

potwierdzenie ciągłej potrzeby realizacji projektu, nadzorowanie wydatków na zasoby oraz porównanie przewidywanej wartości z poniesionymi kosztami.¹³³¹³⁴

- Identyfikacja problemów o podłożu technicznym i funkcjonalnym a następnie bezpośrednie ich rozwiązywanie lub poszukiwanie adekwatnego wsparcia. Zarządzanie ryzykiem jest integralną częścią zarządzania projektem, obejmującą identyfikację, ocenę i planowanie reakcji na ryzyka w celu minimalizacji ich wpływu. W większych projektach, gdzie ryzyko i konsekwencje są znaczące, skuteczne zarządzanie ryzykiem może decydować o sukcesie lub porażce przedsięwzięcia.
- Radzenie sobie z sytuacjami kryzysowymi oraz rozwiązywanie pojawiających się konfliktów. Determinacja i skupienie zespołu na wspólnym celu stanowią jeden z kluczowych czynników mobilizujących do działania. W strukturze organizacyjnej projektu to właśnie kierownik odpowiada za kreowanie poczucia wspólnego kierunku i zaangażowania. Pomimo występowania licznych czynników sprzyjających motywacji, takich jak spontaniczność, osiągnięcia czy entuzjazm, utrzymanie wysokiego poziomu zaangażowania bywa trudne w długotrwałych i wymagających przedsięwzięciach. Czynniki takie jak brak wcześniejszych wzorców, praca na niepełny etat, różnorodność specjalizacji, sporadyczność kontaktów oraz dystans przestrzenny mogą znacząco wpływać na spadek motywacji. Skuteczne zarządzanie projektem wymaga jednak zdolności do budowania entuzjazmu, ducha zespołu, pewności siebie oraz dążenia do doskonałości.
- Sugerowanie zakończenia projektu lub zmiany kierunku działań, gdy założone cele okazują się nieosiągalne. Centralne położenie kierownika umożliwia podejmowanie decyzji związanych z alokacją zasobów, definiowaniem zakresu działań oraz równoważeniem kryteriów harmonogramu, kosztów i jakości.¹³⁵

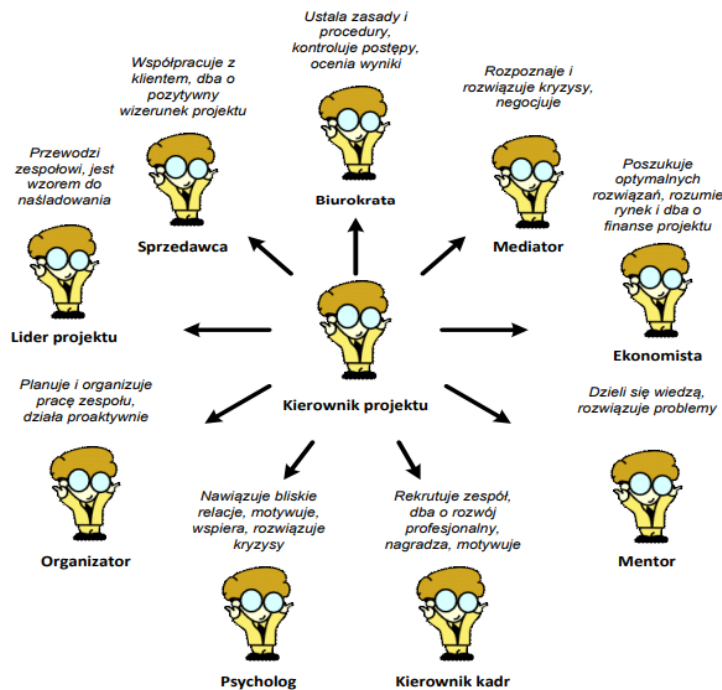
Ciekawa interpretacja roli oraz zadań kierownika projektu została również przedstawiona na rysunku 3.1. Autorzy zwracają większą uwagę na rolę interpersonalną kierownika projektu i określają go w tym kontekście jako mentora, mediatora, psychologa i lidera.

¹³³ D. Roman. *Science, Technology, and Innovation: A Systems Approach*. Grid Publishing, 1980.

¹³⁴ R. Vaupel, G. Schmolke i A. Kruger. *Customer-focused management by projects*. MacMillan Publisher, 2000.

¹³⁵ Nicholas i Steyn, *Project Management for Business, Engineering, and Technology*.

Rysunek 4.1. Role kierownika projektu



Źródło: Wykład *Podstawy Zarządzania Projektami*, M. Juchniewicz

4.3. Kompetencje kierownika projektu

Dywagacje na temat kompetencji, jakie powinien posiadać kierownik projektu, stały się zagadnieniem często poruszonym w literaturze przedmiotu. Jednak nie tylko wiedza i umiejętności techniczne mają znaczenie. Menedżerowie są bardziej skłonni do osiągnięcia lepszych wyników lub pozostawania dłużej na swoim stanowisku, jeśli ich cechy osobiste odpowiadają wymaganiom stanowiska.¹³⁶ Na rysunkach 3.2. oraz 3.3. zostały przedstawione wyniki dostępnych badań. Na podstawie analizy dostępnej literatury i badań w dalszych sekcjach według ważności zostały wymienione kluczowe kompetencje:¹³⁷¹³⁸¹³⁹¹⁴⁰¹⁴¹¹⁴²

¹³⁶ M. D. Mumford i in. „Leadership skills for a changing world: Solving complex social problems”. W: *The Leadership Quarterly* (2000). doi: [https://doi.org/10.1016/S1048-9843\(99\)00041-7](https://doi.org/10.1016/S1048-9843(99)00041-7).

¹³⁷ M. Chipulu i in. „A Multidimensional Analysis of Project Manager Competences”. W: *IEEE Transactions on Engineering Management* (2012). doi: 10.1109/TEM.2012.2215330.

¹³⁸ J. C. Alvarenga i in. „The project manager core competencies to project success”. W: *International Journal of Managing Projects in Business* (2019). doi: 10.1108/IJMPB-12-2018-0274.

¹³⁹ People, *Project Management Benchmark Report*.

¹⁴⁰ P. Ziek i J. D. Anderson. „Communication, dialogue and project management”. W: *International Journal of Managing Projects in Business* (2015). doi: 10.1108/IJMPB-04-2014-0034.

¹⁴¹ J. M. Brill, M. J. Bishop i A. E. Walker. „The Competencies and Characteristics Required of an Effective Project Manager: A Web-Based Delphi Study”. W: *ITLS Faculty Publications* (2006). doi: 10.1007/s11423-006-8251-y.

¹⁴² Arras People. *Project Management Benchmark Report*. 2015.

Tabela 4.1. Kompetencje kierowników według raportu Arras

Ogólne	Menedżerowie Programów	Menedżerowie Zmian	Menedżerowie Projektów
Planowanie i Organizacja	Przywództwo	Planowanie i Organizacja	Planowanie i Organizacja
Umiejętności komunikacyjne	Planowanie i Organizacja	Umiejętności komunikacyjne	Umiejętności komunikacyjne
Przywództwo	Zarządzanie relacjami	Elastyczność	Zarządzanie relacjami
Zarządzanie relacjami	Umiejętności komunikacyjne	Zarządzanie relacjami	Elastyczność
Elastyczność	Świadomość biznesowa	Przywództwo	Przywództwo
Proaktywność/Inicjatywa	Wpływanie	Wpływanie	Umiejętności techniczne
Umiejętności techniczne	Elastyczność	Proaktywność/Inicjatywa	Podjęmowanie decyzji
Wpływanie	Podjęmowanie decyzji	Umiejętności techniczne	Praca zespołowa
Podjęmowanie decyzji	Proaktywność/Inicjatywa	Podjęmowanie decyzji	Proaktywność/Inicjatywa
Praca zespołowa	Umiejętności techniczne	Świadomość organizacyjna	Wpływanie
Świadomość biznesowa	Praca zespołowa	Świadomość biznesowa	Świadomość biznesowa
Dbłość o dokładność	Świadomość organizacyjna	Praca zespołowa	Dbłość o dokładność
Świadomość organizacyjna	Dbłość o dokładność	Dbłość o dokładność	Świadomość organizacyjna

Źródło: Arras People Project Management Benchmark Report 2010

Tabela 4.2. Analiza częstotliwości występowania kompetencji kierowników projektów

Poz.	Kompetencja	Liczba	% wystą- pień	Opis/Przykład
1	Wiedza specjalistyczna branżowa	1288	56%	Wymaga umiejętności technicznych właściwych dla danej branży (np. programowanie, znajomość języka, kwalifikacje inżynierskie, wiedza księgowa, itp.)
2	Umiejętności komunikacyjne	1078	47%	Niezbędne umiejętności komunikacyjne
3	Umiejętności zarządzania zespołem	907	39%	Zarządzanie zespołem
4	Zdolności przywódcze	694	30%	Potrafi prowadzić zespół
5	Wykształcenie wyższe	682	30%	Wykształcenie wyższe
6	Zarządzanie interesariuszami	656	28%	Zarządzanie oczekiwaniami interesariuszy
7	Zarządzanie budżetem	654	28%	Zarządzanie budżetem projektu zgodnie z celami
8	Zarządzanie harmonogramem	653	28%	Zapewnienie realizacji projektu zgodnie z harmonogramem
9	Świadomość biznesowa	488	21%	Zorientowanie na biznes/przedsiębiorczość
10	Praca zespołowa	458	20%	Praca w zespole w strukturze macierzowej

Zróżdło: A Multidimensional Analysis of ProjectManager Competences Maxwell Chipulu, Jun Guan Neoh, Udechukwu (Udi) Ojiako, and Terry Williams 2013

4.3.1. Komunikacja

W najnowszych badaniach oraz pracach kluczowe okazują się umiejętności interpersonalne.¹⁴³ Umiejętności komunikacyjne są kluczowe w różnych kontekstach: negocjacje, komunikacja, relacje z klientami i zarządzanie konfliktem. Literatura opisuje komunikację jako podstawowe narzędzie dla kierownika projektu w jego kluczowej funkcji łączenia i pośredniczenia między wieloma podmiotami zaangażowanymi w projekt: zespołem, sponsorami, sprzedawcami, kierownikami funkcjonalnymi, użytkownikami i klientami oraz innymi interesariuszami. Wyraźnie wynika z tego, że to właśnie kompetencje komunikacyjne pozwalają kierownikom projektów być skutecznymi i mieć wpływ na sukces projektu.¹⁴⁴

¹⁴³ Brill, Bishop i Walker, „The Competencies and Characteristics Required of an Effective Project Manager: A Web-Based Delphi Study”.

¹⁴⁴ Alvarenga i in., „The project manager core competencies to project success”.

Ziek oraz Anderson w swojej pracy podkreślają, że rola komunikacji w projekcie to nie jest jedynie czynnik wpływający na sukces bądź porażkę projektu. Uważają oni, że jest to fundament zarządzania projektami. Zrozumienie roli komunikacji zapewnia kierownikom projektów dodatkowe środki kontroli nad projektem, wykraczające poza budżety i harmonogramy.¹⁴⁵

4.3.2. Przywództwo

Podążając za Burnsem, przywództwo jest „jednym z najczęściej obserwowanych i najmniej rozumianych zjawisk na ziemi”.¹⁴⁶ W związku ze skomplikowaną naturą tej cechy nie jest ona łatwa w wyjaśnieniu. Tabassi i Bakar zidentyfikowali kluczowe komponenty związane ze zjawiskiem przywództwa i zdefiniowali je jako proces, w którym lider ze swoją inteligencją i siłą woli ma wpływ na grupę podwładnych, aby móc rozwijać ich potencjały w celu osiągnięcia celów organizacyjnych w wyznaczonym czasie, finansowaniu i jakości.¹⁴⁷

Kierownik projektu nie może ograniczać się jedynie do zarządzania harmonogramem i zadaniami – kluczowe stają się rozwinięte kompetencje przywódcze, które są fundamentem sukcesu całego przedsięwzięcia. W projektach, w których nieuniknione są kryzysy i konflikty, zdolność szybkiego podejmowania decyzji oraz umiejętność skutecznego komunikowania się z zespołem nabierają szczególnego znaczenia. Nicholas podkreśla, że to właśnie przywództwo jest jedną z czterech funkcji kierownika, który ukierunkowuje pracę swoich pracowników oraz motywuje ich do realizacji wspólnego celu.¹⁴⁸ W badaniu Alvarenga cechy przywódcze zostały uznane za najważniejszą kompetencję u kierownika projektu.¹⁴⁹

4.3.3. Zarządzanie sobą (ang. Self-management)

Kompetencje zarządzania sobą obejmują wizję, poznanie, odporność emocjonalną, samoświadomość. Odzwierciedlają one potrzebę kierowników projektów, którzy są oddani samorozwojowi i ciągłemu uczeniu się. Odporność emocjonalna jest również bardzo istotna dla kierowników projektów: ciągła presja, niepewność, ryzyko, konflikty zespołowe i inne codzienne wyzwania mogą prowadzić do stresu, a w skrajnych sytuacjach do wypalenia

¹⁴⁵ Ziek i Anderson, „Communication, dialogue and project management”.

¹⁴⁶ J. M. Burns. *Leadership*. Harper & Row. 1978.

¹⁴⁷ A. A. Tabassi i in. „Leadership Competences of Sustainable Construction Project Managers”. W: *Journal of Cleaner Production* (2016). doi: 10.1016/j.jclepro.2016.02.076.

¹⁴⁸ Nicholas i Steyn, *Project Management for Business, Engineering, and Technology*.

¹⁴⁹ Alvarenga i in., „The project manager core competencies to project success”.

zawodowego.¹⁵⁰ Samoświadomość oraz odporność emocjonalna są również podkreślane jako kluczowe cechy kierownika projektu w pracy Turnera i Mullera. Według ich badań samoświadomość jest zwłaszcza szczególnie ważna w projektach o średniej złożoności.¹⁵¹

4.3.4. Umiejętności zarządcze

Kierownik projektu musi zdecydować, jak przydzielić zasoby ludzkie, finansowe i informacyjne do różnych zadań projektu. Kompetencja ta opiera się na planowaniu, organizowaniu, koordynowaniu i kontrolowaniu zadań.¹⁵² Znaczenie tych umiejętności podkreślają również w swojej pracy Udo i Koppensteiner. Według nich kierownik projektu musi znać podstawy procesów zarządzania projektami, metodologie, narzędzia i techniki oraz potrafić dostosować je do organizacji. Jednym ze sposobów na sprawdzenie tych kompetencji jest certyfikacja bądź odpowiednie studia.¹⁵³ Odpowiednie wykształcenie pojawiło się również w wynikach badań IEEE, aż 30% respondentów uznało stopień naukowy z dziedziny zarządzania za istotną cechę kierownika projektu.¹⁵⁴

4.3.5. Umiejętności techniczne

Kolejną kompetencją kierownika projektu, która często pojawia się w literaturze, są umiejętności techniczne rozumiane jako wiedza domenowa oraz znajomość narzędzi potrzebnych w zarządzaniu projektami.¹⁵⁵¹⁵⁶ Znajomość domeny i wiedzy specyficznej dla branży jest cenna, gdyż to kierownik projektu identyfikuje potrzeby użytkowników i opracowuje rozwiązania, które zmieniają sytuacje biznesowe. Główną odpowiedzialnością kierownika projektu jest zapewnienie, że szybko planowane zmiany są zrozumiane, zaplanowane, wdrożone i strategicznie wykorzystane w organizacji.¹⁵⁷ Nie byłby on w stanie tego osiągnąć bez dobrego zrozumienia projektu oraz działań wykonywanych w ramach projektu. Ciekawym opisem poziomu kompetencji technicznych kierownika projektu jest podejście zaproponowane przez

¹⁵⁰ Alvarenga i in., „The project manager core competencies to project success”.

¹⁵¹ R. Turner i R. Müller, „Leadership competency profiles of successful project managers”. W: *International Journal of Project Management* (2010). DOI: 10.1016/j.ijproman.2009.09.003.

¹⁵² P. Gottschalk i J. T. Karlsen, „A comparison of leadership roles in internal IT projects versus outsourcing projects.” W: *Industrial Management & Data Systems* (2005). DOI: 10.1108/026355705106332203.

¹⁵³ N. Udo i S. Koppensteiner, „What are the core competencies of a successful project manager? Paper presented at PMI® Global Congress 2004—EMEA”. W: *Project Management Institute* (2004).

¹⁵⁴ Chipulu i in., „A Multidimensional Analysis of Project Manager Competences”.

¹⁵⁵ People, *Project Management Benchmark Report*.

¹⁵⁶ Brill, Bishop i Walker, „The Competencies and Characteristics Required of an Effective Project Manager: A Web-Based Delphi Study”.

¹⁵⁷ Gottschalk i Karlsen, „A comparison of leadership roles in internal IT projects versus outsourcing projects.”

Nicholasa. Tłumaczy on, że kompetencje kierownika powinny pokrywać cały zakres projektu, ale ograniczając go jedynie do pierwszego poziomu struktury podziału pracy (ang. Work breakdown structure, WBS).¹⁵⁸

4.4. Znaczenie wiedzy z zakresu inżynierii oprogramowania dla kierownika projektu

Aby podejmować świadome decyzje, kierownicy projektów muszą rozumieć aspekty techniczne projektu. Ich kompetencje powinny obejmować pełny zakres projektu. W środowiskach nisko- lub nietechnologicznych tę znajomość można rozwijać poprzez doświadczenie i szkolenia nieformalne. W projektach o wysokim zaawansowaniu technologicznym wymagania są surowsze i zwykle obejmują karierę ukształtowaną w środowisku technologicznym oraz wiedzę z zakresu nauki lub inżynierii.

Choć kierownicy projektów rzadko przeprowadzają analizy techniczne, muszą być wykwalifikowani do podejmowania ocen technicznych oraz zdolni do analizy i integracji pracy oraz wiedzy. Wielu specjalistów technicznych nie radzi sobie dobrze ze współpracą z innymi, ponieważ większość kształcenia w dziedzinie inżynierii i technologii kładzie nacisk na analizę, pomijając umiejętności pracy zespołowej. Aby skutecznie komunikować się ze wszystkimi uczestnikami projektu i integrować ich pracę, kierownik projektu musi rozumieć i posługiwać się językiem specjalistów technicznych.¹⁵⁹

Z takim postrzeganiem polemizuje jednak badanie przeprowadzone wśród brazylijskich kierowników projektów informatycznych.¹⁶⁰ Było to jakościowe badanie eksploracyjne z udziałem 16 brazylijskich profesjonalistów IT z różnych sektorów. Wbrew tradycyjnemu postrzeganiu w środowisku projektów informatycznych, umiejętności z zakresu inżynierii oprogramowania uznano za mniej istotne niż kompetencje behawioralne, biznesowe i menedżerskie. Analiza wykazała, że najważniejsze kompetencje to:

- zarządzanie zespołem,
- wiedza domenowa,
- komunikacja,
- zarządzanie projektami,
- umiejętności interpersonalne.

¹⁵⁸ Nicholas i Steyn, *Project Management for Business, Engineering, and Technology*.

¹⁵⁹ Tamże.

¹⁶⁰ C. C. Silva de Araújo i C. D. Pedron. „IT project manager competencies and IT project success: a qualitative study.” W: *Organisational Project Management* 2.1 (2015).

Kategorie takie jak zarządzanie zespołem, wiedza domenowa, umiejętności interpersonalne i komunikacja zebrały łącznie 349 wzmianek, podczas gdy umiejętności z zakresu inżynierii oprogramowania – jedynie 5. Dla respondentów kompetencje związane z relacjami i biznesem są więc kluczowe dla sukcesu projektu informatycznego. Niektórzy zauważyli nawet, że kierownicy z silnym profilem technicznym mogą bagatelizować inne aspekty zarządzania, takie jak strategia organizacyjna, umiejętności polityczne czy zarządzanie ludźmi. W badaniu została przytoczona w tym kontekście wypowiedź respondentów:

- "Był bardziej techniczny, brakowało mu wiedzy politycznej. Brakowało mu politycznej siły, by projekt działał."
- "Czasami kierownik z dobrymi umiejętnościami technicznymi nie skupia się na ludziach."

Badanie jednak nie bagatelizuje całkowicie roli umiejętności z zakresu inżynierii oprogramowania, uznaje je za "niezbędne, ale niewystarczające". Niektórzy z respondentów badania podkreślali, że wiedza techniczna ułatwia komunikację z programistami oraz usprawnia zarządzanie projektem. Przytoczono w tym kontekście następujące wypowiedzi.

- "Gdy masz umiejętności techniczne, możesz porozumieć się z osobą rozwijającą system i przekazać programiście oczekiwania klienta."
- "Jeśli kierownik nie ma wiedzy technicznej, musi stale polegać na kimś, kto ją posiada."

Zgromadzone dane wskazują, że umiejętności z zakresu inżynierii oprogramowania nie wystarczą, aby zapewnić pozytywne rezultaty w zarządzaniu projektami. Nie oznacza to, że kierownicy projektów IT powinni porzucić kompetencje techniczne wypracowane w trakcie kariery. Sugeruje to jednak, że powinni łączyć umiejętności techniczne z kompetencjami interpersonalnymi i menedżerskimi, aby skuteczniej osiągać sukces projektu.¹⁶²

Podobne badania przeprowadził Stevenson wśród osób odpowiedzialnych za rekrutację kierowników projektów informatycznych.¹⁶³ Badanie to wykazało niską ocenę ekspertyzy technicznej jako kompetencji kierowników projektów. Biorąc pod uwagę zmiany i konsolidację ról w zespołach projektowych oraz trend wzrostowy zatrudniania kierowników o profilu technicznym w ciągu ostatniej dekady, oczekiwano, że umiejętności techniczne zostaną

¹⁶¹ Silva de Araújo i Pedron, „IT project manager competencies and IT project success: a qualitative study.”

¹⁶² Tamże.

¹⁶³ D. H. Stevenson i J. A. Starkweather, „PM critical competency index: IT execs prefer soft skills”. W: *International journal of project management* 28.7 (2010), s. 663–671.

uznane za istotne kryterium. Jedną z interpretacji tych wyników jest to, że firmy mniej cenią kompetencje techniczne kierowników projektów zdobyte w innych firmach, a bardziej skupiają się na szkoleniu swoich pracowników wewnątrz, w oparciu o własną technologię. Inna interpretacja zakłada, że respondenci uważali, iż podstawowy poziom wiedzy technicznej jest inherentny u wszystkich kandydatów, więc nie traktowali go jako kryterium różnicującego talenty.

Wyniki badania jednoznacznie wskazują, że kompetencje „miękkie” takie jak przywództwo, umiejętność komunikacji na wielu poziomach, zdolności werbalne i pisemne, postawa oraz zdolność radzenia sobie z niejednoznacznością i zmianami – są kluczowymi cechami, które są poszukiwane wśród kierowników projektów informatycznych i które decydują o skutecznym zarządzaniu projektami.¹⁶⁴

Ważność umiejętności technicznych a menedżerskich zależy od typu projektu. W projektach innowacyjnych kierownik musi mieć wyższe kompetencje techniczne ze względu na złożoność problemów i techniczne nastawienie zespołu. W projektach rozwojowych lub nietechnicznych kluczowe są zdolności menedżerskie ze względu na współpracę z wieloma obszarami funkcjonalnymi. Ogólnie rzecz biorąc, kierownicy muszą mieć wystarczające umiejętności techniczne, by rozumieć problem, choć nadmierne skupienie na umiejętnościach technicznych może prowadzić do zaniedbywania roli menedżerskiej. Nie ma jednak substytutu dla silnych kompetencji zarządczych w tej roli.

Aby sprostać wymaganiom zarówno technicznym, jak i administracyjnym, w projektach czasem zatrudnia się dwóch menedżerów: technicznego i administracyjnego. Często występuje to w projektach budowlanych, gdzie architekt odpowiada za kwestie techniczne, a tzw. kierownik projektu zajmuje się „dokumentacją administracyjną”. Obecność dwóch menedżerów komplikuje koordynację, komunikację i kwestie władzy, ponieważ obaj dzielą odpowiedzialność. Co więcej, gdy kierownik projektu jest podporządkowany architektowi, jego zdolność do zarządzania projektem jest ograniczona. Podobny podział występuje w branży filmowej: producent zarządza zasobami, harmonogramami i budżetami (pełniąc rolę kierownika projektu), podczas gdy reżyser nadzoruje kwestie techniczno-artystyczne. Ponieważ tworzenie filmu to przedsięwzięcie artystyczne, reżyserzy potrzebują elastyczności w budżetach i harmonogramach zdjęć, ale koszty również mają znaczenie – producent staje przed dylematem: „jaką cenę zapłacić za kreatywność?”.

¹⁶⁴ Tamże.

4. Kierownik projektu

Idealnie, projekt powinien mieć jednego kierownika, a wszyscy inni pełniący role menedżerskie lub administracyjne podlegają jemu.¹⁶⁵

¹⁶⁵ Nicholas i Steyn, *Project Management for Business, Engineering, and Technology*.

5. Badanie ankietowe wśród pracowników branży IT

By móc udzielić trafniejszej odpowiedzi na pytanie postawione w tytule pracy, analiza literatury została dodatkowo wsparta badaniem ankietowym przeprowadzonym wśród pracowników branży informatycznej.

5.1. Cel badania

Celem badania było zidentyfikowanie kluczowych kompetencji, które powinien posiadać skuteczny kierownik projektu informatycznego na podstawie dotychczasowych doświadczeń pracowników branży informatycznej. Dodatkowym aspektem tego badania w porównaniu do innych dostępnych prac było szczególne zwrócenie uwagi na umiejętności z zakresu inżynierii oprogramowania. Co pozwoliło na bardziej dokładne określenie wartości poszczególnych kompetencji technicznych i na jakim poziomie powinny być one posiadane przez kierownika projektu informatycznego. Poruszona została również kwestia szans i zagrożeń wynikających z posiadania przez kierownika projektu wiedzy technicznej.

5.2. Metodologia oraz opis badania

Badanie zostało przeprowadzone w formie dobrowolnej i anonimowej ankiety internetowej. Dostęp do ankiety był ograniczony, każdy z respondentów był indywidualnie zapraszany do udziału w badaniu. W badaniu wzięli udział pracownicy różnych przedsiębiorstw i poziomów organizacyjnych, co pozwoliło na uzyskanie szerokiego obrazu.

Ankieta została podzielona na trzy sekcje. Pierwsza część dotyczyła ogólnych informacji o respondentach, takich jak ich doświadczenie zawodowe, stanowisko oraz rodzaj projektów, w których uczestniczyli. Druga część skupiła się na ocenie kompetencji kierownika projektu w różnych obszarach, takich jak umiejętności interpersonalne, przywódcze, techniczne i analityczne. W tej części ankietowani wartościowali znaczenia różnych kompetencji kierownika projektu w skali Likerta (1–5), gdzie 1 oznacza brak znaczenia, a 5 oznacza bardzo wysokie znaczenie. Ostatnia część ankiety skupiona była na aspekcie umiejętności z zakresu inżynierii oprogramowania. Oprócz pytań zamkniętych bezpośrednio nawiązujących do pytania zadane w tytule pracy, respondenci mieli możliwość na bardziej szczegółowe wyrażenie opinii w formie pytań otwartych. Ankieta została skonstruowana w taki sposób, aby umożliwić respondentom ocenę zarówno kompetencji technicznych, jak i miękkich. W szczególności

Rysunek 5.1. Czas pracy w branży IT



Źródło: Opracowanie własne

zwrócono uwagę na umiejętności związane z inżynierią oprogramowania, które są kluczowe w kontekście zarządzania projektami informatycznymi. Badanie zostało zrealizowane w formie anonimowej ankiety internetowej. Wzięło w nim udział 30 respondentów reprezentujących różne stanowiska w sektorze IT. Kwestionariusz zawierał pytania zamknięte, a także pytania otwarte, umożliwiające rozszerzenie odpowiedzi o opinie jakościowe.

5.3. Charakterystyka próby

Wśród respondentów dominowali programiści (20 osób), następnie kierownicy projektów (5 osób) oraz inne role (3 osoby) i testerzy oprogramowania (2 osoby). Większość ankietowanych jest pracownikami branży informatycznej ponad 5 lat, dzięki czemu ich odpowiedzi są wsparte dużą wiedzą empiryczną. Dodatkowo w badaniu wzięli udział pracownicy mający doświadczenie w różnych typach projektów, co pozwoliło na uzyskanie szerokiego obrazu sytuacji. Co trzeci ankietowany miał również doświadczenie na stanowisku kierowniczym, dzięki czemu posiadają oni lepsze zrozumienie zarządzania zespołem i pracą.

5.4. Wyniki badania

5.4.1. Porównanie średnich ocen kompetencji

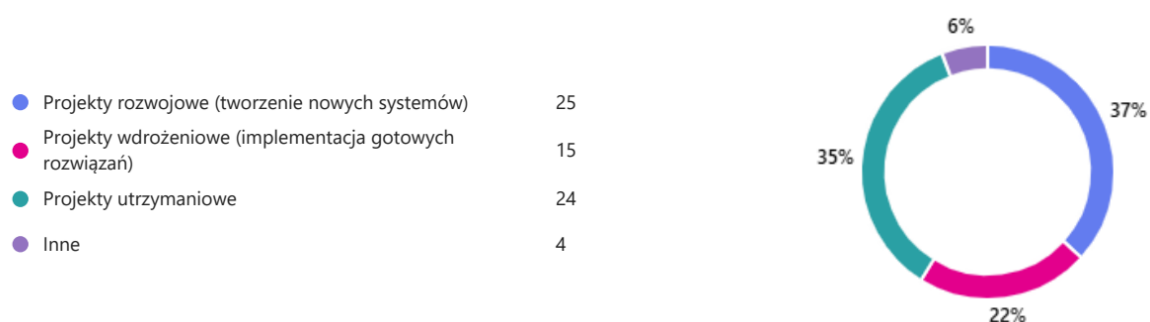
Wyniki badania pokazują, że Kategorie kompetencji interpersonalnych i przywódczych uzyskały najwyższe oceny, co potwierdza ich kluczowe znaczenie w zarządzaniu projektami IT. Dodatkowo warto zwrócić uwagę, że pracownicy z większym doświadczeniem w branży ocenili te kompetencje jeszcze wyżej niż ogół. Pojedynczo najważniejszą kompetencją okazały

Tabela 5.1. Średnie oceny kompetencji kierownika projektu IT

Kategoria	Średnia / Średnia 10+
Kompetencje interpersonalne i komunikacyjne	
Umiejętności komunikacyjne	4,67 / 5,00
Empatia i umiejętność słuchania	4,23 / 4,20
Negocjacje i rozwiązywanie konfliktów	4,17 / 4,50
Średnia kategorii	4,36 / 4,57
Kompetencje przywódcze i zarządcze	
Przywództwo i motywowanie	3,97 / 4,40
Zarządzanie zespołem i delegowanie	4,23 / 4,50
Zarządzanie czasem i organizacja pracy	4,43 / 4,80
Średnia kategorii	4,21 / 4,57
Kompetencje strategiczne i analityczne	
Myślenie strategiczne	4,17 / 4,10
Zarządzanie ryzykiem	4,17 / 4,20
Zdolność do podejmowania decyzji	4,4 / 4,50
Znajomość i rozumienie domeny biznesowej	4,07 / 4,20
Średnia kategorii	4,21 / 4,25
Kompetencje finansowe i organizacyjne	
Zarządzanie budżetem i zasobami	3,6 / 4,20
Orientacja na wyniki i efektywność	3,77 / 4,10
Wiedza teoretyczna z zakresu zarządzania projektami	3,1 / 3,50
Średnia kategorii	3,49 / 3,93
Kompetencje personalne	
Odporność na stres	3,83 / 4,20
Cierpliwość	3,9 / 4,20
Chęć rozwoju i uczenia się	3,3 / 3,90
Średnia kategorii	3,68 / 4,10
Kompetencje z zakresu inżynierii oprogramowania	
Formułowanie wymagań dotyczących oprogramowania	3,57 / 3,30
Metody testowania i weryfikacji poprawności	2,73 / 2,50
Znajomość cyklu życia systemów informatycznych	2,5 / 3,90
Znajomość wzorców projektowych i architektonicznych	2,15 / 2,40
Umiejętność programowania	2,17 / 2,00
Wiedza w zakresie baz danych	2,27 / 1,90
Wiedza w zakresie algorytmów i struktur danych	1,93 / 2,00
Znajomość systemów operacyjnych, sieci komputerowych i bezpieczeństwa	2,17 / 2,30
Średnia kategorii	2,67 / 2,54

Średnia – średnia ocen dla wszystkich badanych; Średnia 10+ – średnia ocen dla pracowników z stażem 10 lat i więcej. Źródło: Opracowanie własne.

Rysunek 5.2. Charakter realizowanych projektów



Źródło: Opracowanie własne

się umiejętności komunikacyjne. Wśród pracowników ze stażem 10+ otrzymały one średnią 5.00. Następnie w kolejności z pojedynczych umiejętności znalazło się zarządzanie czasem i organizacja pracy oraz delegowanie zadań. Wyniki te sugerują rolę jaką według badanych powinien pełnić kierownik projektu w zespole. Powinien on być osobą, która nie tylko zarządza zespołem, ale również jest w stanie skutecznie komunikować się z jego członkami oraz motywować ich do działania. Wysoka ocena kompetencji strategicznych i analitycznych poazuje, że kierownicy projektów powinni być w stanie myśleć krytycznie i podejmować decyzje na podstawie analizy danych. szczególne wysoko w tej kategorii oceniono umiejętność podejmowania decyzji oraz zarządzanie ryzykiem. Kompetencje te są szczególnie istotne w kontekście projektów informatycznych, gdzie zmiany mogą występować w każdej chwili, a kierownik projektu musi być w stanie szybko dostosować się do nowych warunków. Kompetencje finansowe i organizacyjne oceny poniżej 4, co sugeruje, że są one mniej istotne w kontekście zarządzania projektami IT. Jednak respondenci z większym doświadczeniem wyraźniej wyżej niż ogół ocenili rolę zarządzania budżetem i zasobami oraz orientację na wyniki. Szczególne nisko została oceniona wiedza teoretyczna z zakresu zarządzania projektami, co może sugerować, że praktyczne umiejętności i naturalne predyspozycje są bardziej cenione niż teoretyczna wiedza. indywidualne kompetencje personalne kierownika projektu również nie uzyskały zbyt wysokiej oceny, gdyż średnia dla tej Kategorii wyniosła poniżej 4. Warto zwrócić w tym przypadku uwagę na różnice między średnią ogólną a pracownikami bardziej doświadczonymi. Ta druga grupa zwróciła wyraźnie podkreśliła znaczenie takich cech jak cierpliwość i odporność na stres. Co może sugerować, że w miarę zdobywania doświadczenia w branży IT, respondenci zaczynają dostrzegać znaczenie tych cech

w kontekście zarządzania projektami. Kształtowanie cierpliwości i odporność na stres może być kluczowe dla skutecznego zarządzania projektami, zwłaszcza w obliczu wyzwań i stresu, które często towarzyszą pracy w branży IT. Kompetencje z zakresu inżynierii oprogramowania uzyskały najniższe oceny, co może wskazywać na ich marginalne znaczenie w kontekście kierowania projektami informatycznymi. Respondenci wskazali, że kierownik projektu nie musi być ekspertem technicznym, ale powinien mieć podstawową wiedzę na temat technologii i procesów związanych z inżynierią oprogramowania. W tej kategorii szczególnie wyróżniały się umiejętności formułowania wymagań dotyczących oprogramowania oraz znajomość cyklu życia systemów informatycznych. Bardzo duża różnica występuje w przypadku oceny znajomości cyklu życia systemów informatycznych, gdzie pracownicy z większym doświadczeniem ocenili tę umiejętność na 3,90, podczas gdy ogół na 2,50. Może to sugerować, że w celu skutecznego zarządzania projektem kierownik powinien dobrze znać cykl życia systemów informatycznych, aby móc skutecznie komunikować się z zespołem i podejmować decyzje dotyczące projektu. Warto również zauważyć, że umiejętności programowania oraz znajomość algorytmów i struktur danych uzyskały najniższe oceny, co sugeruje, że nie są one kluczowe dla kierownika projektu. Respondenci wskazali, że kierownik projektu nie musi być ekspertem technicznym, ale powinien mieć podstawową wiedzę na temat technologii a w szczególności procesów związanych z inżynierią oprogramowania.

5.4.2. Ocena wymaganego doświadczenia technicznego kierownika projektu

W kolejnej części badania, która skupiała się już konkretnie na ocenie znaczenia umiejętności z zakresu inżynierii oprogramowania u kierownika projektu, respondenci zostali poproszeni o wyrażenie opinii, czy kierownik projektu IT powinien posiadać praktyczne doświadczenie w inżynierii oprogramowania. Tylko 13% odpowiedziało stanowczo, iż jest to niezbędne, 23% respondentów oceniła praktyczne doświadczenie jako niepotrzebne. Zdecydowana większość zaś zwróciła uwagę, iż zależy to przede wszystkim od typu projektu. Ma to również potwierdzenie w przytaczanej w poprzednich rozdziałach literaturze, gdyż Nicholas i Steyn¹⁶⁶ wyraźnie podkreślili, że wymagane umiejętności z zakresu inżynierii oprogramowania u kierownika są uwarunkowane właśnie zaawansowaniem technicznym realizowanego projektu. Następnie respondenci, którzy odpowiedzieli na to pytanie "Tak" lub "Zależy" zostali poproszeni o ocenę poziomu umiejętności w dziedzinie inżynierii oprogramowania jakie powinien mieć kierownik

¹⁶⁶ Nicholas i Steyn, *Project Management for Business, Engineering, and Technology*.

Rysunek 5.3. Czy kierownik projektu IT powinien posiadać praktyczne doświadczenie w inżynierii oprogramowania?



Źródło: Opracowanie własne

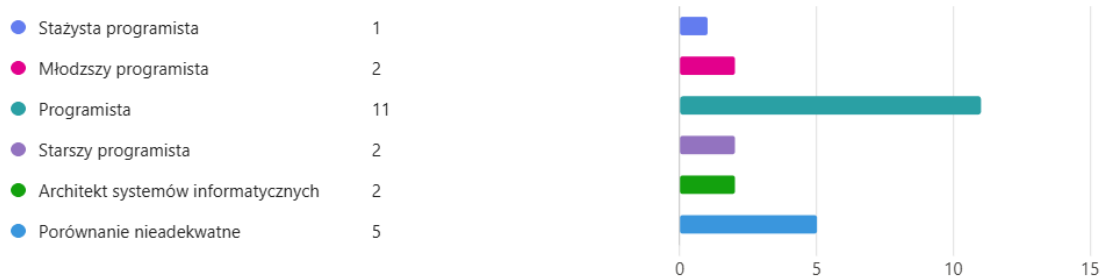
projektu informatycznego. W tej części badania respondenci mieli możliwość wyrażenia swojej opinii za pomocą porównania wymaganych umiejętności do kompetencji programisty na różnych etapach kariery. Respondenci mieli do wyboru pięć poziomów, które odpowiadały poziomowi umiejętności programisty na różnych etapach kariery zawodowej. Poziomy te to: stażysta, młodszy programista, programista, starszy programista oraz architekt systemów informatycznych. Dodatkowo określona została kategoria "Porównanie nieadekwatne" w celu zapewnienia większej swobody odpowiadającym i weryfikacji ich interpretacji umiejętności w dziedzinie inżynierii oprogramowania. Kategoria ta uzyskała stosunkowo wysoką popularność, aż pięciu odpowiadających wybrało tę odpowiedź. Co może sugerować, że biegłość techniczna u kierownika projektu nie jest tym samym co wymagane jest od programisty. Zdecydowana większość respondentów jednak wybrała opcję programista, według raporty BulldogJob średni czas doświadczenia dla programisty to 5 lat.¹⁶⁷ Jest to stosunkowo długi okres i może oznaczać duży próg wejścia dla przyszłych kierowników projektów informatycznych, szczególnie w przypadku projektów zaawansowanych technicznie.

5.5. Wpływ wiedzy z zakresu inżynierii oprogramowania na zarządzanie projektem informatycznym

W następnym pytaniu ankietowani zostali poproszeni o określenie jaki według nich wpływ na poszczególne aspekty zarządzania projektem informatycznym ma wiedza z zakresu inżynierii oprogramowania. Badane były trzy składowe: efektywna komunikacja, zarządzanie

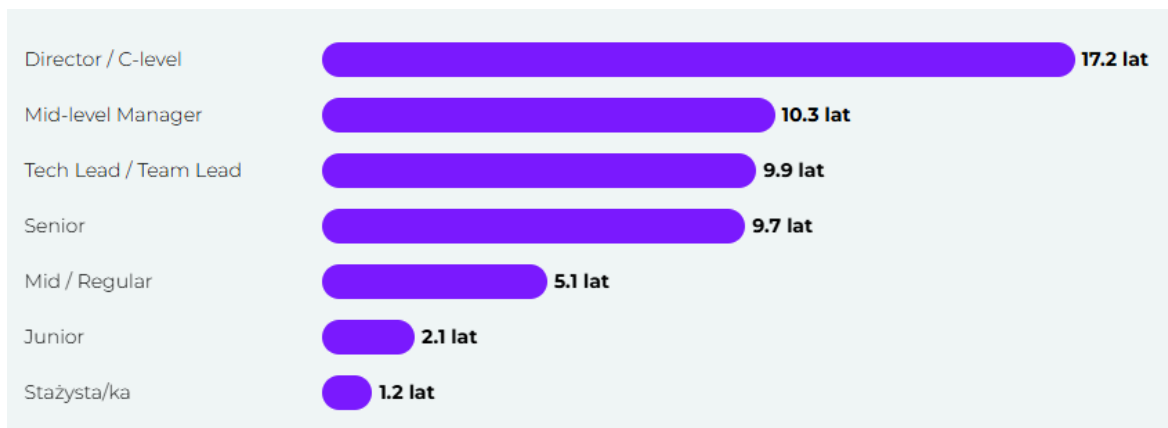
¹⁶⁷ Bulldogjob. *Raport z badania społeczności IT 2025*. <https://bulldogjob.pl/it-report/2025/programista>. Accessed: 2025-05-11. 2025.

Rysunek 5.4. Ocena poziomu umiejętności w dziedzinie inżynierii oprogramowania jakie powinien mieć kierownik projektu informatycznego



Źródło: Opracowanie własne

Rysunek 5.5. Poziom doświadczenia a doświadczenie w latach



Źródło: <https://bulldogjob.pl/it-report/2025/programista>

Rysunek 5.6. Wpływ wiedzy z zakresu inżynierii oprogramowania na poszczególne aspekty zarządzania projektem



Źródło: Opracowanie własne

ryzykiem oraz realizacją celów projektu, do oceny poszczególnych aspektów respondenci mieli do wyboru pięciostopniową skalę, gdzie 1 oznacza brak wpływu, a 5 oznacza bardzo duży wpływ. Znacząca większość oceniła wpływ wiedzy z zakresu inżynierii oprogramowania na efektywną komunikację oraz zarządzanie ryzykiem na 4 lub 5. Co może sugerować, że wiedza techniczna jest istotnym czynnikiem w kontekście tych aspektów zarządzania projektem. Dużo większą rozbieżność odpowiedzi widać w przypadku oceny realizacji celów projektu, respondenci byli bardziej podzieleni, co może sugerować, że wiedza techniczna nie jest kluczowym czynnikiem wpływającym na ten aspekt zarządzania projektem.

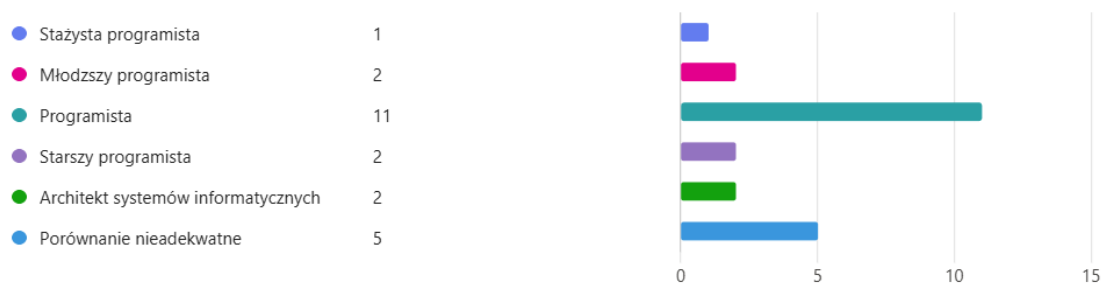
Tabela 5.2. Średnie oceny wpływu wiedzy z zakresu inżynierii oprogramowania na poszczególne aspekty zarządzania projektem informatycznym

Kategoria	Średnia / Średnia kierownik
Efektywna komunikacja z zespołem	3,40 / 3,45
Zarządzanie ryzykiem	3,27 / 3,27
Realizacja celów projektu	3,20 / 3,00
Średnia wszystkich kategorii	3,29 / 3,24

Średnia – średnia ocen dla wszystkich badanych; Średnia kierownik – średnia ocen dla pracowników z doświadczeniem na stanowisku kierowniczym. Źródło: Opracowanie własne.

Dodatkowo w tabeli zostało przedstawione porównanie średnich ocen dla wszystkich badanych oraz dla pracowników, którzy mieli doświadczenie na stanowiskach kierowniczych. Nie widać szczególnych różnic pomiędzy obiema grupami, osoby z doświadczeniem kierowniczym nie wydają się widzieć większego niż reszta wpływu wiedzy z zakresu inżynierii oprogramowania na wymienione aspekty zarządzania projektem informatycznym.

Rysunek 5.7. Czy ma Pan/Pani doświadczenie sytuacji, w której brak wiedzy technicznej u kierownika projektu negatywnie wpłynął na realizację projektu?



Źródło: Opracowanie własne

5.6. Negatywny wpływ braku wiedzy technicznej na zarządzanie projektem informatycznym

W ostatnim pytaniu zamkniętym badani odpowiadali czy doświadczyli oni negatywnego wpływu braku wiedzy technicznej u kierownika projektu na realizację projektu. Połowa ankietowanych odpowiedziała, że "Nie", zaś pozostała część podzieliła się na dwie grupy, które odpowiedziały "Tak" oraz "Nie wiem".

Dodatkowo osoby, które odpowiedziały "Tak" zostały poproszone o podanie konsekwencji, które ich zdaniem wynikły z braku wiedzy technicznej u kierownika projektu. Respondenci wskazali na następujące problemy:

- wydłużenie czasu realizacji projektu, w związku ze złą wyceną czasu realizacji zadań,
- brak decyzyjności spowodowany strachem przed podjęciem złej decyzji i nieznajomością tematu,
- źle sformułowane wymagania - nieprezyjne lub niemożliwe do zrealizowania,

Warto przytoczyć również dwie odpowiedzi, które szczególnie dobrze podsumowują problem:

Brak zrozumienia tematu przez kierownika projektu powodował, że nie chciał on się zaangażować w rozwiązywanie problemu i oczekiwał jedynie na wypracowanie rozwiązania przez inżynierów.

Opóźnienia w realizacji projektu - np. brak sprawczosci w koordynowaniu zadań

na styku wielu zespołów. Brak sprawczości na poziomie decyzyjnym/zarządu. Brak zrozumienia wartości dostarczanych przez rozwiązanie IT.

Wypowiedzi te jasno wskazują, że kierownik projektu, który nie posiada wiedzy technicznej, może mieć trudności w podejmowaniu decyzji oraz koordynowaniu działań zespołu. W rezultacie może to prowadzić do opóźnień w realizacji projektu oraz braku efektywności w zarządzaniu zespołem.

Interesująca wydaje się również wypowiedź dotycząca outsourcingu:

W przypadku outsourcingu lub korzystania z usług firm zewnętrznych PM może zostać łatwo oszukany przez zewnętrznego dostawcę.

Odpowiedź ta wskazuje na wcześniej niewspomniane ryzyko związane z brakiem wiedzy technicznej u kierownika projektu. W przypadku współpracy z zewnętrznymi dostawcami, co jest powszechnym trendem w branży IT, brak technicznego zrozumienia może prowadzić do sytuacji, w której dostawca narzuca nieoptymalne lub przesadnie kosztowne rozwiązania, a kierownik projektu nie jest w stanie skutecznie weryfikować ich zasadności z biznesowego punktu widzenia.

5.7. Zagrożenia wynikające z nadmiernego skupienia się kierownika projektu na aspektach technicznych

Ostatnia część badania było to pytanie otwarte i dotyczyło zagrożeń wynikających z nadmiernego skupienia się kierownika projektu na aspektach technicznych. Respondenci zostali poproszeni o wskazanie, jakie problemy mogą wystąpić w przypadku, gdy kierownik projektu zbyt koncentruje się na szczegółach technicznych. W toku badania wyłoniły się cztery główne obszary ryzyka, na które zwracali uwagę uczestnicy:

- **Micromanagement i wchodzenie w kompetencje zespołu** - gdy kierownik projektu angażuje się w każdy techniczny detal, zaczyna pełnić rolę „wąskiego gardła” w procesie decyzyjnym i realizacji zadań. Zamiast delegować odpowiedzialność doświadczonym programistom lub architektom, narzuca on im własne rozwiązania, co osłabia ich motywację, podważa poczucie autonomii oraz prowadzi do zaburzeń w naturalnym przepływie informacji. Tego typu mikrozarządzanie nie tylko wydłuża czas pracy nad

poszczególnymi zadaniami, ale również wprowadza konflikty kompetencyjne, gdy specjalista czuje, że jego wiedza i doświadczenie są lekceważone.

- **Zaniedbanie roli menedżerskiej i celów biznesowych** - skupienie na szczegółach technicznych powoduje, że kierownik traci z pola widzenia nadrzędne cele biznesowe projektu. Zamiast nadzorować postęp prac, zapewniać efektywną komunikację między interesariuszami oraz dbać o jakość dostarczanej wartości, poświęca większość czasu na analizę rozwiązań technicznych. W efekcie może dojść do rozmycia wymagań, braku spójności dokumentacji oraz stworzenia produktu, który choć technicznie dopracowany, to nie odpowiada potrzebom klienta ani strategicznym założeniom organizacji.
- **Ryzyko opóźnień i technicznego przeinwestowania** - zjawisko overengineeringu, czyli dążenie do idealnego rozwiązania technicznego,¹⁶⁸ często skutkuje przeciąganiem faz implementacji. W miejsce prostego i szybkiego wdrożenia pojawiają się rozbudowane konstrukcje architektoniczne „na zapas”, które nie zawsze przekładają się na realne korzyści, a jedynie generują dodatkowe koszty i opóźnienia.

Poniżej przedstawiono kilka wypowiedzi respondentów, które ilustrują te zagrożenia:

1. *„Zbytne dążenie do stworzenia idealnego technicznie systemu, co opóźnia tworzenie takiego systemu.”* Ta uwaga trafnie oddaje paradoks overengineeringu: zamiast dostarczyć prosty, działający produkt, zespół wpada w spiralę ulepszeń, zagrażając terminowemu zakończeniu projektu.
2. *„Odklejenie od biznesu. Overengineering. Tworzenie bottleneck’ów.”* Skondensowana definicja ryzyk: micromanagement łączy się tu z utratą perspektywy biznesowej, a wąskie gardła wynikają zarówno z nadmiernego uszczegółowienia, jak i braku klarownego priorytetu na potrzeby klienta.
3. *„W przypadku outsourcingu lub korzystania z usług firm zewnętrznych PM może zostać łatwo oszukany przez zewnętrznego dostawcę.”* Rzadziej omawiany aspekt – brak technicznej bezstronności w relacjach B2B może narazić projekt na niekorzystne zobowiązania, gdy dostawca proponuje rozwiązania trudne w niezależnej weryfikacji.

¹⁶⁸ M. Smółka. *The Over-Engineering Pendulum*. <https://threedots.tech/post/the-over-engineering-pendulum/>. Accessed: 2025-05-11. 2024.

5.8. Wnioski

Analiza wyników badania ankietowego wskazuje, że kluczową wartość w pracy kierownika projektu informatycznego stanowią umiejętności interpersonalne i komunikacyjne, które pozwalają na skuteczne budowanie relacji z zespołem oraz utrzymanie klarownej wymiany informacji. Równie istotne okazały się być kompetencje przywódcze i organizacyjne — zdolność motywowania, delegowania zadań oraz efektywnego zarządzania czasem i zasobami, co przekłada się na realizację założeń projektu w terminie.

Choć techniczne kompetencje z zakresu inżynierii oprogramowania otrzymały od respondentów niższe oceny, badanie wykazało, że podstawowa wiedza na temat cyklu życia systemów, formułowania wymagań czy znajomość metod weryfikacji kodu są ważne i przydatne w roli kierownika projektu informatycznego. Ankietowani zauważyli również potencjał w wiedzy z zakresu inżynierii oprogramowania jako wsparcie efektywnej komunikacji z zespołem. Z drugiej strony jednak zbyt duże zaangażowanie kierownika w szczegóły techniczne może prowadzić do mikrozarządzania, narastania kosztów oraz opóźnień spowodowanych overengineeringiem i utratą koncentracji na celach biznesowych.

Kierownik projektu powinien zatem znajdować złoty środek: łączyć silne kompetencje miękkie i przywódcze z kontekstową, adekwatną do stopnia skomplikowania projektu wiedzą techniczną. Dzięki takiemu zrównoważonemu podejściu możliwe jest zachowanie równowagi między sprawnym zarządzaniem zespołem a trafnym podejmowaniem decyzji biznesowych oraz rzetelną współpracą z ekspertami technicznymi.

Można więc stwierdzić, że skuteczny kierownik projektu informatycznego powinien funkcjonować jako swego rodzaju łącznik – mediator pomiędzy światem technologicznym a biznesowym, osoba, która nie tylko rozumie język kodu, ale przede wszystkim potrafi przekształcić wymagania klienta w realne działania zespołu. Tylko takie zrównoważenie kompetencji miękkich i twardych pozwala na stworzenie środowiska sprzyjającego efektywnej współpracy, minimalizacji ryzyka i dostarczaniu wartości dla interesariuszy.

Wyniki badania korespondują z ustaleniami przedstawionymi w literaturze przedmiotu, omawianej w poprzednich rozdziałach, co wzmacnia ich wiarygodność oraz potwierdza aktualność poruszanych zagadnień.

6. Podsumowanie i wnioski końcowe

Celem niniejszej pracy było zbadanie, czy kierownik projektu informatycznego powinien posiadać kompetencje z zakresu inżynierii oprogramowania, a jeśli tak – to w jakim zakresie i w jakich sytuacjach są one rzeczywiście potrzebne. Zagadnienie to zostało przeanalizowane zarówno z perspektywy teoretycznej, opartej na literaturze przedmiotu, jak i praktycznej – poprzez przeprowadzenie badania ankietowego wśród pracowników branży IT.

Wnioski płynące z tych dwóch źródeł okazały się komplementarne: z jednej strony potwierdzono istotność kompetencji miękkich i przywódczych w codziennej pracy kierownika projektu, z drugiej zaś strony uwidoczniła się rola kontekstowej wiedzy technicznej, szczególnie w przypadku bardziej złożonych przedsięwzięć informatycznych.

6.1. Złożoność roli kierownika projektu IT

Współczesny projekt informatyczny jest przedsięwzięciem interdyscyplinarnym, łączącym w sobie elementy zarządzania, technologii, analizy biznesowej, a często także kompetencji miękkich związanych z komunikacją z klientem oraz prowadzeniem zespołu. Kierownik projektu pełni w tym kontekście rolę centralną — odpowiada nie tylko za harmonogram i budżet, ale również za zapewnienie spójności działań w obrębie całego zespołu projektowego.

Analiza literatury pokazała, że zakres wymaganych kompetencji kierownika projektu jest bardzo szeroki i obejmuje nie tylko wiedzę z obszaru zarządzania, ale również umiejętności interpersonalne, przywódcze, analityczne i techniczne. Szczególną uwagę poświęcono tutaj specyfice projektów informatycznych, które znacząco różnią się od przedsięwzięć realizowanych w innych dziedzinach inżynierii czy zarządzania. Ich złożoność wynika nie tylko z samej natury technologii informatycznych, ale także z wysokiej dynamiki zmian, braku pełnej przewidywalności oraz trudności w jednoznacznym zdefiniowaniu i utrzymaniu wymagań przez cały cykl życia projektu. Projekty IT często opierają się na zmieniających się założeniach, współpracy między wieloma zespołami oraz integracji różnych komponentów — niekiedy rozwijanych równolegle, a niekiedy dziedziczonych po poprzednich systemach.

Już na etapie planowania pojawia się wiele krytycznych decyzji: wybór architektury systemu, stosu technologicznego, sposobu integracji, modelu zespołu czy nawet cyklu życia oprogramowania. Błędna decyzja w którymkolwiek z tych obszarów może prowadzić do kaskady problemów — od niedoszacowania kosztów i czasu, przez niedopasowanie

kompetencji zespołu do wybranej technologii, aż po fundamentalne ograniczenia wpływające na wydajność, bezpieczeństwo lub skalowalność rozwiązania.

W odróżnieniu od projektów budowlanych czy przemysłowych, gdzie produkt końcowy jest z reguły fizyczny i bardziej przewidywalny, w projektach informatycznych „produkt” — czyli system informatyczny — powstaje jako efekt wielu składowych, w tym częstych dynamicznych zmian w wymaganiach biznesowych. To sprawia, że konsekwencje błędów popełnionych na wczesnym etapie nie są łatwe do wykrycia od razu, a ich naprawa bywa kosztowna i czasochłonna. Na przykład błędnie zdefiniowane wymagania funkcjonalne mogą skutkować implementacją systemu, który nie odpowiada rzeczywistym potrzebom klienta, zaś wybór niewłaściwej architektury może uniemożliwić rozwój, skalowanie lub integrację z przyszłymi modułami.

Dlatego właśnie w projektach informatycznych tak istotna jest rola kierownika projektu jako osoby nie tylko koordynującej działania, ale także rozumiejącej konsekwencje wyborów technicznych oraz potrafiącej ocenić ryzyka związane z decyzjami podejmowanymi przez zespół. Wymaga to nie tyle bycia ekspertem technicznym, ile zdolności do prowadzenia dialogu z architektami i programistami, trafnego zadawania pytań i rozumienia, jak decyzje technologiczne wpływają na harmonogram, budżet i zakres prac. W tym kontekście, kompetencje kierownika projektu wykraczają poza klasyczne umiejętności zarządzania i obejmują również zdolności analityczne myślenia oraz podstawowego rozumienia złożonej materii technologicznej — właśnie po to, by móc w porę identyfikować zagrożenia i wspierać podejmowanie decyzji zgodnych z celami biznesowymi projektu.

6.2. Najważniejsze wyniki badania empirycznego

W celu pogłębienia wiedzy teoretycznej przeprowadzono badanie ankietowe wśród 30 osób zatrudnionych w branży IT, reprezentujących różne role i poziomy doświadczenia. Badanie pozwoliło na zidentyfikowanie realnych oczekiwań i obserwacji dotyczących pracy kierownika projektu.

Najważniejsze wnioski z badania to:

- Kompetencje interpersonalne, komunikacyjne oraz przywódcze zostały ocenione jako zdecydowanie najistotniejsze — najwyższe średnie noty uzyskały m.in. umiejętności komunikacyjne (4,67/5) oraz zarządzanie czasem i zespołem.

- Kompetencje techniczne z zakresu inżynierii oprogramowania, takie jak programowanie, znajomość struktur danych czy systemów operacyjnych, zostały ocenione najniżej — średnia ocena tej kategorii wyniosła 2,67/5.
- Pomimo niskiej ogólnej oceny kompetencji technicznych, respondenci zwrócili uwagę na ich użyteczność w kontekście efektywnej komunikacji z zespołem i lepszego zarządzania ryzykiem.
- Aż 63% ankietowanych stwierdziło, że wymagany poziom technicznej wiedzy u kierownika projektu powinien zależeć od rodzaju i stopnia zaawansowania projektu.
- Tylko 13% respondentów uznało, że praktyczne doświadczenie w inżynierii oprogramowania jest niezbędne, co pokazuje wyraźnie, że dominującą postawą jest elastyczność, a nie dogmatyczne podejście.

Dodatkowo, respondenci podkreślili ryzyko związane zarówno z brakiem wiedzy technicznej (np. problemy z komunikacją, błędne wymagania, podatność na manipulacje ze strony dostawców), jak i z nadmiernym zaangażowaniem w kwestie techniczne, które może prowadzić do mikrozarządzania, overengineeringu i oderwania od realnych celów biznesowych.

6.3. Odpowiedź na pytanie badawcze

W świetle przeprowadzonych analiz można stwierdzić, że kierownik projektu informatycznego nie musi być ekspertem technicznym w tradycyjnym rozumieniu tego pojęcia. Oczekiwanie od niego kompetencji na poziomie doświadczonego programisty czy architekta, jest nie tylko nieuzasadnione, ale i potencjalnie szkodliwe, jeśli prowadzi do nadmiernego ingerowania w zadania zespołu technicznego i ingerowania aspektów biznesowych projektu.

Z drugiej strony, całkowity brak technicznego rozeznania może skutkować poważnymi trudnościami w zarządzaniu projektem — brakiem decyzyjności, niezrozumieniem ryzyk, problemami komunikacyjnymi czy trudnościami w interpretowaniu propozycji zespołu.

Odpowiedzią na pytanie badawcze nie jest więc prosta, nie można jednoznacznie powiedzieć „tak” lub „nie”, ale raczej: „to zależy”. Rola i zakres wiedzy technicznej kierownika projektu powinny być dostosowane do specyfiki danego przedsięwzięcia — w projektach o mniejszej złożoności technicznej kompetencje miękkie będą dominujące, natomiast w projektach zaawansowanych technologicznie lub tworzących produkt czysto techniczny a nie biznesowy znajomość podstawowych zasad inżynierii oprogramowania staje się czynnikiem kluczowym i wymagany do skutecznego zarządzania.

6.4. Rekomendacje i znaczenie praktyczne

Dla praktyków i organizacji zarządzających projektami informatycznymi płyną z tej pracy istotne wnioski. Kluczowym czynnikiem sukcesu jest znalezienie równowagi pomiędzy kompetencjami technicznymi a miękkimi. Kierownik projektu powinien być osobą, która potrafi komunikować się z programistami, rozumieć kontekst techniczny wymaganych decyzji, ale niekoniecznie musi je podejmować samodzielnie. Jego rola to raczej łączenie światów — biznesu, technologii i ludzi — w spójną całość, która pozwala realizować cele projektu.

Warto również podkreślić, że wyniki badania ankietowego są zbieżne z obserwacjami zawartymi w literaturze, co dodatkowo wzmacnia ich wiarygodność. Model kompetencyjny kierownika projektu powinien być tworzony nie według sztywnego wzorca, lecz elastycznie. Powinien on uwzględniać charakterystyki zespołu, rodzaju projektu oraz poziomu ryzyka i złożoności technologicznej.

6.5. Zakończenie

W realiach współczesnych projektów informatycznych, które są coraz bardziej złożone i wielowymiarowe, niezbędne staje się hybrydowe podejście do zarządzania — oparte na współpracy, komunikacji i zrozumieniu technologii, ale bez konieczności technicznego mikrozarządzania. Kierownik projektu nie musi być specjalistą od kodu, ale powinien być partnerem dla specjalistów — kimś, kto potrafi ich zrozumieć, zapytać, zakwestionować i przełożyć język technologii na język wartości biznesowej. Takie podejście wydaje się najbardziej zbieżne zarówno z wynikami badań empirycznych, wiedzą zawartą w literaturze, jak i z praktyką współczesnych organizacji.

Bibliografia

- Agile Alliance. *Manifesto for Agile Software Development*. 2001. URL: <http://agilemanifesto.org/>.
- Alshamrani, A. i A. Bahattab. „A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model”. W: *International Journal of Computer Science Issues (IJCSI)* 12.1 (2015), s. 106.
- Alvarenga, J. C. i in. „The project manager core competencies to project success”. W: *International Journal of Managing Projects in Business* (2019). DOI: 10.1108/IJMPB-12-2018-0274.
- Arora, R. i N. Arora. „Analysis of SDLC models”. W: *International Journal of Current Engineering and Technology* 6.1 (2016), s. 268–272.
- Auditor General of Canada, Report of the. *Systems under development: Managing the risks*. 1995.
- Barki, H., S. Rivard i J. Talbot. „Toward an assessment of software development risk.” W: *Journal of Management Information Systems* (1993).
- Blackburn, S. „The project manager and the project-network”. W: *International Journal of Project Management* (2002). DOI: 10.1016/S0263-7863(01)00069-2.
- Boehm, B. „Get Ready for Agile Methods, with Care”. W: *Computer* 35.1 (2002), s. 64–69.
- Brill, J. M., M. J. Bishop i A. E. Walker. „The Competencies and Characteristics Required of an Effective Project Manager: A Web-Based Delphi Study”. W: *ITLS Faculty Publications* (2006). DOI: 10.1007/s11423-006-8251-y.
- Bulldogjob. *Raport z badania społeczności IT 2025*. <https://bulldogjob.pl/it-report/2025/programista>. Accessed: 2025-05-11. 2025.
- Burns, J. M. *Leadership*. Harper & Row. 1978.
- Celestin, M., H. Jean Paul i N. Clementine. „Analysis of Effective Communication and Project Success: Survey on Electricity Access Roll Out Project at EDCL -EARP”. W: *International Journal of Computational Research and Development* (2023).
- Chipulu, M. i in. „A Multidimensional Analysis of Project Manager Competences”. W: *IEEE Transactions on Engineering Management* (2012). DOI: 10.1109/TEM.2012.2215330.
- Chmielarz, W. *Information Technology Project management*. Wydawnictwo Naukowe Wydziału Zarządzania, 2015.
- Crawford, L., J. B. Hobbs i J. R. Turner. *Project Categorizations Systems: Aligning Capability with Strategy for Better Results*. 2005.
- Currie, W. „The art of justifying new technology to top management”. W: *Omega* (1989).
- Ćwiklicki, M. *Metodyka przeglądu zakresu literatury (scoping review)*. 2020.
- Dybå, T. „Improvisation in Small Software Organizations”. W: *IEEE Software* 17.5 (2000), s. 82–87.
- Dybå, T. i T. Dingsøyr. „What do we know about agile software development?” W: *IEEE software* 26.5 (2009), s. 6–9.
- Ebrahim, N. A., S. Ahmed i Z. Taha. „Virtual teams: A literature review”. W: *Australian journal of basic and applied sciences* 3.3 (2009), s. 2653–2669.

6. Bibliografia

- Farley, D. *Modern Software Engineering: Doing What Works to Build Better Software Faster*. Addison-Wesley Professional, 2021.
- Gartner, Inc. *Gartner Says Worldwide IT Spending Forecast to Grow 2.4 Percent in 2017*. 2014.
- Gemino, A., B. Reich i C. Sauer. „A Temporal Model of Information Technology Project Performance”. W: *J. of Management Information Systems* 24 (sty. 2008), s. 9–44. DOI: 10.2753/MIS0742-1222240301.
- Gido, J. i J. P. Clements. „Successful Project Management”. W: *South Western Hill* (1999).
- Gottschalk, P. i J. T. Karlsen. „A comparison of leadership roles in internal IT projects versus outsourcing projects.” W: *Industrial Management & Data Systems* (2005). DOI: 10.1108/026355705106332203.
- Group, Standish. *CHAOS Report*. 2015.
- Group, World Bank. *World Development Report*. 2016.
- Haggerty, N. *Understanding the Link between IT Project Manager Skills and Project Success*. 2000.
- Institute, Project Management. *PMBOK Guide – Fourth Edition*. 2009.
- *PMBOK Guide – Seventh Edition*. 2021.
- *PMBOK Guide – Sixth Edition*. 2017.
- *Project Management Job Growth and Talent Gap*. 2017.
- Jałocha, B. „Projectification of the European Union and its Implications for Public Labour Market Organisations in Poland”. W: *Journal of Project, Program and Portfolio Management* (2012).
- Jiang, J. J., G. Klein i S. Margulis. *Important Behavioral Skills for IS Project Managers: The Judgments of Experienced IS Professionals*. 1998.
- Juchniewicz, M. „Projektyzacja – koncepcja, zakres, konsekwencje”. W: *Studia i Prace Kolegium Zarządzania i Finansów SGH* (2018).
- Kanter, R. M. „Power Failure in Management Circuits”. W: *Harvard Business Review* 57.4 (1979), s. 65–75.
- Keil, M. i in. „A framework for identifying software project risks.” W: *Communications of the ACM* (1998).
- Kerzner, H. *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. John Wiley & Sons, 2017.
- Langer, N., S. Slaughter i T. Mukhopadhyay. *How do Project Managers' Skills Affect Project Success in IT Outsourcing?* 2008.
- Loh, L. i N. Venkatraman. „Determinants of information technology outsourcing: A cross-sectional analysis”. W: *Journal of management information systems* 9.1 (1992), s. 7–24.
- Lyytinen, K. i D. Robey. „Learning failure in information systems development.” W: *Information Systems Journal* (1999).
- Marchewka, J. *Information Technology Project management*. Northern Illinois University, 2003.
- Maylor, H. i in. „From Projectification to Programmification”. W: *International Journal of Project Management* (2006).
- McConnell, S. *Rapid Development: Taming Wild Software Schedules*. Microsoft Press, 1996.
- McFarlan, F. W. „Portfolio approach to information systems.” W: *Harvard Business Review*, (1981).
- McGrew, A. „A Global Society”. W: *Modernity and Its Futures*. Red. S. Hall, D. Held i A. McGrew. Cambridge: Polity Press, 1990.

- Meyer, B. „Agile”. W: *The good, the hype and the ugly*. Switzerland: Springer International Publishing (2014).
- Midler, C. „Projectification of the Firm: The Renault Case”. W: *Scandinavian Journal of Management* (1995).
- Mintzberg, H. *Power In and Around Organizations*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
- Mumford, M. D. i in. „Leadership skills for a changing world: Solving complex social problems”. W: *The Leadership Quarterly* (2000). DOI: [https://doi.org/10.1016/S1048-9843\(99\)00041-7](https://doi.org/10.1016/S1048-9843(99)00041-7).
- Nerur, S. i V. Balijepally. „Theoretical Reflections on Agile Development Methodologies”. W: *Communications of the ACM* 50.3 (2007), s. 79–83.
- Nicholas, J. M. i H. Steyn. *Project Management for Business, Engineering, and Technology*. Elsevier, 2012.
- Nidiffer, K., C. Woody i T. A. Chick. *Program Manager’s Guidebook for Software Assurance*. Software Solutions i CERT Divisions, Software Engineering Institute, 2018.
- Nieto-Rodriguez, A. *The Focused Organizations*. Gower, 2017.
- People, Arras. *Project Management Benchmark Report*. 2010.
- *Project Management Benchmark Report*. 2015.
- Podgórska, M. i M. Pichlak. *Analysis of project managers’ leadership competencies Project success relation: what are the competencies of polish project leaders?* 2018.
- Pollack, J., J. Helm i D. Adler. „What is the Iron Triangle, and how has it changed?” W: *International Journal of Managing Projects in Business* (2018). DOI: [10.1108/IJMPB-09-2017-0107](https://doi.org/10.1108/IJMPB-09-2017-0107).
- Powell, P. „Information technology evaluation: is it different?” W: *Journal of the operational research society* (1992).
- Prawelska-Skrzypek, G. *Zarządzanie projektami jako szansa i wyzwanie dla administracji publicznej*. Wydawnictwo SGH, 2011.
- PWC. *Capital Projects and Infrastructure Spending, Outlook to 2025*. 2014.
- Report Summary no. 200, Parliamentary Office of Science i Technology. *Government IT projects*. 2003.
- Robertson, R. i K. E. White. „What is globalization?” W: *The Blackwell companion to globalization* (2007), s. 54–66.
- Roman, D. *Science, Technology, and Innovation: A Systems Approach*. Grid Publishing, 1980.
- Rosenau, J. *Successful Project Management*. John Wiley & Sons Australia, Limited, 1998.
- Royce, W. W. „Managing the Development of Large Software Systems”. W: *Proceedings of IEEE WESCON* (1970), s. 1–9.
- Rubinstein, D. „Standish Group report: There’s less development chaos today”. W: *Software Development Times* (2007).
- Sauer, C., A. Gemino i B. H. Reich. „IT project performance: The impact of size and volatility”. W: *Communications of the ACM* (2007).
- Schwalbe, K. *Information Technology Project management*. Cengage, 2018.
- Sharma, R. i P. Yetton. „The contingent effects of management support and task interdependence on successful information systems implementation.” W: *MIS Quarterly* (2003).
- Silva de Araújo, C. C. i C. D. Pedron. „IT project manager competencies and IT project success: a qualitative study.” W: *Organisational Project Management* 2.1 (2015).

6. Bibliografia

- Smółka, M. *The Over-Engineering Pendulum*. <https://threedots.tech/post/the-over-engineering-pendulum/>. Accessed: 2025-05-11. 2024.
- Sommerville, I. *Software Engineering*. 10th. Pearson, 2015.
- Sommerville, J., N. Craig i J. Hendry. „The role of the project manager: All things to all people?” W: *Structural Survey* (2010). DOI: 10.1108/02630801011044235.
- Stevenson, D. H. i J. A. Starkweather. „PM critical competency index: IT execs prefer soft skills”. W: *International journal of project management* 28.7 (2010), s. 663–671.
- Suburayan, V., P. Davachelvan i S. Kayalvizhi. „Evolving A New Model (SDLC Model2010) For Software Development Life Cycle (SDLC)”. W: *International Journal of Computer Science and Network Security* (2010).
- Tabassi, A. A. i in. „Leadership Competences of Sustainable Construction Project Managers”. W: *Journal of Cleaner Production* (2016). DOI: 10.1016/j.jclepro.2016.02.076.
- Thomaidis, N. S., N. Nikitakos i G. D. Dounias. „The evaluation of information technology projects: A fuzzy multicriteria decision-making approach”. W: *International Journal of Information Technology & Decision Making* (2006).
- Thomas, D. i A. Hunt. *Pragmatyczny programista Od czeladnika do mistrza*. Helion, 2021.
- Tohidi, H. „Human resources management main role in information technology project management”. W: *Procedia CS* 3 (grud. 2011), s. 925–929. DOI: 10.1016/j.procs.2010.12.151.
- Turner, J. R. *Gower Handbook of Project Management*. Routledge, 2016.
- Turner, R. i R. Müller. „Communication and Co-operation on Projects Between the Project Owner As Principal and the Project Manager as Agent”. W: *European Management Journal* (2004). DOI: 10.1016/j.emj.2004.04.010.
- „Leadership competency profiles of successful project managers”. W: *International Journal of Project Management* (2010). DOI: 10.1016/j.ijproman.2009.09.003.
- Udo, N. i S. Koppensteiner. „What are the core competencies of a successful project manager? Paper presented at PMI® Global Congress 2004—EMEA”. W: *Project Management Institute* (2004).
- Umaji, K. i W. Paireekreng. *A Study of Project Manager Skills for Enterprise Data Platform Project*. 2021.
- Van Wyngaard, C., J. H. Pretorius i L. Pretorius. *Theory of the triple constraint — A conceptual review*. 2012. DOI: 10.1109/IEEM.2012.6838095.
- Vaupel, R., G. Schmolke i A. Kruger. *Customer-focused management by projects*. MacMillan Publisher, 2000.
- Washizaki, H. *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide)*. Waseda University Japan, 2024.
- Williams, B. R. *Technology, Investment and Growth*. London: Chapman & Hall, 1967.
- Wyrozębski, P. i in. *Project Management - Challenges and Research Results*. 2016. DOI: 10.6084/m9.figshare.4502792.
- Yetton, P. i in. „A model of information systems development project performance.” W: *Information Systems Journal* (2000).

- Ziek, P. i J. D. Anderson. „Communication, dialogue and project management”. W: *International Journal of Managing Projects in Business* (2015). doi: 10.1108/IJMPB-04-2014-0034.
- Zmud, R. W. „Management of large software development efforts”. W: *MIS Quarterly* (1980).

Spis rysunków

2.1	Porównanie projektów i funkcji	8
2.2	Struktura w organizacjach	9
2.3	Żelazny trójkąt projektu	10
2.4	Stosunek projektów do operacji w organizacjach	12
2.5	Procent organizacji z biurem zarządzania projektami	13
2.6	Rezultaty projektów	13
2.7	Wskaźnik sukcesu dużych projektów informatycznych w sektorze publicznym	15
2.8	Porównanie sukcesu dla projektów informatycznych i wszystkich projektów .	16
3.1	Porównanie cyklu życia oprogramowania do cyklu życia projektu	22
3.2	Wartości Agile	26
3.3	Porównanie modelu kaskadowego z Agile	28
4.1	Role kierownika projektu	42
5.1	Czas pracy w branży IT	52
5.2	Charakter realizowanych projektów	54
5.3	Czy kierownik projektu IT powinien posiadać praktyczne doświadczenie w inżynierii oprogramowania?	56
5.4	Ocena poziomu umiejętności w dziedzinie inżynierii oprogramowania jakie powinien mieć kierownik projektu informatycznego	57
5.5	Poziom doświadczenia a doświadczenie w latach	57
5.6	Wpływ wiedzy z zakresu inżynierii oprogramowania na poszczególne aspekty zarządzania projektem	58
5.7	Czy ma Pan/Pani doświadczenie sytuacji, w której brak wiedzy technicznej u kierownika projektu negatywnie wpłynął na realizację projektu?	59

Spis tabel

2.1	Rezultaty projektów w latach 2011-2015	14
2.2	Podsumowanie wyników badania CHAOS	15
3.1	Kategoryzacja ryzyk według modelu czasowego	35

4.1	Kompetencje kierowników według raportu Arras	43
4.2	Analiza częstotliwości występowania kompetencji kierowników projektów . .	44
5.1	Średnie oceny kompetencji kierownika projektu IT	53
5.2	Średnie oceny wpływu wiedzy z zakresu inżynierii oprogramowania na poszczególne aspekty zarządzania projektem informatycznym	58

Spis załączników

Czy kierownik projektu informatycznego powinien być specjalistą do spraw inżynierii oprogramowania?

Streszczenie.

Celem niniejszej pracy magisterskiej było zbadanie, czy i w jakim zakresie kierownik projektu informatycznego powinien posiadać kompetencje z zakresu inżynierii oprogramowania. W pierwszej części przeprowadzono kompleksową analizę literatury przedmiotu, uwzględniającą zarówno klasyczne podejścia zarządcze (PMBOK, PRINCE2), jak i specyfikę cyklu życia oprogramowania (SDLC, Agiles). Wskazano, że projekty informatyczne charakteryzują się wysoką dynamiką zmian, przyrostowością wymagań oraz złożonością technologiczną, co wymaga od kierownika umiejętności łączenia perspektywy biznesowej z kontekstem technicznym.

W części empirycznej przeprowadzono anonimową ankietę wśród 30 pracowników branży IT (programiści, kierownicy projektów, testerzy), w której oceniano znaczenie różnych kompetencji kierownika projektu w skali Likerta oraz zadawano pytania otwarte dotyczące dotychczasowych doświadczeń. Wyniki wykazały, że najwyżej cenione są kompetencje interpersonalne i komunikacyjne, a także umiejętności przywódcze i organizacyjne. Kompetencje techniczne otrzymały znacznie niższe oceny, jednak respondenci podkreślali wartość podstawowego rozumienia cyklu życia oprogramowania i formułowania wymagań jako wsparcia efektywnej współpracy i zarządzania ryzykiem.

Analiza otwartych odpowiedzi ujawniła dwa główne zagrożenia: brak wiedzy technicznej prowadzi do opóźnień i błędnych decyzji, natomiast nadmierne zaangażowanie kierownika projektu informatycznego w szczegóły techniczne skutkuje mikrozarządzaniem i overengineeringiem. W odpowiedzi na te wyzwania sformułowano rekomendacje: priorytet kompetencji miękkich oraz elastyczne podejście do zakresu wiedzy z zakresu inżynierii oprogramowania w przypadku kierownika projektu.

Wnioski pracy wskazują, że skuteczny kierownik projektu informatycznego pełni rolę mediatora pomiędzy światem biznesu a zespołem technicznym, łącząc zdolności komunikacyjne i przywódcze z adekwatną do złożoności projektu wiedzą inżynierską. Taki model kompetencji sprzyja osiąganiu celów biznesowych przy zachowaniu wysokiej jakości rozwiązań technologicznych.