



---

SZKOŁA GŁÓWNA HANDLOWA W WARSZAWIE  
WARSAW SCHOOL OF ECONOMICS

Instytut Infrastruktury, Transportu i Mobilności

# Praca dyplomowa magisterska

na kierunku E-biznes

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

{Miłosz Truszkowski}

Numer albumu 131551

promotor

Dr Hab. Bartosz Gucza

WARSZAWA 2025



# Spis treści

<b>1. Wstęp</b>	5
<b>2. Charakterystyka projektów</b>	7
2.1. Definicja projektu	7
2.2. Żelazny trójkąt projektu	9
2.3. Projektyzacja	11
2.4. Rezultaty projektów	13
2.5. Projekty informatyczne w kontekście rosnącej liczby projektów	13
<b>3. Specyfika projektów informatycznych</b>	16
3.1. Niechęć do projektów informatycznych	17
3.2. Cykl życia w projektach informatycznych	18
3.2.1. Cykl życia projektu	18
3.2.2. Cykl życia produktu	19
3.2.3. Integracja PLC i SDLC w projektach informatycznych	19
3.3. Modele SDLC	20
3.3.1. Model kaskadowy	22
3.3.2. Agile	24
3.4. Różnorodne technologie	26
3.5. Specjaliści IT	26
3.6. Ryzyka w projektach IT	27
3.6.1. Czasowy model ryzyk	29
3.7. Ewaluacja projektów informatycznych	33
3.8. Ostatnie trendy w projektach informatycznych	34
3.8.1. Globalizacja	34
3.8.2. Outsourcing	34
3.8.3. Zespoły zdalne	35
<b>4. Kierownik projektu</b>	37
4.1. Kierownik projektu	37
4.2. Role i zadania kierownika projektu	37
4.3. Kompetencje kierownika projektu	39
4.3.1. Komunikacja	40
4.3.2. Przywództwo	42
4.3.3. Zarządzanie sobą (ang. Self-management)	42
4.3.4. Umiejętności zarządcze	43
4.3.5. Umiejętności techniczne	43
4.4. Znaczenie wiedzy z zakresu inżynierii oprogramowania dla kierownika projektu	44

<b>5. Badanie opinii na temat wykształcenia i roli kierownika projektu informatycznego</b>	47
5.1. Grupa badawcza	47
5.2. Zadawane pytania	47
5.3. Wyniki badania	47
5.4. Wnioski	47
<b>6. Podsumowanie</b>	48
<b>Bibliografia</b>	49
<b>Wykaz symboli i skrótów</b>	54
<b>Spis rysunków</b>	54
<b>Spis tabel</b>	54
<b>Spis załączników</b>	54

# 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.<sup>12</sup> 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.<sup>3</sup> 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. Badania 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.<sup>45</sup>

<sup>1</sup> Nicole Haggerty. *Understanding the Link between IT Project Manager Skills and Project Success*. 2000.

<sup>2</sup> Nishtha Langer, Sandra Slaughter i Tridas Mukhopadhyay. *How do Project Managers' Skills Affect Project Success in IT Outsourcing?* 2008.

<sup>3</sup> Lynn Crawford, J. Brian Hobbs i J. Rodney Turner. *Project Categorizations Systems: Aligning Capability with Strategy for Better Results*. 2005.

<sup>4</sup> Marzena Podgórska i Magdalena Pichlak. *Analysis of project managers' leadership competencies Project success relation: what are the competencies of polish project leaders?* 2018.

<sup>5</sup> James J. Jiang, Gary Klein i Steve Margulis. *Important Behavioral Skills for IS Project Managers: The Judgments of Experienced IS Professionals*. 1998.

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.<sup>6</sup> 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 zbyt wiele tematu inżynierii oprogramowania, lub jedynie na pewnej gałęzi projektów informatycznych.<sup>7</sup> 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.

---

<sup>6</sup> Marek Ćwiklicki. *Metodyka przeglądu zakresu literatury (scoping review)*. 2020.

<sup>7</sup> Kittisak Umaji i Worapat 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.<sup>8</sup> Turner<sup>9</sup> 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. Kerzner<sup>10</sup> oraz Turner<sup>11</sup> za istotę projektu 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 pionowe funkcyjne oraz warstwy zarządcze. Wyspy te w ramach projektu łączone są w tymczasową organizację mającą wspólny cel. Jednak osiągnięcie współpracy między

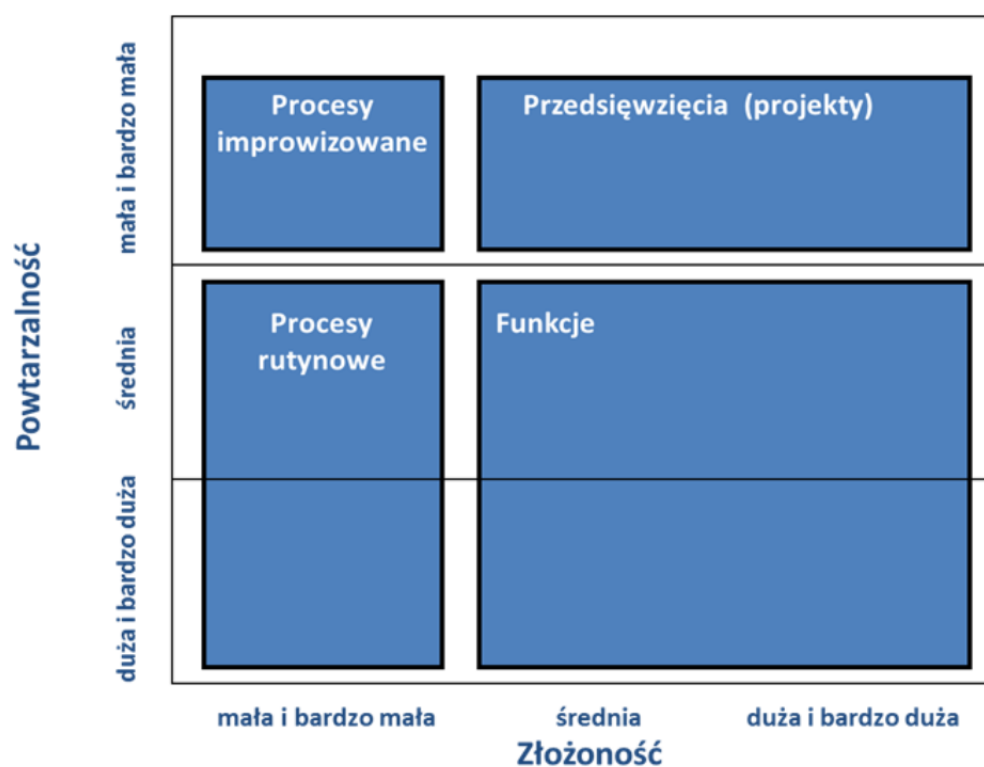
<sup>8</sup> Project Management Institute. *PMBOK Guide – Fourth Edition*. 2009.

<sup>9</sup> J. R. Turner. *Gower Handbook of Project Management*. Routledge, 2016.

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

<sup>11</sup> Turner, *Gower Handbook of Project Management*.

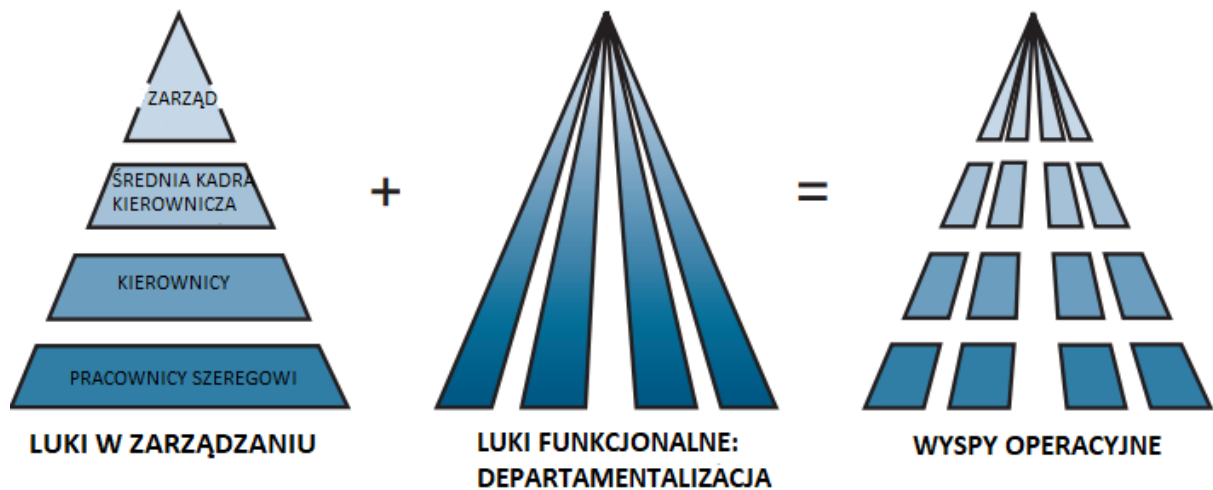
**Rysunek 2.1.** Porównanie projektów i funkcji



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

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 przewozu 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

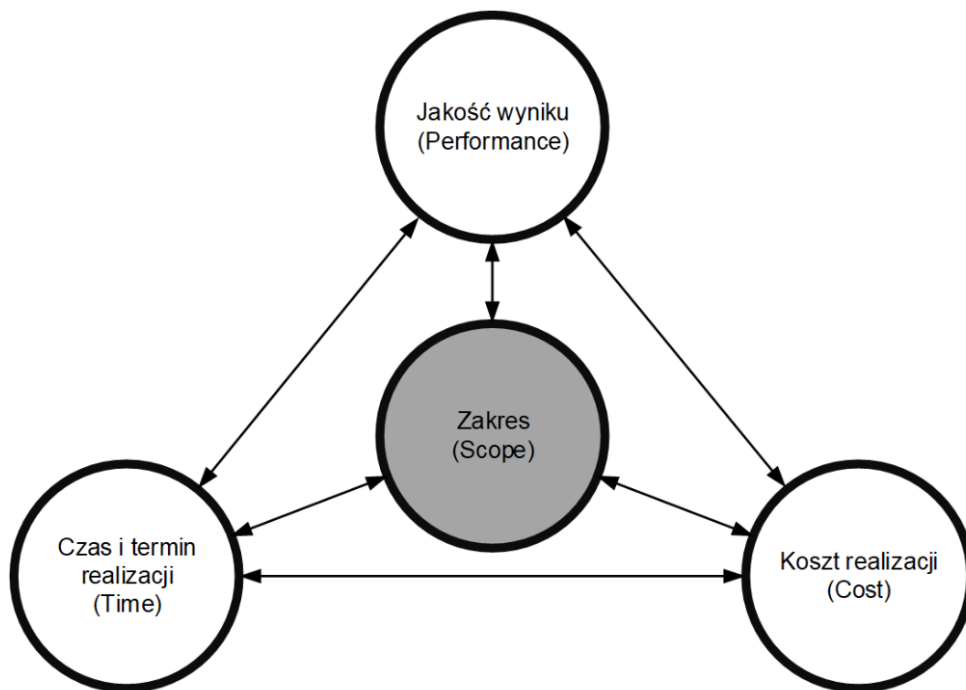
W celu ułatwienia zrozumienia kluczowych ograniczeń projektu opracowano koncepcję żelaznego trójkąta projektu. Ukazuje ona wzajemne zależności między czterema kluczowymi ograniczeniami projektu: zakresem, czasem, jakością i kosztem.<sup>1213</sup> Zakres definiuje wszystkie prace niezbędne do stworzenia finalnego produktu, usługi lub rezultatu projektu. Obejmuje on cele, zadania, funkcje i cechy projektu.<sup>14</sup> 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

<sup>12</sup> Julien Pollack, Jane Helm i Daniel 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.

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

<sup>14</sup> C. Van Wyngaard, Jan-Harm Pretorius i Leon Pretorius. *Theory of the triple constraint — A conceptual review*. 2012. DOI: 10.1109/IEEM.2012.6838095.

**Rysunek 2.3.** Żelazny trójkąt projektu



Źródło: Wykład *Podstawy Zarządzania Projektami*, Mateusz Juchniewicz

aspekty funkcjonalne, jak i niefunkcjonalne, 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.
- 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.<sup>15</sup> Proces ten wpływa zarówno na struktury organizacyjne, jak i na podejście do zarządzania w sektorach publicznym i prywatnym.<sup>16</sup> Pojęcie projektyzacji zostało wprowadzone już w latach 90. XX wieku. Kluczowe badania przeprowadzone przez Midlera oraz Maylora i współpracowników<sup>17</sup> 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.<sup>18</sup> Z drugiej strony jednak, ciągła intensyfikacja realizowanych projektów może prowadzić do nadmiernego obciążenia pracowników, wzrostu presji i stresu, a także do fragmentacji działań organizacyjnych.<sup>19</sup>

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.<sup>20</sup> 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%.<sup>21</sup> 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.”<sup>22</sup>

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

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

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

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

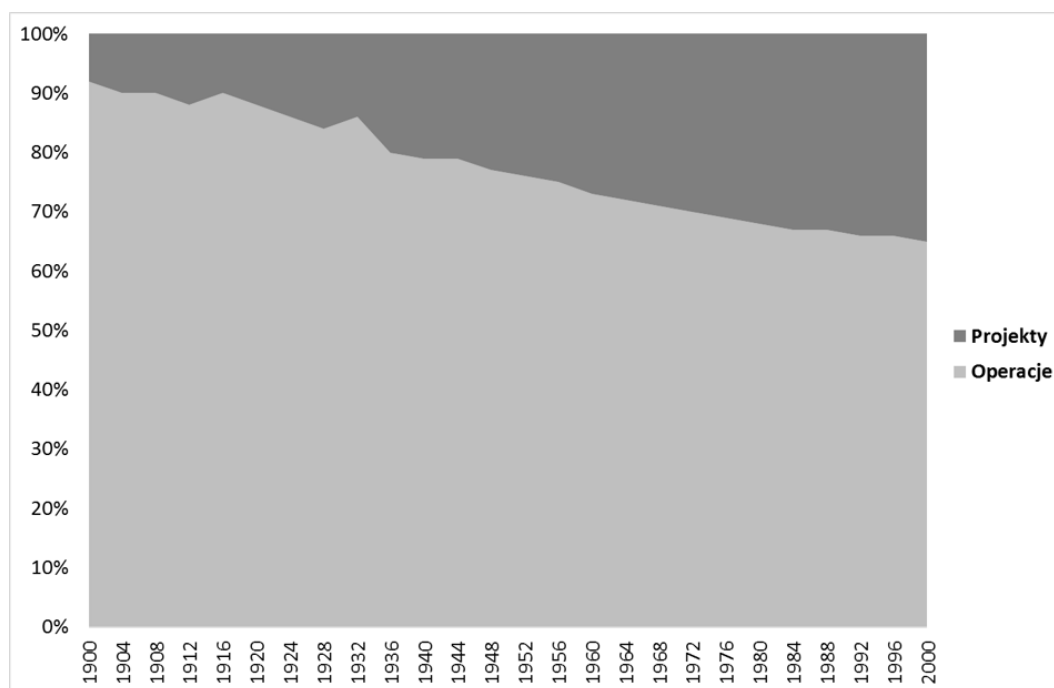
<sup>19</sup> B. Jałocha. „Projectification of the European Union and its Implications for Public Labour Market Organisations in Poland”. W: *Journal of Project, Program and Portfolio Management* (2012).

<sup>20</sup> PWC. *Capital Projects and Infrastructure Spending, Outlook to 2025*. 2014.

<sup>21</sup> A. Nieto-Rodriguez. *The Focused Organizations*. Gower, 2017.

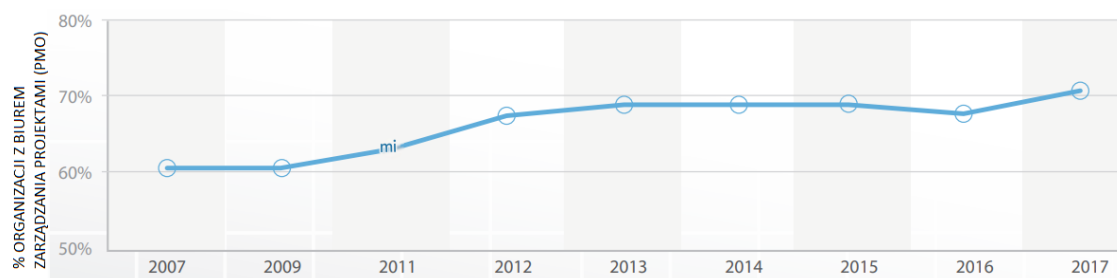
<sup>22</sup> Project Management Institute. *Project Management Job Growth and Talent Gap*. 2017.

**Rysunek 2.4.** Stosunek projektów do operacji w organizacjach



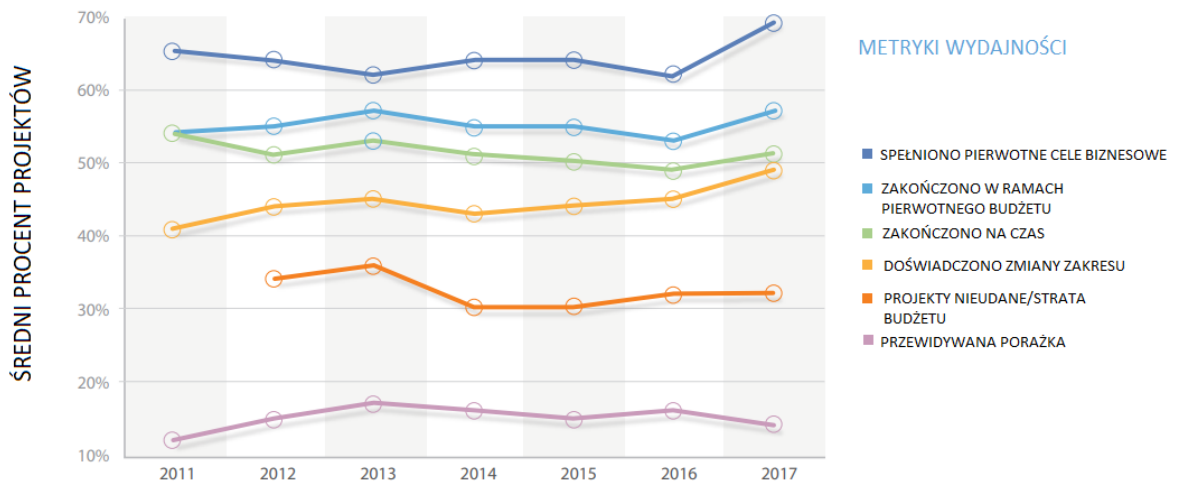
Źródło: Wykład *Podstawy Zarządzania Projektami*, Mateusz Juchniewicz

**Rysunek 2.5.** Procent organizacji z biurem zarządzania projektami



Źródło: *Pulse of the Profession*, PMI 2017

Rysunek 2.6. Rezultaty projektów



Źródło: *Pulse of the Profession*, PMI 2017

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

## 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% respondentów zgłasza, że projekty, w których uczestniczyli, zakończyły się porażką.<sup>24</sup> 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.<sup>25</sup> 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,

<sup>23</sup> Praweńska-Skrzypek, *Zarządzanie projektami jako szansa i wyzwanie dla administracji publicznej*.

<sup>24</sup> Arras People. *Project Management Benchmark Report*. 2010.

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

## 2. Charakterystyka projektów

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
<b>Sukces</b>	29%	27%	31%	28%	29%
<b>Wyzwania</b>	49%	56%	50%	55%	52%
<b>Niepowodzenie</b>	22%	17%	19%	17%	19%

Źródło: *CHAOS Report*, Standish Group 2015

**Tabela 2.2.** Podsumowanie wyników badania CHAOS

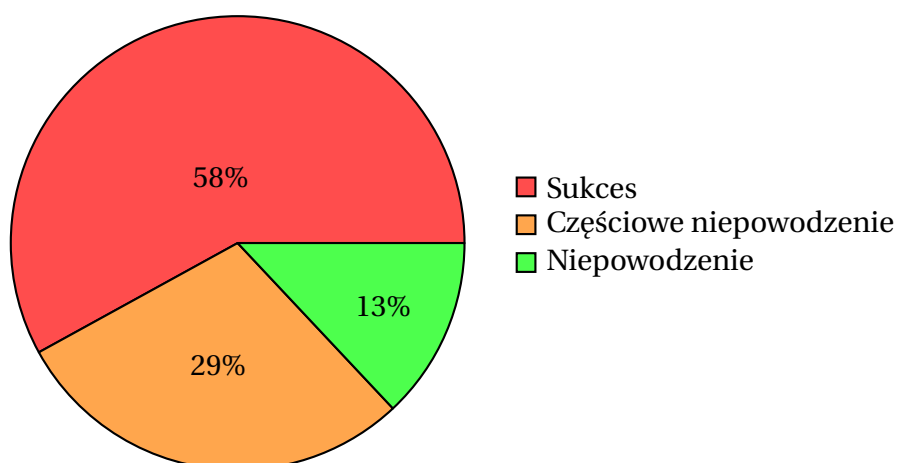
	Duże	Średnie	Małe
<b>Średni koszt rozwoju</b>	\$2,322,000	\$1,331,000	\$434,000
<b>Średnie przekroczenie kosztów</b>	189%	178%	112%
<b>Średnie przekroczenie harmonogramu</b>	222%	220%	239%
<b>Zachowane oryginalne funkcje</b>	42%	65%	74%
<b>Projekty zakończone sukcesem</b>	9.6%	16.2%	28.0%
<b>Projekty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

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

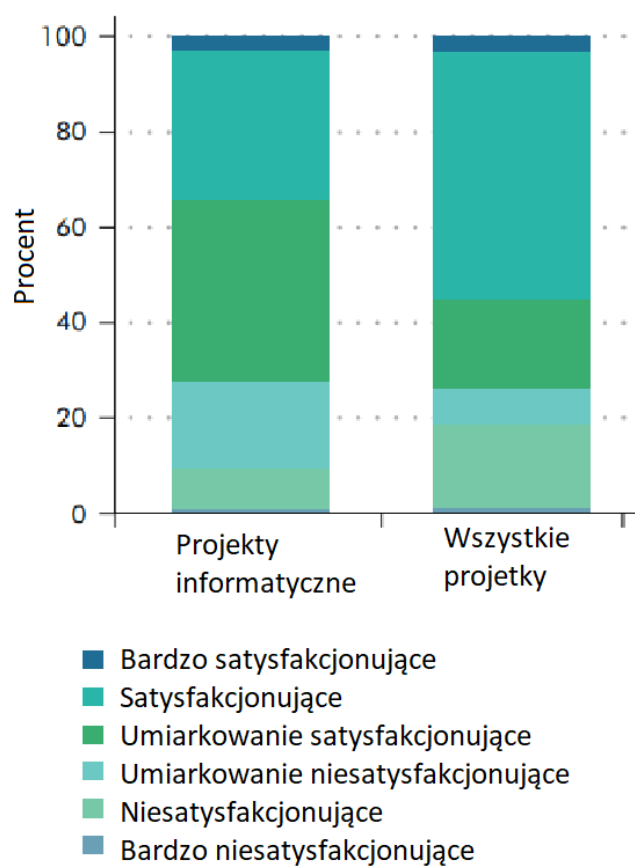
<sup>26</sup> World Bank Group. *World Development Report*. 2016.

**Rysunek 2.7.** Wskaźnik sukcesu dużych projektów informatycznych w sektorze publicznym



Źródło: *World Development Report*, World Bank Group 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.<sup>27</sup> Badania branżowe<sup>28</sup> oraz raporty rządowe<sup>29,30</sup> wywołują 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.<sup>31</sup> 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, połączona z dynamicznym rozwojem branży i mnogością technologii, powoduje, że projekty informatyczne znacząco się wyróżniają na tle reszty dziedzin.<sup>32</sup>

---

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

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

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

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

<sup>31</sup> Andrew Gemino, Blaize Reich i Chris 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.

<sup>32</sup> Kathy Schwalbe. *Information Technology Project management*. Cengage, 2018.



### 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.<sup>33</sup> Thomas i Hunt w swojej książce zawarli ważne zdanie na ten temat: "programiści pomagają użytkownikom zrozumieć, czego użytkownicy chcą".<sup>34</sup> 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, 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.<sup>35</sup>

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

---

<sup>33</sup> Witold Chmielarz. *Information Technology Project management*. Wydawnictwo Naukowe Wydziału Zarządzania, 2015.

<sup>34</sup> David Thomas i Andrew Hunt. *Pragmatyczny programista Od czeladnika do mistrza*. Helion, 2021.

<sup>35</sup> Jack T. Marchewka. *Information Technology Project management*. Northern Illinois University, 2003.

wdrożone” bywa pułapką utrudniającą odpowiedź na zmieniającą się rzeczywistość – zarówno wirtualną, jak i realną.<sup>36</sup>

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.<sup>37</sup>

#### 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.<sup>38</sup>

##### 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.<sup>39</sup> Chociaż cykle życia projektów mogą się różnić w zależności od branży lub projektu, wszystkie cykle życia projektów będą miały początek, środek i koniec.<sup>40</sup> Cykl życia projektu (PLC) jest to ciąg faz, przez które projekt przechodzi od rozpoczęcia po zakończenie.<sup>41</sup> 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.<sup>42</sup>

---

<sup>36</sup> Chmielarz, *Information Technology Project management*.

<sup>37</sup> Marchewka, *Information Technology Project management*.

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

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

<sup>40</sup> James Rosenau. *Successful Project Management*. John Wiley & Sons Australia, Limited, 1998.

<sup>41</sup> Project Management Institute. *PMBOK Guide – Seventh Edition*. 2021.

<sup>42</sup> Schwalbe, *Information Technology Project management*.

### 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.<sup>43</sup>

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

Większość specjalistów IT zna koncepcję cyklu życia produktu, szczególnie w przypadku rozwoju oprogramowania.<sup>45</sup> 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.<sup>46</sup> SDLC określa logiczną kolejność etapów rozwoju systemu oraz decyduje o przejściu od jednej czynności do kolejnej.<sup>47</sup>

Podstawowe fazy SDLC to

- planowanie,
- analiza,
- projektowanie,
- wdrożenie,
- utrzymanie i wsparcie.<sup>48</sup>

### 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

---

<sup>43</sup> Project Management Institute. *PMBOK Guide – Sixth Edition*. 2017.

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

<sup>45</sup> Schwalbe, *Information Technology Project management*.

<sup>46</sup> Marchewka, *Information Technology Project management*.

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

<sup>48</sup> Marchewka, *Information Technology Project management*.

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.<sup>49</sup>

#### 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.<sup>50</sup> 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.<sup>51</sup> Każdy system oprogramowania posiada unikalne cechy odzwierciedlające potrzeby interesariuszy – zarówno biznesowe, jak i techniczne. Odpowiedni cykl życia powinien uwzględniać wszystkie te wymagania.<sup>52</sup> Wybór modelu SDLC powinien być dokonywany na podstawie wymagań projektu.<sup>53</sup>

PMBOK® krótko opisuje pięć cykli życia produktu lub rozwoju. Dwa czynniki są istotne przy wyborze odpowiedniego cyklu życia: stopień zmienności wymagań oraz częstotliwość dostarczania użytecznych wyników. Na przykład, dla produktu o niskim stopniu zmienności wymagań i niskiej częstotliwości dostarczania, odpowiedni będzie cykl życia predykcyjny.<sup>5455</sup>

- **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).<sup>5657</sup>
- **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.<sup>585960</sup> Długość iteracji definiuje się dla każdego projektu.<sup>61</sup>

---

<sup>49</sup> Marchewka, *Information Technology Project management*.

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

<sup>51</sup> Adel Alshamrani i Abdullah 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.

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

<sup>53</sup> Ritika Arora i Neha Arora. „Analysis of SDLC models”. W: *International Journal of Current Engineering and Technology* 6.1 (2016), s. 268–272.

<sup>54</sup> Institute, *PMBOK Guide – Sixth Edition*.

<sup>55</sup> Schwalbe, *Information Technology Project management*.

<sup>56</sup> Institute, *PMBOK Guide – Fourth Edition*.

<sup>57</sup> Schwalbe, *Information Technology Project management*.

<sup>58</sup> Ian Sommerville. *Software Engineering*. 10th. Pearson, 2015.

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

<sup>60</sup> Institute, *PMBOK Guide – Sixth Edition*.

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

**Rysunek 3.1.** Porównanie cyklu życia oprogramowania do cyklu życia projektu

To podejście najlepiej sprawdza się w przypadku wysokiego stopnia zmienności i niskiej częstotliwości dostarczania.<sup>62</sup>

- **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.<sup>63</sup><sup>64</sup><sup>65</sup> To podejście najlepiej sprawdza się w przypadku niskiego stopnia zmienności i wysokiej częstotliwości dostarczania.<sup>66</sup>
- **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.<sup>67</sup><sup>68</sup>
- **Cykl życia hybrydowy:** Jest to kombinacja podejścia adaptacyjnego i predykcyjnego. 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.<sup>69</sup>

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.<sup>70</sup> 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.<sup>71</sup> 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

---

<sup>62</sup> Schwalbe, *Information Technology Project management*.

<sup>63</sup> Sommerville, *Software Engineering*.

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

<sup>65</sup> Institute, *PMBOK Guide – Sixth Edition*.

<sup>66</sup> Schwalbe, *Information Technology Project management*.

<sup>67</sup> Institute, *PMBOK Guide – Sixth Edition*.

<sup>68</sup> Schwalbe, *Information Technology Project management*.

<sup>69</sup> Institute, *PMBOK Guide – Seventh Edition*.

<sup>70</sup> Sommerville, *Software Engineering*.

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

stabilności wymagań. Wymagania określają potrzebne informacje, funkcje, zachowanie, wydajność oraz interfejsy.<sup>72</sup>

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.<sup>73</sup>

Pierwszy formalny opis modelu kaskadowego często przypisuje się artykułowi opublikowanemu w 1970 roku przez Winstona W. Royce'a (1929–1995),<sup>74</sup> 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.<sup>75</sup>

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

#### **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.<sup>77</sup>

<sup>72</sup> Arora i Arora, „Analysis of SDLC models”.

<sup>73</sup> Velmourougan 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).

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

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

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

<sup>77</sup> Arora i Arora, „Analysis of SDLC models”.



Rysunek 3.2. Enter Caption

#### 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.<sup>78</sup>

Manifest Agile<sup>79</sup> 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.<sup>80</sup>

Agile wyraźnie rozróżnia wartości i zasady (np. ciągle 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.<sup>81</sup>

Możemy postrzegać metody zwinne, takie jak Extreme Programming (XP) i Scrum, jako reakcję na modele predykcyjne, które kładą nacisk na "zracjonalizowane, inżynierskie podejście",<sup>82</sup> obejmujące szczegółowe planowanie, sformalizowane procesy oraz rygo-

---

<sup>78</sup> Arora i Arora, „Analysis of SDLC models”.

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

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

<sup>81</sup> Tamże.

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



rystyczne wykorzystanie komponentów.<sup>83</sup> W przeciwieństwie do nich, metody zwinne odpowiadają na wyzwania nieprzewidywalnego świata, podkreślając wartość, jaką ludzie i ich relacje wnoszą w rozwój oprogramowania.<sup>84</sup><sup>85</sup>

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”.<sup>86</sup> 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.<sup>87</sup>

#### **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**

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

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

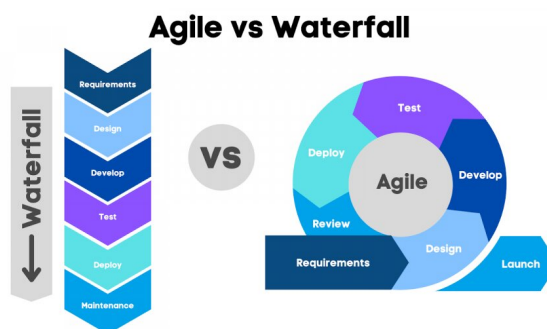
<sup>84</sup> Tore Dyba i Torgeir Dingsoyr. „What do we know about agile software development?” W: *IEEE software* 26.5 (2009), s. 6–9.

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

<sup>86</sup> Bertrand Meyer. „Agile”. W: *The good, the hype and the ugly*. Switzerland: Springer International Publishing (2014).

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

<sup>88</sup> Arora i Arora, „Analysis of SDLC models”.



Rysunek 3.3. Enter Caption

#### 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 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 również szybkich procesów zarządzania projektami IT.<sup>89</sup>

#### 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:

---

<sup>89</sup> Schwalbe, *Information Technology Project management*.

- 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.<sup>90</sup> Nawet technicznie doskonałe inicjatywy często upadają z powodu jej braku.<sup>91</sup>

#### 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 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.<sup>92</sup>

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.<sup>93</sup>

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

<sup>90</sup> Mbonigaba Celestin, Hategekimana Jean Paul i Niyigena 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).

<sup>91</sup> Hamid 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.

<sup>92</sup> Institute, *PMBOK Guide – Sixth Edition*.

<sup>93</sup> Institute, *PMBOK Guide – Seventh Edition*.

poświęcić ryzykom związanym z wymaganiami jakościowymi oprogramowania, takimi jak bezpieczeństwo funkcjonalne (ang. safety) czy bezpieczeństwo IT (ang. security).<sup>94</sup><sup>95</sup>

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

- 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?
  - 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 pomyśleć i 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:

---

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

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

<sup>96</sup> Schwalbe, *Information Technology Project management*.

- 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.<sup>97</sup> Zwraca on w swojej pracy 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.<sup>9899</sup> 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.<sup>100101</sup> 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

Zanim projekt się rozpocznie, kierownik projektu wyrabia sobie oczekiwania co do poziomu wsparcia ze strony najwyższego kierownictwa oraz użytkowników. Jednakże ocze-

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

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

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

<sup>100</sup> McFarlan, „Portfolio approach to information systems.”

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

kiwany 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.<sup>102</sup> 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.<sup>104105106</sup>

Drugą kategorię ryzyk stanowią zmiany wpływające na projekty, w tym zmiany celów projektu, kluczowych osób oraz warunków zewnętrznych, z którymi projekt musi się mierzyć. Są to ryzyka zmienności projektu.<sup>107</sup> Kategoria ta jest podobna do kategorii ryzyka „środowiskowego” zidentyfikowanej przez Keil.<sup>108</sup> Ta zmienność często pozostaje poza kontrolą członków zespołu projektowego, ale może mieć znaczący wpływ na wyniki projektu.<sup>109</sup>

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

---

<sup>102</sup> Standish Group. *CHAOS Report*. 2015.

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

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

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

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

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

<sup>108</sup> Keil i in., „A framework for identifying software project risks.”

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

Table 1. Summarizing Categories of Risk Factors

	Proposed risk factor categories			
	A priori risks		Emergent risks	
	Knowledge	Structure	Organizational support	Volatility
Znurd [38]	Degree of novelty	Technological complexity, project size	User involvement	Personnel changes
McFarlan [24]	Experience with the technology	Project size		Project structure
Barki et al. [2]	Expertise (lack of knowledge), technological newness (need for new software and hardware)	Application size, technical complexity	Organizational environment, expertise (user support)	Technological newness (software and hardware suppliers)
Kail et al. [19]	Misunderstanding requirements, managing user expectations, lack of knowledge, changing requirements		Top management support, user commitment, lack of user involvement, inappropriate staffing	Conflict between departments, changing scope, changing requirements

(continues)

**Rysunek 3.4.** Gemino, Reich i Sauer, „A Temporal Model of Information Technology Project Performance”

Table 1. Continued

	Proposed risk factor categories			
	A priori risks		Emergent risks	
	Knowledge	Structure	Organizational support	Volatility
Ropponen and Lyytinen [28]	Requirement management risks	System functionality risks	Resource usage and performance risks, personnel management risks	Scheduling and timing risks, subcontracting risks, personnel management risks
Wallace et al. [35]	Team risk, requirements risk	Complexity risks, planning and control risks	Organizational environment risk (organizational support), user risks	Organizational environment risks (politics and stability)
Standish Group [17]	Experienced project manager, standard software infrastructure, firm requirements	Minimize scope, reliable estimates	Top management support, user involvement	

**Rysunek 3.5.** Gemino, Reich i Sauer, „A Temporal Model of Information Technology Project Performance”



### 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.<sup>110</sup>

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.<sup>111</sup>

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 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.<sup>112</sup>

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

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

<sup>111</sup> Marchewka, *Information Technology Project management*.

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

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

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

#### 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.<sup>115</sup>

##### 3.8.1. Globalizacja

Globalizacja to termin powszechnie używany, lecz o niejasnym znaczeniu, nawet wśród tych, którzy się nim posługują.<sup>116</sup> 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.<sup>117</sup>

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.<sup>118</sup>

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

- **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.<sup>119</sup>

##### 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

---

<sup>115</sup> Schwalbe, *Information Technology Project management*.

<sup>116</sup> Roland Robertson i Kathleen E White. „What is globalization?” W: *The Blackwell companion to globalization* (2007), s. 54–66.

<sup>117</sup> Anthony McGrew. „A Global Society”. W: *Modernity and Its Futures*. Red. Stuart Hall, David Held i Anthony McGrew. Cambridge: Polity Press, 1990.

<sup>118</sup> Schwalbe, *Information Technology Project management*.

<sup>119</sup> Tamże.

macierzystych.<sup>120</sup> 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.<sup>121 122 123</sup>

W związku z rosnącym wykorzystaniem outsourcingu w projektach IT, kierownicy projektów muszą lepiej zrozumieć globalne i zamówieniowe aspekty, w tym pracę z zespołami zdalnymi.<sup>124</sup>

#### 3.8.3. Zespoły zdalne

Globalizacja projektów zwiększyła zapotrzebowanie na zespoły zdalne,<sup>125</sup> 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.<sup>126</sup>

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.<sup>127</sup>

#### Zalety zespołów wirtualnych

<sup>120</sup> Tamże.

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

<sup>122</sup> Rosabeth Moss Kanter. „Power Failure in Management Circuits”. W: *Harvard Business Review* 57.4 (lip. 1979), s. 65–75.

<sup>123</sup> Henry Mintzberg. *Power In and Around Organizations*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

<sup>124</sup> Schwalbe, *Information Technology Project management*.

<sup>125</sup> Institute, *PMBOK Guide – Sixth Edition*.

<sup>126</sup> Schwalbe, *Information Technology Project management*.

<sup>127</sup> Nader Ale Ebrahim, Shamsuddin Ahmed i Zahari Taha. „Virtual teams: A literature review”. W: *Australian journal of basic and applied sciences* 3.3 (2009), s. 2653–2669.

- **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.

## 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.<sup>128</sup> 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.<sup>129</sup> 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.<sup>130</sup> 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 celu<sup>131</sup>

### 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:

- 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

<sup>128</sup> Turner, *Gower Handbook of Project Management*.

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

<sup>130</sup> John M. Nicholas i Herman Steyn. *Project Management for Business, Engineering, and Technology*. Elsevier, 2012.

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

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.<sup>132</sup> 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, potwierdzenie ciągłej potrzeby realizacji projektu, nadzorowanie wydatków na zasoby oraz porównanie przewidywanej wartości z poniesionymi kosztami.<sup>133 134</sup>
- 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 więk-

---

<sup>132</sup> Rodney Turner i Ralf 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.

<sup>133</sup> Daniel Roman. *Science, Technology, and Innovation: A Systems Approach*. Grid Publishing, 1980.

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

szych 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.<sup>135</sup>

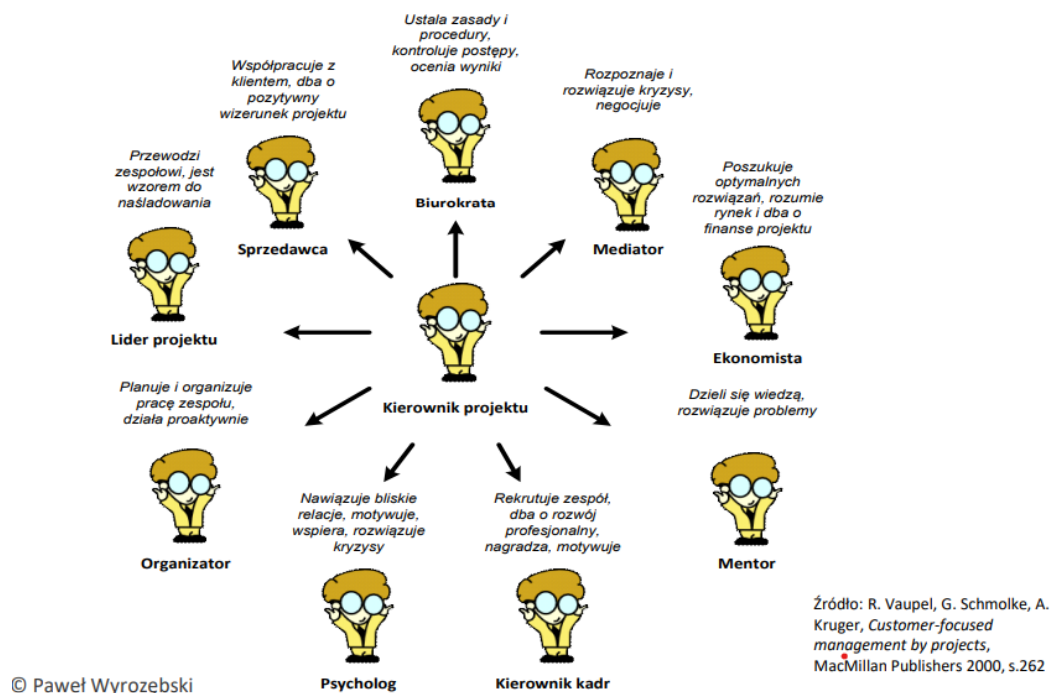
Ciekawa interpretacja roli oraz zadań kierownika projektu została również przedstawiona na rysunku 3.1. Autorzy zwracają większą uwagę na rolę interpersonalną kierownika projektu i określają go w tym kontekście jako mentora, mediatora, psychologa i lidera.

#### 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.<sup>136</sup> Na rysunkach 3.2. oraz 3.3. zostały przedstawione wyniki dostępnych badań. Na podstawie analizy dostępnej literatury i

<sup>135</sup> Nicholas i Steyn, *Project Management for Business, Engineering, and Technology*.

<sup>136</sup> Michael 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).



**Rysunek 4.1.** Źródło: R. Vaupel, G. Schmolke, A. Kruger, *Customer-focused management by projects*, MacMillan Publishers 2000, s.262

badan w dalszych sekcjach według ważności zostały wymienione kluczowe kompetencje:<sup>137138139140141142</sup>

### 4.3.1. Komunikacja

W najnowszych badaniach oraz pracach kluczowe okazują się umiejętności interpersonalne.<sup>143</sup> 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 komu-

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

<sup>138</sup> Jeferson Carvalho 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.

<sup>139</sup> People, *Project Management Benchmark Report*.

<sup>140</sup> Paul Ziek i J. Dwight Anderson. „Communication, dialogue and project management”. W: *International Journal of Managing Projects in Business* (2015). DOI: 10.1108/IJMPB-04-2014-0034.

<sup>141</sup> Jennifer M. Brill, M. J. Bishop i Andrew 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.

<sup>142</sup> Arras People. *Project Management Benchmark Report*. 2015.

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



ALL	Prog. Mgr's	Change Mgr's	Project Mgr's	Support
Planning and Organising	Leadership	Planning and Organising	Planning and Organising	Planning and Organising
Communication skills	Planning and Organising	Communication skills	Communication skills	Communication skills
Leadership	Relationship Management	Adaptability	Relationship Management	Adaptability
Relationship Management	Communication skills	Relationship Management	Adaptability	Technical
Adaptability	Commercial Awareness	Leadership	Leadership	Proactive/Initiative
Proactive/Initiative	Influencing	Influencing	Technical	Team Working
Technical - Methods / Tools	Adaptability	Proactive/Initiative	Decision Making	Relationship Management
Influencing	Decision Making	Technical	Team Working	Influencing
Decision Making	Proactive/Initiative	Decision Making	Proactive/Initiative	Leadership
Team Working	Technical	Organisational Awareness	Influencing	Concern for accuracy
Commercial Awareness	Team Working	Commercial Awareness	Commercial Awareness	Organisational Awareness
Concern for accuracy	Organisational Awareness	Team Working	Concern for accuracy	Decision Making
Organisational Awareness	Concern for accuracy	Concern for accuracy	Organisational Awareness	Commercial Awareness

PPM Competencies by Role

**Rysunek 4.2.** Źródło: Arras People Project Management Benchmark Report 2010

Rank	Keyword (Variable)	Count	% Count	Description/Example
1	industry_specific_req	1288	56%	Requires technical skills unique to industry (e.g. programming language, engineering qualifications, accounting knowledge, etc.)
2	comm_skills_req	1078	47%	Communication skills essential
3	team_manage_req	907	39%	"manage teams"
4	leadership_req	694	30%	"leader/ can lead team members"
5	degree_education_req	682	30%	Bachelor's degree
6	stakeholder_manage_req	656	28%	"Manage their expectations..."
7	budget_manage_req	654	28%	"...managing project budgets to targets..."
8	time_manage_req	653	28%	"ensure project is on time"
9	commercial_aware_req	488	21%	"business-minded/ entrepreneurial"
10	team_work_req	458	20%	"work with team in a matrix structure"

**Rysunek 4.3.** Źródło: A Multidimensional Analysis of ProjectManager Competences Maxwell Chipulu, Jun Guan Neoh, Udechukwu (Udi) Ojiako, and Terry Williams 2013

nikacyjne pozwalają kierownikom projektów być skutecznymi i mieć wpływ na sukces projektu.<sup>144</sup>

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.<sup>145</sup>

##### 4.3.2. Przywództwo

Podążając za Burnsem, przywództwo jest „jednym z najczęściej obserwowanych i najmniej rozumianych zjawisk na ziemi”.<sup>146</sup> 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.<sup>147</sup>

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.<sup>148</sup> W badaniu Alvarenga cechy przywódcze zostały uznane za najważniejszą kompetencję u kierownika projektu.<sup>149</sup>

##### 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 zawodowego.<sup>150</sup> Samoświadomość oraz odporność emocjonalna są również podkreślane jako kluczowe cechy kierownika projektu w pracy Turnera i Mullera. Według

---

<sup>144</sup> Alvarenga i in., „The project manager core competencies to project success”.

<sup>145</sup> Ziek i Anderson, „Communication, dialogue and project management”.

<sup>146</sup> J. M. Burns. *Leadership*. Harper & Row. 1978.

<sup>147</sup> Amin Akhavan Tabassi i in. „Leadership Competences of Sustainable Construction Project Managers”. W: *Journal of Cleaner Production* (2016). DOI: 10.1016/j.jclepro.2016.02.076.

<sup>148</sup> Nicholas i Steyn, *Project Management for Business, Engineering, and Technology*.

<sup>149</sup> Alvarenga i in., „The project manager core competencies to project success”.

<sup>150</sup> Tamże.

ich badań samoświadomość jest zwłaszcza szczególnie ważna w projektach o średniej złożoności.<sup>151</sup>

#### 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.<sup>152</sup> 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.<sup>153</sup> Odpowiednie wykształcenie pojawiło się również w wynikach badań IEEE, aż 30% respondentów uznało stopień naukowy z dziedziny zarządzania za istotną cechę kierownika projektu.<sup>154</sup>

#### 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.<sup>155156</sup> 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.<sup>157</sup> 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 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).<sup>158</sup>

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

<sup>152</sup> Petter Gottschalk i Jan Terje Karlsen. „A comparison of leadership roles in internal IT projects versus outsourcing projects.” W: *Industrial Management & Data Systems* (2005). DOI: 10.1108/026355705106332203.

<sup>153</sup> 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).

<sup>154</sup> Chipulu i in., „A Multidimensional Analysis of Project Manager Competences”.

<sup>155</sup> People, *Project Management Benchmark Report*.

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

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

<sup>158</sup> Nicholas i Steyn, *Project Management for Business, Engineering, and Technology*.

### 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.<sup>159</sup>

Z takim postrzeganiem polemizuje jednak badanie przeprowadzone wśród brazylijskich kierowników projektów informatycznych.<sup>160</sup> 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.

161

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."

---

<sup>159</sup> Nicholas i Steyn, *Project Management for Business, Engineering, and Technology*.

<sup>160</sup> Cíntia Cristina Silva de Araújo i Cristiane Drebes Pedron. „IT project manager competencies and IT project success: a qualitative study.” W: *Organisational Project Management* 2.1 (2015).

<sup>161</sup> Tamże.

- "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.<sup>162</sup>

Podobne badania przeprowadził Stevenson wśród osób odpowiedzialnych za rekrutację kierowników projektów informatycznych.<sup>163</sup> Badanie to wykazało niską ocenę eksperytyzacji 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 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 wewnętrznie, 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.<sup>164</sup>

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

<sup>162</sup> Tamże.

<sup>163</sup> Deborah H Stevenson i Jo Ann Starkweather. „PM critical competency index: IT execs prefer soft skills”. W: *International journal of project management* 28.7 (2010), s. 663–671.

<sup>164</sup> Tamże.

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?”.

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

---

<sup>165</sup> Nicholas i Steyn, *Project Management for Business, Engineering, and Technology*.

## **5. Badanie opinii na temat wykształcenia i roli kierownika projektu informatycznego**

Ankieta

### **5.1. Grupa badawcza**

### **5.2. Zadawane pytania**

### **5.3. Wyniki badania**

### **5.4. Wnioski**

## **6. Podsumowanie**



## Bibliografia

- Agile Alliance. *Manifesto for Agile Software Development*. 2001. URL: <http://agilemanifesto.org/>.
- Alshamrani, Adel i Abdullah Bahattab. „A comparison between three SDLC models water-fall model, spiral model, and Incremental/Iterative model”. W: *International Journal of Computer Science Issues (IJCSI)* 12.1 (2015), s. 106.
- Alvarenga, Jeferson Carvalho 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, Ritika i Neha 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, Sarah. „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, Jennifer M., M. J. Bishop i Andrew 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.
- Burns, J. M. *Leadership*. Harper & Row. 1978.
- Celestin, Mbonigaba, Hategekimana Jean Paul i Niyigena 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, Maxwell i in. „A Multidimensional Analysis of Project Manager Competences”. W: *IEEE Transactions on Engineering Management* (2012). DOI: 10.1109/TEM.2012.2215330.
- Chmielarz, Witold. *Information Technology Project management*. Wydawnictwo Naukowe Wydziału Zarządzania, 2015.
- Crawford, Lynn, J. Brian Hobbs i J. Rodney 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, Marek. *Metodyka przeglądu zakresu literatury (scoping review)*. 2020.
- Dyba, Tore i Torgeir Dingsoyr. „What do we know about agile software development?” W: *IEEE software* 26.5 (2009), s. 6–9.
- Dybå, T. „Improvisation in Small Software Organizations”. W: *IEEE Software* 17.5 (2000), s. 82–87.

- Ebrahim, Nader Ale, Shamsuddin Ahmed i Zahari Taha. „Virtual teams: A literature review”. W: *Australian journal of basic and applied sciences* 3.3 (2009), s. 2653–2669.
- Farley, D. *Modern Software Engineering: Doing What Works to Build Better Software Faster*. Addison-Wesley Professional, grud. 2021.
- Gartner, Inc. *Gartner Says Worldwide IT Spending Forecast to Grow 2.4 Percent in 2017*. 2014.
- Gemino, Andrew, Blaize Reich i Chris 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, Petter i Jan Terje 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, Nicole. *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, James J., Gary Klein i Steve 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, Rosabeth Moss. „Power Failure in Management Circuits”. W: *Harvard Business Review* 57.4 (lip. 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, Nishtha, Sandra Slaughter i Tridas Mukhopadhyay. *How do Project Managers' Skills Affect Project Success in IT Outsourcing?* 2008.
- Loh, Lawrence i Natarjan 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, Jack T. *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, Anthony. „A Global Society”. W: *Modernity and Its Futures*. Red. Stuart Hall, David Held i Anthony McGrew. Cambridge: Polity Press, 1990.
- Meyer, Bertrand. „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, Henry. *Power In and Around Organizations*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
- Mumford, Michael 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, John M. i Herman 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, Marzena i Magdalena Pichlak. *Analysis of project managers' leadership competencies Project success relation: what are the competencies of polish project leaders?* 2018.
- Pollack, Julien, Jane Helm i Daniel 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.
- Powell, Philip. „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, Roland i Kathleen E White. „What is globalization?” W: *The Blackwell companion to globalization* (2007), s. 54–66.
- Roman, Daniel. *Science, Technology, and Innovation: A Systems Approach*. Grid Publishing, 1980.
- Rosenau, James. *Successful Project Management*. John Wiley & Sons Australia, Limited, 1998.
- Royce, Winston 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, Kathy. *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, Cíntia Cristina i Cristiane Drebes Pedron. „IT project manager competencies and IT project success: a qualitative study.” W: *Organisational Project Management* 2.1 (2015).
- Sommerville, Ian. *Software Engineering*. 10th. Pearson, 2015.
- Sommerville, James, Nigel Craig i Julie Hendry. „The role of the project manager: All things to all people?” W: *Structural Survey* (2010). DOI: 10.1108/02630801011044235.
- Stevenson, Deborah H i Jo Ann Starkweather. „PM critical competency index: IT execs prefer soft skills”. W: *International journal of project management* 28.7 (2010), s. 663–671.
- Suburayan, Velmourougan, 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, Amin Akhavan i in. „Leadership Competences of Sustainable Construction Project Managers”. W: *Journal of Cleaner Production* (2016). DOI: 10.1016/j.jclepro.2016.02.076.
- Thomaidis, Nikolaos S, Nikitas Nikitakos i Georgios D Dounias. „The evaluation of information technology projects: A fuzzy multicriteria decision-making approach”. W: *International Journal of Information Technology & Decision Making* (2006).
- Thomas, David i Andrew Hunt. *Pragmatyczny programista Od czeladnika do mistrza*. Helion, 2021.
- Tohidi, Hamid. „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, Rodney i Ralf 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, Kittisak i Worapat Paireekreng. *A Study of Project Manager Skills for Enterprise Data Platform Project*. 2021.
- Van Wyngaard, C., Jan-Harm Pretorius i Leon 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, Paweł 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, Paul i J. Dwight 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).

## Wykaz symboli i skrótów

**EiTI** – Wydział Elektroniki i Technik Informatycznych

**PW** – Politechnika Warszawska

**WEIRD** – ang. *Western, Educated, Industrialized, Rich and Democratic*

## 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 . . . . .	12
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 .	15
3.1	Porównanie cyklu życia oprogramowania do cyklu życia projektu . . . . .	21
3.2	Enter Caption . . . . .	24
3.3	Enter Caption . . . . .	26
3.4	Gemino, Reich i Sauer, „A Temporal Model of Information Technology Project Performance” . . . . .	31
3.5	Gemino, Reich i Sauer, „A Temporal Model of Information Technology Project Performance” . . . . .	32
4.1	Źródło: R. Vaupel, G. Schmolke, A. Kruger, Customer-focused management by projects, MacMillan Publishers 2000, s.262 . . . . .	40
4.2	Źródło: Arras People Project Management Benchmark Report 2010 . . . . .	41
4.3	Źródło: A Multidimensional Analysis of ProjectManager Competences Maxwell Chipulu, Jun Guan Neoh, Udechukwu (Udi) Ojiako, and Terry Williams 2013 .	41

## Spis tabel

2.1	Rezultaty projektów w latach 2011-2015 . . . . .	14
2.2	Podsumowanie wyników badania CHAOS . . . . .	14

## Spis załączników