

UNIwersytet Ekonomiczny w Krakowie

Kolegium Nauk o Zarządzaniu i Jakości

Katedra Procesu Zarządzania



Gabriela Kupiec

Nr. indeksu: 123456

Współczesne wyzwania w pracy menadżera projektów IT

Praca dyplomowa | wersja: 0.1.0

Pod kierunkiem
dr Krzysztof Woźniak

Kraków 2025

*Dla moich rodziców oraz najbliższych przyjaciół
za niezłomną wiarę w moje zwycięstwo.*

Spis treści

Wstęp	4
1. Rola i zadania menadżera projektów IT	5
1.1. Definicja i znaczenie zarządzania projektami w IT.....	5
1.1.1. Podstawowe założenia zarządzania projektami w środowisku tech- nologicznym	6
1.1.2. Rola zespołu projektowego w realizacji projektu.....	10
1.2. Kluczowe obowiązki i kompetencje menadżera projektów IT	12
1.2.1. Kluczowe zadania menadżera projektu IT.....	12
2. Praktyczne zastosowanie.....	18
2.1. Tworzenie obiektów LaTeXa.....	18
3. Wyniki oraz podsumowanie	22
Bibliografia.....	23
Spis tabel.....	24
Spis rysunków.....	25

Wstęp

Oto kilka słów wstępu dla początkowego rozdziału publikacji rozpoczynającego dywagację autora na temat jego pracy.

Jak widać rozdział ten nie zawiera numeracji, co zwykle jest bardzo porządaną cechą dla rozdziału wprowadzającego czytelnika w całość dokumentu.

Nastomiast jak zostało to dokonane można bez trudu podejrzeć w pierwszych 5 liniijkach kodów źródłowych pliku `chap_0_intro.tex` zamieszczonego na stronie <https://github.com/egel/latex-thesis-example>.

Zapraszam serdecznie do przejrzenia i wypróbowania niniejszego repozytorium.

Rozdział 1

Rola i zadania menadżera projektów IT

Treść dla rozdziału pierwszego zwykle zawiera w sobie teoretyczne podwaliny pod dalszą część Twojej pracy dyplomowej (czy to licencjackiej, magisterskiej lub nawet doktorskiej).

Jak również zapewne zauważyłeś drogi Czytelniku, rozdział ten posiada już normalną numerację, tak jak to powinno wyglądać w wydruku dla pracy końcowej (patrz spis treści).

Kolumna 1	Kolumna 2
wiersz1-kolumna1	wiersz1-kolumna2
wiersz2-kolumna1	wiersz2-kolumna2

Tabela 1.1: Prosta tabelka dla przykładu

1.1. Definicja i znaczenie zarządzania projektami w IT

Jak widać można zagnieżdżać treści w wygodne sekcje (ang. *sections*) tak jak również w dowolnym momencie umieszczać rysunki lub fotografie — tak jak to widać na rysunku ??.

Czasem zdarzy się również tak, że przeniesie zdjęcie na kolejną stronę, jednak w pewnych okolicznościach to celowy zabieg który wykonuje za nas kompilator LaTeXa — Super! :)

Wtedy można bez kołototu odwołać się do konkretnego zdjęcia, tabeli, kodu źródłowego, przypisu bibliograficznego, ect. poprzez referencję (komendę `\ref{?}`) i podanie zamiast

znaku zapytania odwołania, czyli tzw. label-ki). Całość opisaną powyżej można odnaleźć w kodzie źródłowym do niniejszego rozdziału. Tak, to na prawdę jest, aż takie proste :)

1.1.1. Podstawowe założenia zarządzania projektami w środowisku technologicznym

Projekt to tymczasowe przedsięwzięcie podejmowane w celu stworzenia unikalnego produktu, usługi lub rezultatu (PMI, 2017). Cechami charakterystycznymi projektu są: czasowość (posiada wyraźny początek i koniec), unikalność (tworzy coś nowego lub znacząco różnego od istniejących rozwiązań) oraz orientacja na cel (skoncentrowanie na osiągnięciu konkretnych rezultatów). W odróżnieniu od procesów operacyjnych, które mają charakter powtarzalny i służą utrzymaniu bieżącej działalności organizacji, projekty mają charakter jednorazowy i są zorientowane na zmianę. Przykładowo, codzienne utrzymanie systemu IT to proces operacyjny, natomiast wdrożenie nowego systemu CRM – to projekt (Kerzner, 2017). Cykl życia projektu określa logiczną sekwencję etapów od jego rozpoczęcia do zakończenia. W przypadku projektów technologicznych typowy cykl życia obejmuje kilka faz (inicjacja, planowanie, realizacja, monitorowanie i kontrola oraz zamknięcie). Każdy z tych etapów ma określone konkretne zadania i działania. [Tabela] Faza inicjacji to moment, w którym identyfikuje się potrzebę realizacji projektu, ocenia jego wykonalność i definiuje wstępny zakres. Efektem tej fazy jest uzasadnienie biznesowe (business case) oraz karta projektu (project charter), które zatwierdzają rozpoczęcie prac (PMI, 2017). Faza planowania to przygotowanie szczegółowego planu działania. W projektach IT często stosuje się tutaj narzędzia takie jak JIRA, MS Project czy Confluence do współdzielenia planu z zespołem. Realizacja polega na wdrażaniu zaplanowanych działań, tworzeniu oprogramowania, testowaniu i stopniowym dostarczaniu produktu. Często odbywa się to z wykorzystaniem podejść zwinnych (np. Scrum, Kanban), które umożliwiają iteracyjne rozwijanie funkcjonalności i częste prezentowanie efektów interesariuszom. Faza monitorowania i kontroli występuje równolegle z realizacją, polega na śledzeniu postępów, mierze efektywności i wprowadzaniu korekt. Faza zamknięcia kończy projekt. Efektywne zamknięcie pozwala na uczenie się organizacji i wprowadzanie dobrych praktyk w kolejnych inicjatywach. Interesariusze (ang. stakeholders) to osoby lub podmioty, które mają wpływ na projekt lub są pod jego wpływem. W projektach IT do kluczowych interesariuszy zaliczają się: klienci i użytkownicy końcowi, którzy definiują potrzeby i oczekiwania; zespół projektowy, który realizuje zadania techniczne; sponsor projektu, który zapewnia finansowanie i wsparcie strategiczne, oraz menedżerowie produktu i projektów, którzy ko-

Faza	Działania
Inicjacja	<p>Analiza wykonalności – ocena, czy projekt ma sens pod kątem technicznym, ekonomicznym i organizacyjnym.</p> <p>Zbieranie wymagań biznesowych – spotkania z interesariuszami, identyfikacja potrzeb, oczekiwań i ograniczeń.</p> <p>Analiza ryzyka – wstępna identyfikacja potencjalnych zagrożeń i szans dla projektu.</p>
Planowanie	<p>Tworzenie harmonogramu (np. metodą CPM, narzędzia Gantt) – określenie kolejności i czasu trwania zadań.</p> <p>Planowanie zasobów – przypisanie osób, sprzętu i budżetu do konkretnych zadań.</p> <p>Budżetowanie – szacowanie i alokacja kosztów.</p> <p>Zarządzanie zakresem i ryzykiem – opracowanie strategii reagowania na niepewność.</p>
Realizacja	<p>Codzienne stand-upy (daily meetings)</p> <p>Sprinty i inkrementy</p> <p>Systematyczne testy i kontrola jakości</p>
Monitorowanie	<p>Pomiar postępów względem harmonogramu i budżetu</p> <p>Zarządzanie zmianami – formalne wprowadzanie zmian w zakresie lub wymaganiach projektu.</p> <p>Raportowanie – informowanie interesariuszy o statusie projektu.</p>
Zamknięcie	<p>Ewaluacja projektu – sprawdzenie, czy zostały osiągnięte cele oraz został zachowany budżet (kompleksowa ocena projektu).</p> <p>Retrospektywa zespołu</p> <p>Przekazanie produktu/usługi użytkownikom – wraz z dokumentacją i szkoleniami.</p> <p>Formalne zakończenie – archiwizacja dokumentacji, rozliczenie zespołu, zamknięcie umów.</p>

Tabela 1.2: Fazy zarządzania projektem i ich działania

ordynują działania i komunikację. Zarządzanie interesariuszami wymaga identyfikacji ich potrzeb i oczekiwań oraz aktywnego angażowania ich w cały cykl życia projektu (Giangregorio, 2014). Komunikacja, jasność ról i systematyczne dostarczanie wartości są kluczowe dla powodzenia projektów IT (ProductCraft, n.d.). Zarządzanie projektami IT, mimo że bazuje na tych samych ogólnych zasadach co zarządzanie projektami w innych sektorach, cechuje się wyraźną specyfiką wynikającą z charakteru środowiska technologicznego. Branża IT rozwija się niezwykle szybko, co oznacza, że technologie wykorzystywane na początku projektu mogą być przestarzałe już na jego końcu. Dlatego projekty IT wymagają dużej elastyczności i gotowości do zmian. Zespoły muszą być w stanie szybko dostosować się do nowych narzędzi, bibliotek, frameworków czy standardów rynkowych (Ries, 2011). Projekty IT są złożone pod względem struktury i wymagają ścisłej współpracy między programistami, architektami, testerami, specjalistami od UX/UI, analitykami biznesowymi oraz właścicielami produktu. Oznacza to konieczność skutecznej komunikacji pomiędzy wieloma specjalistycznymi dziedzinami. Wymaga to od kierownika projektu nie tylko zdolności organizacyjnych, ale i przynajmniej podstawowego zrozumienia technologii – co odróżnia go np. od menedżera w przemyśle budowlanym czy produkcyjnym. W projektach IT zmiany w wymaganiach są niemal nieuniknione – użytkownicy końcowi często dostrzegają rzeczywiste potrzeby dopiero podczas interakcji z prototypami. Wymusza to stosowanie podejść iteracyjnych i zwinnych (Agile), które umożliwiają częste dostarczanie funkcjonalności i zbieranie informacji zwrotnej. Zespoły IT coraz częściej pracują w modelu rozproszonym – w różnych strefach czasowych, krajach i kulturach. Oznacza to nowe wyzwania w zakresie komunikacji, zarządzania czasem, synchronizacji pracy i budowania zaufania w zespole. Zarządzanie projektami IT wymaga zatem biegłości w narzędziach wspierających zdalną współpracę (np. Slack, Miro, Confluence) oraz wrażliwości międzykulturowej (Meyer, 2014). Ważnym aspektem jest zarządzanie ryzykiem, który stanowi jeden z kluczowych elementów powodzenia projektów IT. Szybko zmieniające się środowisko technologiczne, zależność od zespołów wielofunkcyjnych, niestabilne wymagania i złożoność systemów technologicznych generują szereg unikalnych ryzyk, które wymagają skutecznej identyfikacji i zarządzania. Projekty technologiczne mogą być narażone na wiele typów ryzyk, które można sklasyfikować jak w tabeli. [tabela] Skuteczna identyfikacja i analiza ryzyka jest kluczem do skutecznego zarządzania projektem. W projektach stosuje się różne techniki identyfikacji ryzyk. Analiza SWOT, jest najczęściej wybieraną metodą i polega na identyfikacji mocnych i słabych stron projektu, a także rozpoznaniu szans i zagrożeń zewnętrznych. Kolejną techniką jest burza mózgów, w której biorą udział zespół projektowy i interesariusze. Można także oprzeć się na doświadczeniach z poprzednich projektów poprzez stworzenie listy kontrolnej ryzyk. Ważne jest też oszacowanie

Rodzaj ryzyka	Przykład
Techniczne	Niedojrzałość technologii, problemy z integracją systemów, awarie środowiska developerskiego.
Organizacyjne	Brak zaangażowania interesariuszy, brak jasno określonej roli właściciela produktu.
Finansowe	Niedoszacowanie budżetu, przekroczenie kosztów, nieoczekiwane koszty licencji.
Zarządcze i komunikacyjne	Nieefektywna komunikacja w zespołach zdalnych, błędne założenia w planie projektu.

Tabela 1.3: Rodzaje ryzyk w projektach IT

prawdopodobieństwa wystąpienia danego ryzyka oraz jego wpływu na projekt, co umożliwia ustalenie priorytetów wśród działań prewencyjnych. Zgodnie z podejściem PMBOK (2021), wyróżniamy kilka głównych strategii radzenia sobie z ryzykiem - unikanie ryzyka, minimalizacja ryzyka, przeniesienie ryzyka oraz akceptacja ryzyka. Unikanie ryzyka to jedna z najskuteczniejszych strategii, jeśli jest możliwa do zastosowania. Eliminując ryzyko u źródła, całkowicie można zapobiec jego wystąpieniu. Jednak często wiąże się z ograniczeniem innowacyjności - np. rezygnacja z nowej technologii może oznaczać utratę przewagi konkurencyjnej. Trzeba więc dobrze rozważyć, czy warto rezygnować z szansy w imię bezpieczeństwa. Minimalizacja ryzyka pozwala ograniczyć prawdopodobieństwo lub skutki ryzyka, zachowując jednocześnie możliwość realizacji pierwotnych planów. Wymaga jednak zasobów (czasu, budżetu, ludzi), co może wpłynąć na inne obszary projektu. Dobrze sprawdza się w projektach o średnim i wysokim poziomie złożoności. Przeniesienie odpowiedzialności za ryzyko może być wygodne, ale nie oznacza jego całkowitego wyeliminowania oraz często wiąże się z dodatkowymi kosztami. Akceptacja ryzyka to strategia rozsądna, jeśli ryzyko jest niskie lub koszty jego redukcji są nieproporcjonalne do ewentualnych strat. Wymaga jednak świadomego zarządzania - udokumentowania decyzji, monitorowania sytuacji i przygotowania planu awaryjnego. Menadżerowie IT wykorzystują różnorodne narzędzia, które wspomagają ich w zarządzaniu ryzykiem w projekcie. JIRA, Microsoft Project, Asana pomagają rejestrację i monitoring ryzyk projektowych. Risk Register / Risk Matrix stosuje się do dokumentacji ryzyk, oceny wpływu na projekt oraz potencjalnym planem zarządzania nimi. Analiza Monte Carlo jest to zaawansowana analiza statystyczna, pozwalająca przewidywać skutki różnych scenariuszy w harmonogramie i budżecie.

1.1.2. Rola zespołu projektowego w realizacji projektu

W zarządzaniu projektami IT zespół projektowy odgrywa kluczową rolę w realizacji celów, zapewnieniu jakości oraz skutecznym reagowaniu na zmiany. Sukces przedsięwzięcia w środowisku technologicznym w dużej mierze zależy od kompetencji, współpracy i zaangażowania członków zespołu. Zespół projektowy to interdyscyplinarna grupa specjalistów, których role są dostosowane do charakteru projektu oraz wybranej metodyki zarządzania, takiej jak Agile czy Waterfall. Każdy członek zespołu wnosi konkretną wartość i odpowiada za istotne obszary działań, które bezpośrednio wpływają na postęp oraz jakość realizacji projektu. Aby realizacja projektu przebiegała sprawnie, niezbędne jest jasne określenie ról i zakresu odpowiedzialności każdej osoby zaangażowanej w prace projektowe. Lider projektu (Project Manager) odpowiada za całościowe zarządzanie projektem: planowanie, nadzór, koordynację działań i kontakt z interesariuszami. Jego zadaniem jest także zarządzanie ryzykiem i reagowanie na zmiany. Według H. Kerznera (2017), PM pełni rolę integratora, zapewniając spójność działań zespołu i zgodność z celami strategicznymi projektu. Analityk biznesowy identyfikuje wymagania użytkowników, analizuje potrzeby klienta, przygotowuje dokumentację wymagań oraz wspiera zespół techniczny w ich interpretacji. Melissa Perri (2018) podkreśla, że analityk pełni rolę tłumacza między światem biznesu a zespołem technicznym. Programista (developer) odpowiedzialny jest za projektowanie, implementację i utrzymanie kodu źródłowego. W projektach IT często współpracuje z testerami i designerami. Tester (QA – Quality Assurance) zapewnia jakość produktu poprzez planowanie, wykonywanie i automatyzację testów, zgłaszanie błędów oraz współpracę z developerami w celu ich usunięcia. Administrator systemów / DevOps zarządza środowiskiem produkcyjnym, bezpieczeństwem i infrastrukturą systemową. Każda z powyższych ról jest krytyczna, brak jasnego przypisania zadań może prowadzić do duplikacji działań, opóźnień lub luk w odpowiedzialności. W zarządzaniu projektami kluczowe znaczenie ma nie tylko przypisanie zadań według specjalizacji, ale także uwzględnienie rzeczywistych kompetencji oraz dostępności członków zespołu. Menadżer projektów odpowiada za skuteczne zarządzanie zespołem, co obejmuje zarówno planowanie pracy, jak i monitorowanie postępów oraz zapewnianie odpowiedniego poziomu współpracy. Każdy członek zespołu ponosi odpowiedzialność za terminową realizację przydzielonych zadań, dbałość o jakość i zgodność z wymaganiami projektowymi, aktywną współpracę z innymi, a także bieżące informowanie o napotkanych problemach czy przeszkodach. Zgodnie ze standardami PMBOK (PMI, 2021), jednym z kluczowych narzędzi wspierających przejrzystość i rozliczalność w zespole jest macierz RACI, która precyzuje, kto w projekcie jest odpowiedzialny za wykonanie zadania, kto podejmuje de-

cyzje i zatwierdza działania, kto udziela konsultacji oraz kto powinien być informowany. Takie podejście umożliwia jasne zdefiniowanie ról i minimalizuje ryzyko nieporozumień w trakcie realizacji projektu. Jak zauważa Teresa Torres (2021), zespoły o wysokiej kulturze współpracy potrafią efektywniej zarządzać wiedzą, szybciej się adaptują i są bardziej odporne na zmiany personalne, co ma szczególne znaczenie w dynamicznych projektach IT. Nowoczesne zespoły IT korzystają z różnorodnych form współpracy, dostosowując je do specyfiki organizacji oraz przyjętej metodyki zarządzania projektem. W środowiskach zwinnych, takich jak Scrum czy Kanban, współpraca opiera się na iteracyjnych cyklach pracy, częstej komunikacji i elastycznym reagowaniu na zmiany. Dużą rolę odgrywa tu grupowe podejście do zadań – wspólne opracowywanie funkcjonalności, warsztaty z klientem czy planowanie sprintów sprzyjają integracji zespołu i lepszemu zrozumieniu celów biznesowych. W zespołach programistycznych często praktykuje się pracę w parach, która nie tylko poprawia jakość kodu, ale również wspiera rozwój kompetencji i budowanie wzajemnego zaufania. Codzienną współpracę ułatwiają także narzędzia cyfrowe, takie jak Jira, Trello, GitHub, Confluence czy Google Workspace, które zapewniają transparentność działań, dostęp do aktualnej dokumentacji i wspólne środowisko pracy. Przykładem skutecznego zastosowania kultury współpracy jest model organizacyjny firmy Spotify, który opiera się na autonomicznych zespołach zwanych „squads” i większych strukturach wspierających współdzielenie wiedzy (chapters i guilds). Spotify postawiło na decentralizację decyzji i silne wsparcie dla kultury feedbacku, co przyczyniło się do szybkiego skalowania i utrzymania spójności organizacyjnej mimo rosnącej liczby zespołów. Model ten został opisany przez Henrika Kniberga i Andresa Ivarssona w dokumencie „Scaling Agile @ Spotify” (2012), który do dziś stanowi jedno z najczęściej cytowanych źródeł w obszarze zwinnego zarządzania zespołami. Efektywna komunikacja to jeden z kluczowych filarów sukcesu projektów IT. Sprawny przepływ informacji między członkami zespołu pozwala unikać nieporozumień, szybciej rozwiązywać problemy oraz dostarczać produkty wyższej jakości. W dobie pracy zdalnej i zespołów rozproszonych jej znaczenie jest jeszcze większe, z powodu braku bezpośredniego kontaktu wymaga bardziej świadomego i zaplanowanego podejścia do komunikacji. Podstawowym narzędziem synchronizacji działań są regularne spotkania projektowe, które umożliwiają wymianę informacji, wspólne planowanie i bieżące rozwiązywanie trudności. W metodykach zwinnych, takich jak Scrum, każde spotkanie ma jasno określoną funkcję i cel. Codzienne stand-upy to krótkie, maksymalnie 15-minutowe spotkania, podczas których każdy członek zespołu dzieli się informacją o tym, co robił, co planuje oraz jakie napotkał przeszkody. Jak zauważa Giangregorio (2020), tego typu rytuał zwiększa indywidualną odpowiedzialność oraz pozwala na szybką reakcję w przypadku pojawiających się problemów. Kolejnymi ważnymi wydarzeniami

są Sprint Planning — planowanie zakresu prac na nadchodzącą iterację, Retrospektywa — analiza współpracy i wskazanie obszarów do poprawy, oraz Review/Demo — prezentacja postępów przed klientem i zebranie informacji zwrotnej. Warto także pamiętać o znaczeniu nieformalnych kontaktów, zwłaszcza w zespołach zdalnych. Organizowanie okazjonalnych spotkań integracyjnych, nawet online, pomaga budować relacje, poprawia atmosferę i wzmacnia zaufanie w zespole, co przekłada się bezpośrednio na efektywność pracy. W projektach IT korzysta się z różnych narzędzi wspierających komunikację i zarządzanie pracą, które także zostały wspomniane wcześniej. Platformy takie jak Slack czy Microsoft Teams umożliwiają szybką komunikację tekstową, wideokonferencje oraz integrację z innymi systemami. Narzędzia typu Jira i Trello służą do śledzenia postępów, zarządzania backlogiem i zadaniami, natomiast Confluence, Notion czy Google Workspace pozwalają na współdzielenie dokumentacji i wiedzy. Dodatkowo Miro i Figma wspierają współpracę wizualną, ułatwiając mapowanie procesów czy projektowanie interfejsów. Według badań McKinsey, zespoły korzystające z nowoczesnych narzędzi komunikacyjnych zwiększają swoją produktywność nawet o 20–25%. Transparentność komunikacji oraz dostęp do kluczowych informacji projektowych mają fundamentalne znaczenie dla sukcesu zespołu. Ważne jest, aby statusy zadań i postępy były udostępniane wszystkim członkom zespołu, co pozwala na bieżące monitorowanie realizacji prac. Kluczowe jest także jasne definiowanie oczekiwań, terminów i kryteriów akceptacji, które pomagają uniknąć nieporozumień. Przejrzyste raportowanie problemów i wszelkich zmian umożliwia szybką reakcję na trudności. Nie mniej istotna jest archiwizacja dokumentów, co jest szczególnie ważne przy rotacji członków zespołu lub w przypadku długofalowych projektów. Jak wskazuje Roman Pichler (2016), brak dostępu do informacji oraz słaba komunikacja należą do najczęstszych przyczyn niepowodzeń projektów IT.

1.2. Kluczowe obowiązki i kompetencje menadżera projektów IT

1.2.1. Kluczowe zadania menadżera projektu IT

Planowanie projektu stanowi jeden z kluczowych etapów zarządzania projektem IT, a odpowiedzialność za jego prawidłowe przeprowadzenie spoczywa na menadżerze projektu. Jest to moment decydujący o sukcesie całego przedsięwzięcia, ponieważ błędy lub niedociągnięcia na tym etapie mogą skutkować znacznymi kosztami oraz opóźnieniami w dalszych fazach realizacji. Podstawą skutecznego planowania jest precyzyjne określenie

celów projektu, które powinny spełniać kryteria SMART, czyli być konkretne, mierzalne, osiągalne, istotne oraz określone w czasie. Menadżer projektu, we współpracy z interesariuszami, przygotowuje kartę projektu, zawierającą najważniejsze założenia, cele oraz zakres przedsięwzięcia. [Marty Cagan, *Inspired: How to Create Tech Products Customers Love* (Portland: SVPG Press, 2018), 45–47.] Następnie, istotnym elementem planowania jest opracowanie harmonogramu, który obejmuje identyfikację głównych etapów projektu, przypisanie konkretnych zadań oraz ustalenie terminów ich realizacji. Struktura harmonogramu różni się w zależności od wybranej metodyki zarządzania projektem – w podejściu tradycyjnym, takim jak Waterfall, wykorzystywane są struktury podziału pracy (WBS), diagramy Gantta oraz techniki sieciowe (CPM, PERT), które umożliwiają wizualizację i analizę zadań w czasie, a także określenie ścieżki krytycznej. W metodykach zwinnych, na przykład Scrum, planowanie przebiega iteracyjnie w ramach sprintów, gdzie zadania organizowane są w backlogu produktowym i sprintowym. [Roman Pichler, *Strategize: Product Strategy and Product Roadmap Practices for the Digital Age* (Boston: Addison-Wesley, 2016), 75–78.] Kolejnym etapem jest budżetowanie projektu, obejmujące oszacowanie kosztów związanych z zasobami ludzkimi, licencjami oprogramowania, infrastrukturą techniczną, szkoleniami oraz rezerwami na ryzyka. Menadżer projektu dokonuje estymacji kosztów, wykorzystując metody analogowe, parametryczne oraz bottom-up, co pozwala na precyzyjne określenie wymagań finansowych. Kontrola budżetu w trakcie realizacji umożliwia porównywanie planowanych wydatków z rzeczywistymi i podejmowanie działań korygujących w przypadku odchyłeń. [Harold Kerzner, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, 12th ed. (Hoboken: Wiley, 2017), 150–155.] Wreszcie, planowanie obejmuje alokację zasobów, która polega na efektywnym przydzieleniu zarówno zasobów ludzkich, jak i technicznych. Menadżer musi uwzględnić dostępność i kompetencje pracowników, ich zaangażowanie w inne projekty, potrzeby szkoleniowe, a także konieczność zatrudnienia podwykonawców. Dodatkowo należy zaplanować dostęp do środowisk testowych, sprzętu, licencji oraz narzędzi wspierających proces wytwarzania oprogramowania. Do wsparcia planowania wykorzystywane są nowoczesne narzędzia informatyczne, między innymi Microsoft Project, JIRA, Asana czy Smartsheet, które umożliwiają sprawne zarządzanie zadaniami, harmonogramem oraz zasobami projektu. [Emanuela Giangregorio, *Practical Project Stakeholder Management* (London: Kogan Page, 2019), 102–108.] //Przykład planowania projektu IT – aplikacja mobilna „FitLife” (opracowanie ChatGPT) Celem projektu jest stworzenie aplikacji mobilnej „FitLife”, która pozwala użytkownikom monitorować aktywność fizyczną, sen i nawyki żywieniowe, z możliwością integracji z urządzeniami typu smartwatch. Określenie celów projektu Główne cele projektu zostają zdefiniowane według zasady SMART:

S: Stworzyć aplikację mobilną na systemy Android i iOS, M: Minimum 5 funkcjonalności (krokomierz, monitor snu, dziennik kalorii, integracja z Apple Health/Google Fit, powiadomienia motywujące), A: Projekt realizowany w ciągu 6 miesięcy przy zespole 6-osobowym, R: Aplikacja wspiera zdrowy styl życia użytkownika, zgodnie z trendami rynkowymi, T: Gotowy MVP (minimum viable product) do testów beta w 20. tygodniu. Tworzona zostaje karta projektu z określeniem zakresu, założeń, ograniczeń i kryteriów sukcesu. Harmonogram projektu Zespół wykorzystuje metodykę Scrum (Agile) i planuje projekt w 2-tygodniowych sprintach. W pierwszej fazie tworzony jest backlog produktu zawierający główne funkcje. Następnie zespół planuje sprinty: Sprint 1-2: projektowanie UI/UX i tworzenie makiet aplikacji, Sprint 3-4: implementacja funkcji śledzenia kroków i snu, Sprint 5-6: dziennik kalorii i integracja z urządzeniami, Sprint 7-8: testy wewnętrzne i poprawki UI, Sprint 9-10: beta testy, monitoring, dokumentacja techniczna. Do wizualizacji harmonogramu używany jest JIRA z integracją Gantt Chart oraz Confluence do dokumentowania ustaleń. Budżetowanie projektu Na podstawie wcześniejszych projektów i analizy zasobów przygotowano estymację budżetową: Wynagrodzenia zespołu: 6 osób \times 6 miesięcy \times średnia stawka 18 000 zł = 648 000 zł, Koszty sprzętu i oprogramowania (licencje, testowe urządzenia mobilne): 35 000 zł, Rezerwa na nieprzewidziane koszty (ok. 10%): 68 000 zł. Całkowity budżet projektu: ok. 750 000 zł. Budżet zaplanowano z pomocą narzędzi takich jak Excel oraz Cost Management w Asanie. Alokacja zasobów Zespół projektowy: 1 \times Project Manager, 1 \times UX/UI designer, 2 \times developerzy (jeden iOS, jeden Android), 1 \times backend developer (API, baza danych), 1 \times QA tester (testy manualne i automatyczne). Menadżer projektu zaplanował wykorzystanie środowisk testowych w Firebase oraz GitLab CI/CD do automatyzacji testów i wdrożeń. Harmonogram pracy i obciążenia zasobów śledzony jest w narzędziu TeamGantt. Ten przykład pokazuje, jak menadżer projektu w środowisku IT wykorzystuje konkretne narzędzia, metodyki i procesy, by planowanie było realistyczne, elastyczne i dostosowane do specyfiki projektu technologicznego. // Zarządzanie zespołem projektowym w środowisku IT wymaga szczególnej uwagi na rekrutację odpowiednich specjalistów, budowanie zespołu, motywowanie członków oraz efektywne rozwiązywanie konfliktów. Lider projektu musi zadbać o dobór osób, których kompetencje techniczne i miękkie odpowiadają wymaganiom projektu, co jest kluczowe dla osiągnięcia założonych celów. Budowanie zespołu to nie tylko skompletowanie grupy fachowców, ale również tworzenie atmosfery współpracy i zaufania, co sprzyja kreatywności oraz efektywności pracy. [Camille Fournier, *The Manager's Path: A Guide for Tech Leaders Navigating Growth and Change* (O'Reilly Media, 2017).] Motywowanie zespołu obejmuje zarówno aspekty finansowe, jak i niematerialne, takie jak uznanie, rozwój zawodowy czy możliwość wpływu na decyzje projektowe. [Marty

Cagan, *Inspired: How to Create Tech Products Customers Love* (SVPG Press, 2018).] Rozwiązywanie konfliktów w projekcie IT wymaga od menadżera umiejętności komunikacyjnych i mediacyjnych, co pozwala na szybkie eliminowanie nieporozumień i zapobiega eskalacji problemów. [Kim Scott, *Radical Candor: Be a Kick-Ass Boss Without Losing Your Humanity* (St. Martin's Press, 2017).] Istotnym elementem skutecznego zarządzania zespołem jest ścisła współpraca i jasne przypisanie odpowiedzialności za realizację poszczególnych zadań. Każdy członek zespołu powinien znać swoje obowiązki oraz mieć możliwość konsultacji i współdziałania z innymi specjalistami, co wpływa na terminowość i jakość realizacji projektu. [Martin Eriksson, Nate Walkingshaw, Richard Banfield, *Product Leadership: How Top Product Managers Launch Awesome Products and Build Successful Teams* (O'Reilly Media, 2016).] W praktyce stosuje się różnorodne metody współpracy, takie jak praca w parach, grupach roboczych czy codzienne stand-upy, które wspierają transparentność działań i pozwalają szybko reagować na pojawiające się wyzwania. [Teresa Torres, *Continuous Discovery Habits: Discover Products that Create Customer Value and Business Value* (Product Talk, 2021).] Zapewnienie dostępu do kluczowych informacji wszystkim członkom zespołu jest niezbędne do utrzymania spójności działań i realizacji celów projektu. [MindTools, "Stakeholder Analysis: PM 101," accessed May 30, 2025, https://www.mindtools.com/pages/article/newPPM_07.htm.] Monitorowanie postępów pozwala na kontrolę realizacji założonych celów oraz umożliwia szybkie reagowanie na pojawiające się problemy. Menadżer odpowiada za wdrożenie odpowiednich narzędzi i metod, które umożliwiają transparentne śledzenie przebiegu prac oraz terminów realizacji poszczególnych zadań. Do jego kluczowych zadań należy regularna ocena aktualnego stanu projektu względem planu, a także identyfikacja wszelkich odchyień, które mogą wpłynąć na ostateczny sukces przedsięwzięcia. [Emanuela Giangregorio, *Practical Project Stakeholder Management* (2019).] W praktyce menadżer projektu wykorzystuje systemy takie jak JIRA czy Microsoft Project do monitorowania postępu prac zespołu. Na ich podstawie analizuje kluczowe wskaźniki efektywności, np. prędkość zespołu, stopień realizacji backlogu czy terminy dostaw funkcjonalności. Pozwala to na bieżąco diagnozować potencjalne ryzyka, w tym opóźnienia, przeciążenia zasobów lub zmiany zakresu projektu, i w razie potrzeby podejmować działania korygujące. [Marty Cagan, *Inspired: How to Create Tech Products Customers Love* (SVPG Press, 2018).] Ponadto menadżer prowadzi regularne spotkania statusowe, które mają na celu omówienie bieżących postępów, wymianę informacji między członkami zespołu oraz szybką identyfikację problemów. Spotkania te pomagają również w motywowaniu zespołu oraz utrzymaniu wysokiego poziomu zaangażowania. [Teresa Torres, *Continuous Discovery Habits* (Product Talk, 2021).] Kluczową rolą menadżera jest także odpowiedzialność za komunikację wyników monitorowania z

interesariuszami projektu. To on przygotowuje raporty i prezentacje, które przedstawiają aktualny stan realizacji projektu, ryzyka oraz rekomendacje dotyczące dalszych działań. Umiejętność jasnego i precyzyjnego przekazywania tych informacji jest niezbędna do budowania zaufania oraz utrzymania transparentności procesu zarządzania. [Harvard Business Review, “What Everyone Should Know About ‘Managing Up,’” accessed May 30, 2025, <https://hbr.org>.] Dzięki systematycznemu monitorowaniu menadżer projektu minimalizuje ryzyko niepowodzenia, pozwalając na elastyczne dostosowanie się do zmieniających się wymagań i warunków rynkowych, co jest szczególnie istotne w dynamicznym środowisku IT. [Camille Fournier, *The Manager’s Path* (O’Reilly Media, 2017).] Zarządzanie ryzykiem jest kolejnym fundamentalnym obowiązkiem menadżera projektu IT, który musi nie tylko identyfikować potencjalne zagrożenia, ale również opracowywać strategie minimalizujące ich wpływ na przebieg realizacji przedsięwzięcia. Proces ten rozpoczyna się od systematycznej analizy ryzyk, w której menadżer wykorzystuje różne techniki, aby wychwycić możliwe problemy techniczne, organizacyjne czy finansowe. [Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 7th ed. (Newtown Square, PA: PMI, 2021), 421–433.] Po zidentyfikowaniu ryzyk kluczowe jest określenie ich prawdopodobieństwa wystąpienia oraz możliwych skutków, co pozwala na priorytetyzację działań zapobiegawczych. Menadżer opracowuje plany awaryjne oraz strategie ich unikania, łagodzenia lub przeniesienia odpowiedzialności, dostosowując je do specyfiki projektu i jego otoczenia. [Eric Ries, *Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses* (New York: Crown Business, 2011), 159–165.] Komunikacja dotycząca ryzyk powinna być prowadzona transparentnie zarówno w zespole projektowym, jak i wobec interesariuszy, aby wszyscy zaangażowani mieli świadomość zagrożeń oraz podejmowanych działań zaradczych. [Eric Ries, *Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses* (New York: Crown Business, 2011), 159–165.] Komunikacja z interesariuszami stanowi jeden z fundamentalnych obszarów odpowiedzialności menadżera projektu IT. Efektywna wymiana informacji z klientami, sponsorami, użytkownikami końcowymi oraz członkami zespołu umożliwia precyzyjne zarządzanie oczekiwaniami i budowanie zaufania, co jest niezbędne dla powodzenia przedsięwzięcia. [Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 7th ed. (Newtown Square, PA: PMI, 2021).] Menadżer projektu odpowiada za regularne raportowanie postępów, które powinno być dostosowane do potrzeb różnych grup interesariuszy - od szczegółowych raportów technicznych dla zespołu programistycznego po syntetyczne podsumowania dla zarządu. [Marty Cagan, *Inspired: How to Create Tech Products Customers Love* (Silicon Valley Product Group, 2018).] W trakcie realizacji

projektu menedżer prowadzi negocjacje dotyczące zmian wymagań, alokacji zasobów czy terminów, co wymaga nie tylko umiejętności perswazji, ale także zdolności do wypracowywania kompromisów. Negocjacje w zarządzaniu projektami IT bywają często wyzwaniem, gdyż interesy różnych stron mogą być sprzeczne lub ulegać zmianom pod wpływem nowych informacji czy ograniczeń budżetowych. [Camille Fournier, *The Manager's Path: A Guide for Tech Leaders Navigating Growth and Change* (O'Reilly Media, 2017).] Autorzy książki „Dochodząc do tak. Negocjacje bez poddawania się” podkreślają, że skuteczne negocjacje opierają się na oddzieleniu osób od problemu oraz poszukiwaniu rozwiązań korzystnych dla wszystkich stron, co jest niezwykle istotne w dynamicznym środowisku IT. [Roger Fisher, William Ury, *Getting to Yes: Negotiating Agreement Without Giving In* (Penguin Books, 2011).] Komunikacja powinna przebiegać w sposób otwarty i transparentny, aby zapobiec powstawaniu nieporozumień i konfliktów. Efektywne zarządzanie komunikacją wpływa na utrzymanie zaangażowania interesariuszy oraz ich satysfakcję z realizacji projektu, a także pozwala na wczesne wykrywanie i rozwiązywanie potencjalnych problemów. Z tego względu rola menedżera projektu jako głównego punktu kontaktu i pośrednika komunikacyjnego jest kluczowa w środowisku IT, gdzie dynamika zmian i złożoność technologii wymagają ciągłej koordynacji oraz elastycznego dostosowywania się do nowych wymagań. [Harvard Business Review, “What Everyone Should Know About ‘Managing Up,’” accessed May 30, 2025, <https://hbr.org/2019/05/what-everyone-should-know-about-managing-up>.]

Rozdział 2

Praktyczne zastosowanie

Treść dla niniejszego rozdziału to zwykle praktyczne zastosowanie omawianego zagadnienia. Poprzez analizę rozdziału **Rola i zadania menadżera projektów IT**, a także jego wykorzystanie w praktyce, dyplomanta stara się nakreślić całość rozdziału praktycznego.

2.1. Tworzenie obiektów LaTeXa

Oprócz tego że możemy ładnie dzielić treści, umieszczać zdjęcia, tabele, kody źródłowe, odwoływać się do przypisów bibliograficznych, możemy także¹:

Notka. **Pamiętaj!** LaTeX jest bardzo skrupulatny, tak więc istnieje dla niego widoczna różnica pomiędzy **przypisem dolnym** (który zaobserwowałeś powyżej, `\footnote{}`), a **odwołaniem do bibliografii umieszczonym w przypisie dolnym** (`\footcite{}`).

Dla większości osób piszących na codzień teksty w Wordzie nie jest to żadna różnica, jednak poniekąd jako zecer musisz, również zadbać o odpowiedni i poprawny skład swojej pracy. Twój promotor może tego nie zauważyć (jeśli nie zna LaTeXa), jednak z pewnością doceni bardzo estetyczny wygląd pracy, a także Twoją skrupulatność przy pisaniu — jestem o tym przekonany w 100% — zaprocentuje Ci w przyszłości, gdyż każdy kolejny dłuższy dokument jaki bedziesz pisać w LaTeXu wykonasz znacznie, znacznie szybciej.

Pisać wytłuszczone paragrafy Ich treść może wskazywać na kluczowe aspekty na które chcesz zwrócić większą uwagę w danym rozdziale.

¹Tworzyć przypisy dolne w miejscu w którym rzeczywiście powinnu się znaleźć, a LaTeX przygotuje i sformatuje je za nas!

Mogą również rozciągać się na wiele linijek, więc nie musisz martwić się o to, że będziesz mieć mało miejsca. Wprost przeciwnie, będziesz musiał martwić się o to, aby praca nie przekroczyła określonego limitu (tak się właśnie stało w moim przypadku) ;)

Bądź określać terminy, definicje czy wzory matematyczne i nie muszą mieć one żadnego związku z matematyką tu chodzi bardziej o to, że warto z tych podstawowych elementów korzystać jak najczęściej.

W związku z pewną strukturą w pracy śmiało można także tworzyć²:

- twierzenia(`thm`),
- definicje(`defn`),
- założenia(`prop`),
- wnioski(`cor`),
- przypuszczenia(`conj`),
- przykłady(`exmp`),
- lematy(`lem`),
- spostrzeżenia(`rem`),
- lub notki(`note`)

Definicja 2.1.1 (Mechanika kwantowa). Teoria praw ruchu obiektów poszerzająca zakres mechaniki na sytuacje, dla których przewidywania mechaniki klasycznej nie sprawdzały się. Opisuje przede wszystkim świat mikroskopowy – obiekty o bardzo małych masach i rozmiarach, np. atom, cząstki elementarne itp., ale także takie zjawiska makroskopowe jak nadprzewodnictwo i nadciekłość. Jej granicą dla średnich rozmiarów, energii czy pędów zwykle jest mechanika klasyczna [4].

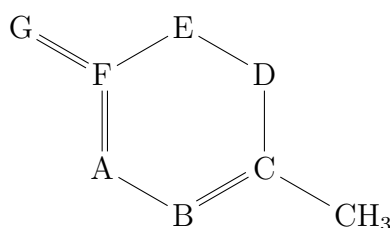
Jeden z najprostrzych przykładów na zobrazowanie prostoty działania trybu matematycznego, do wprowadzania dowolnych wzorów.

²Wiecej informacji znajdziesz pod tym adresem internetowym: http://www.latex-kurs.x25.pl/paper/Twierdzenia_definicje

$$4x = \frac{1+x^3}{2-y^4}$$

Zapewne słusznie zauważyłeś, że napisałem podstawowych, ponieważ liczba pakietów z których można korzystać jest tak wielka, że z pewnością odnajdziesz praktycznie dowolnie interesującą Cię interpretację wprowadzanych przez siebie wyników — przez wykresy (słupkowe, kołowe, 3D ect.), aż po wzory chemiczne, strukturalnie lub trójwymiarowe!

Poniżej drobny przykład zaledwie lekko zawysowujący temat wzorów chemicznych:



Rozdział niewidoczny w spisie treści

Można także jak już wcześniej pisałem (w kodzie źródłowym pracy) ukrywać niektóre rozdziały, podrozdziały, ect. wystarczy zakończyć daną komendę (dla przykładu podrozdziału) znakiem gwiazdki, aby całość wyglądała tak:

```
\section*{Tytuł podrozdziału}
```

Dla osób lubiących się w pisaniu programów, lub tych zmuszonych do publikacji fragmentów kodów źródłowych bądź skomplikowanych danych, można z powodzeniem wykorzystać najlepszy znany mi pakiet tj. `listings`. Efekt można zobaczyć poniżej wraz z podświetleniem i kolorowaniem śladni odpowiedniej dla danego języka, w tym wypadku dla języka C++ w przykładzie sortowania bąbelkowego [1].

Listing 2.1: Sortowanie bąbelkowe w C++

```
1 void BubbleSort(apvector <int> &num)
2 {
3     int i, j, flag = 1;    // set flag to 1 to start first pass
4     int temp;              // holding variable
5     int numLength = num.length( );
6     for(i = 1; (i <= numLength) && flag; i++)
7     {
8         flag = 0;
9         for (j=0; j < (numLength -1); j++)
```

```
10      {
11          if (num[j+1] > num[j])          // ascending order
              simply changes to <
12      {
13          temp = num[j];                  // swap elements
14          num[j] = num[j+1];
15          num[j+1] = temp;
16          flag = 1;                      // indicates that a
              swap occurred.
17      }
18  }
19  }
20  return;    //arrays are passed to functions by address;
              nothing is returned
21 }
```

Rozdział 3

Wyniki oraz podsumowanie

Ten rozdział jest zwykle przedstawieniem całej zebranej wiedzy w jedną spójną całość zwaną **podsumowaniem** lub jak kto woli zakończeniem.

Podsumowujemy tu wszystkie zebrane wyniki oraz wiedzę teoretyczną umieszczoną w poprzednich rozdziałach, a także wyciągamy wniosek, jeśli takowy można wysnuć.

Bibliografia

- [1] MathBits.com. *Bubble sort*. Luty 2015. URL: <http://mathbits.com/MathBits/CompSci/Arrays/Bubble.htm>.
- [2] L. Stal. *Słowa i czyny*. Grudzień 2016.
- [3] Ł. Świerk. *Świerszcze w żółci*. Marzec 2017.
- [4] Wikipedia. *Mechanika Kwantowa*. Luty 2015. URL: <http://poradnia.pwn.pl/lista.php?id=1528>.

Spis tabel

1.1	Prosta tabelka dla przykładu	5
1.2	Fazy zarządzania projektem i ich działania	7
1.3	Rodzaje ryzyk w projektach IT	9

Spis rysunków