

**Kazimierz Frączkowski**

# **Zarządzanie projektem informatycznym**

**Projekty w środowisku wirtualnym  
Czynniki sukcesu i niepowodzeń projektów**



**Oficyna Wydawnicza Politechniki Wrocławskiej  
Wrocław 2003**

Wydział Informatyki i Zarządzania  
Wydziałowy Zakład Informatyki

Opiniodawca  
Zdzisław SZYJEWSKI

Opracowanie redakcyjne i korekta  
Alina KACZAK

Projekt okładki  
Justyna GODLEWSKA

© Copyright by Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2003

OFICyna WYDAWNICZA POLITECHNIKI WROCLAWSKIEJ  
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

ISBN 83-7085-731-0

Drukarnia Oficyny Wydawniczej Politechniki Wrocławskiej. Zam. nr 633/2003.

## Spis treści

Od Autora .....	5
Wprowadzenie .....	6
1. Geneza zarządzania projektami .....	8
1.1. Przegląd historyczny wybranych zagadnień zarządzania .....	8
1.2. Podstawowe pojęcia i definicje stosowane w zarządzaniu projektami .....	11
1.3. Czynniki charakterystyczne projektu .....	13
1.4. Proces .....	14
2. Planowanie projektu informatycznego .....	17
2.1. Cele planowania projektu .....	18
2.2. Definiowanie celów projektu .....	19
2.3. Zasoby projektu .....	20
2.4. Definiowanie ograniczeń w projekcie .....	21
2.5. Strategia realizacji projektu .....	24
2.6. Ocena ryzyka .....	25
2.7. Struktura projektu – dekompozycja projektu na zadania – WBS .....	26
2.8. Szacowania w projekcie .....	31
2.9. Metoda punktów funkcyjnych .....	34
2.10. Harmonogram .....	45
2.11. Sieciowe diagramy zależności .....	48
2.12. Inicjowanie projektu .....	50
3. Śledzenie i zarządzanie zmianami projektu .....	52
3.1. Śledzenie projektu – monitorowanie .....	52
3.2. Zarządzanie jakością w projekcie .....	53
3.3. Źródła i rodzaje zmian .....	59
3.4. Proces zarządzania zmianami .....	60
3.5. Śledzenie i nadzorowanie czasu oraz kosztów projektu .....	65
3.6. Nadzorowanie projektu metodą wartości uzyskanej ( <i>Earned Value</i> ) .....	66
3.7. Podsumowanie .....	74
4. Modele pracy i komunikacji .....	77
4.1. Telepraca .....	78
4.2. Modele organizacji pracy zespołów projektowych .....	80
4.3. Elektroniczny członek zespołu – zespół .....	85
4.4. Wirtualna sieć partnerów przedsięwzięcia .....	89
4.5. Zespoły Projektowe .....	91
4.6. Praca grupowa .....	92
5. Zarządzanie ryzykiem .....	94
5.1. Czynniki mające wpływ na ryzyko .....	99

---

5.2. Identyfikacja ryzyka – metody analizy ryzyka – kwestionariusze, listy kontrolne .....	102
5.3. Akcje naprawcze .....	104
5.4. Metoda punktowa szacowania ryzyka .....	105
5.5. Metoda PERT szacowania ryzyka .....	113
6. Projekty wdrożeniowe – <i>outsourcing</i> .....	118
6.1. Rodzaje projektów informatycznych oraz organizacja pracy i zespołów .....	118
6.2. Wdrożenie przez <i>outsourcing</i> .....	121
7. Czynniki sukcesów i niepowodzeń projektów .....	125
7.1. Niektóre badania statystyczne niepowodzeń projektu .....	125
7.2. Wpływ wielkości projektu na jego sukces .....	128
8. Rola i zadania kierownika projektu .....	130
8.1. Usytuowanie kierownika projektu a jego skuteczność .....	131
8.2. Dobór członków zespołu .....	132
8.3. Rekrutacja członków zespołu .....	133
8.4. Budowa kompetencji w projekcie .....	136
8.5. Konflikt .....	136
8.6. Motywowanie .....	136
8.7. Delegowanie uprawnień .....	139
9. Dodatek .....	141
9.1. Metoda zarządzania projektami PRINCE-2 .....	141
9.2. Oprogramowanie wspomagające zarządzanie zmianami .....	144
9.3. Najpopularniejsze narzędzia <i>open source</i> .....	145
9.4. Oprogramowanie komercyjne do zarządzania zmianami .....	150
9.5. Narzędzie <i>open source</i> do zarządzania projektami .....	153
Słownik pojęć i terminów .....	155
Literatura .....	161

## Od autora

Niniejsza książka stanowi próbę opracowania zagadnień związanych z zarządzaniem przedsięwzięciami informatycznymi. Obejmuje wybrane współczesne poglądy na tematy, które są kluczowe w obszarze planowania, zarządzania zmianami, organizacji zespołów projektowych, szacowania kosztów, monitorowania ryzyka i decydują o sukcesie lub niepowodzeniu projektu. Celem opracowania jest również próba podzielenia się z czytelnikami własnym doświadczeniem zawodowym, nabytym podczas prac nad projektami informatycznymi, uczestniczenia w szkoleniach z tej tematyki oraz prowadzenia wykładów i seminariów na Wydziale Informatyki i Zarządzania Politechniki Wrocławskiej, dla studentów kierunku Inżynieria Oprogramowania w latach 1998–2003. Prace nad książką zainicjowało dostarczenie mi notatek w postaci elektronicznej z cyklu moich wykładów przez studenta – pana Piotra Hojnara. Dziękuję również innym studentom, którzy przygotowując i prezentując w trakcie seminariów niektóre zagadnienia z tej dziedziny, narzekali na brak polskiej literatury z tego przedmiotu, przez co stymulowali pracę nad książką. Studenci, którzy uczestniczyli w dojrzałych dyskusjach na seminariach i przygotowali interesujące wystąpienia, stali się poniekąd współautorami niniejszego opracowania.

Książka jest przeznaczona dla studentów informatyki i zarządzania oraz kierowników projektów. Zawartość opracowania to próba osiągnięcia kompromisu między rozległym zakresem tematycznym tego zagadnienia a programem studiów, który opracowaliśmy wspólnie z panią dr inż. Iwoną Dubielewicz, której dziękuję za cenne propozycje dotyczące omawianych zagadnień i sposobu ich przedstawienia. Zachęty i osobistego wsparcia w pracy udzielił mi prof. Zbigniew Huzar, któremu dedykuję tę pracę.

Będę wdzięczny wszystkim, którzy zechcą przesłać uwagi i sugestie dotyczące podręcznika.

Adres: e-mail: [kazimierzfraczkowski@pwr.wroc.pl](mailto:kazimierzfraczkowski@pwr.wroc.pl)

*Kazimierz Frączkowski*

## Wprowadzenie

Niekiedy, na pewnym etapie kariery zawodowej, nieoczekiwanie stajemy przed szansą zostania kierownikiem projektu (ang. *Project Manager* – PM). Rzadko z wyprzedzeniem planujemy ubieganie się o funkcję kierownika projektu – PM. Najczęściej nie jesteśmy do tego teoretycznie przygotowani, przez co zarządzanie projektem nabiera charakteru „twórczej intuicyjnej improwizacji”. Każdy kolejny kierownik projektu dąży do jego realizacji, musi samodzielnie, od podstaw, wykreować wszelkie rozwiązania, improwizować, weryfikować swoje działania metodą prób i błędów. Tymczasem sztuka zarządzania projektami, która znacznie rozwinęła się w ciągu ostatnich piętnastu lat, stanowi dziś ścisłą, profesjonalną dyscyplinę. Składa się na nią wiedza teoretyczna, konkretny zestaw umiejętności i kompetencji, a także proces certyfikacji przeprowadzany przez np. *Project Management Institute* (PMI®) z siedzibą w Waszyngtonie – ośrodek badawczy zgłębiający arkana tej dziedziny. Profesjonalizm w zarządzaniu projektami ma nie tylko pozytywne odzwierciedlenie w stylu działania organizacji, ale także wpływa na podniesienie motywacji i satysfakcji z pracy osób odpowiedzialnych za projekty. Warto pamiętać, że wiedza w tej dziedzinie rozwija się bardzo dynamicznie i tylko stałe podnoszenie kompetencji oparte na doświadczeniu przodujących w zarządzaniu projektami instytutów naukowych i stowarzyszeń branżowych gwarantuje przewagę konkurencyjną ich adeptom.

Wiedza na temat zarządzania projektami nie powinna być zarezerwowana jedynie dla najwyższego kierownictwa, odpowiedzialnego za całość organizacji. Tej grupie jest ona bezsprzecznie potrzebna do realizacji celów strategicznych, podejmowania trudnych decyzji i alokowania zasobów w sposób sprzyjający realizacji projektów. Jednak bez odpowiedniego przygotowania zespołów projektowych nie ma gwarancji, że zadania przydzielane im w poszczególnych etapach projektów zostaną prawidłowo zrealizowane. Wszyscy członkowie organizacji potrzebują wiedzy i umiejętności w zakresie zarządzania projektami – jedni bardziej pogłębionej i rozbudowanej, inni mniej szczegółowej. Mimo że ci z nas, którzy choć raz zarządzali projektem lubią nazywać siebie *Project Managerami*, coraz częściej liczą się formalne, poświadczone certyfikatem uprawnienia do tego tytułu. Dyplomy wydawane przez MT&DC, przy współpracy z *Educational Services Institute International* (ESI®), potwierdzają uzyskanie przez uczestnika kursu gruntownego wykształcenia i nabycia kompetencji

---

w tym zakresie. Otrzymanie certyfikatu ESI<sup>®</sup> łączy się z fundamentalnym poznaniem niezbędnej tematyki związanej z zarządzaniem projektem i otwiera drogę do uzyskania dyplomu *Project Management Professional* (PMP<sup>®</sup>). Nie bez znaczenia i nie przypadkiem przedmiot ten znalazł się w programie studiów na kierunku informatyka. Mam skromną nadzieję, że praca ta pomoże w pewnym stopniu Czytelnikowi w szerszym spojrzeniu na realizację przedsięwzięć informatycznych, uwzględniając w zarządzaniu projektami omawiane zagadnienia.

# 1. Geneza zarządzania projektami

## 1.1. Przegląd historyczny wybranych zagadnień zarządzania

Potrzeba zarządzania jest pochodną złożoności przedsięwzięć, których realizacja rozciąga się w czasie, angażuje zasoby i ma określony cel. Można postawić tezę, że zarządzanie może obejmować działania pojedynczego człowieka do dużych złożonych przedsiębiorstw, korporacji, organizmów państwowych i międzynarodowych. Z tymi ostatnimi mamy do czynienia dopiero od setek lat, ale zarządzanie uprawiano już od tysiącleci.

Powstające w starożytności cywilizacje swój rozwój i osiągnięcia zawdzięczają jednostkom, które posługiwały się adekwatnymi do istniejącego otoczenia efektywnymi metodami zarządzania. W Egipcie nie powstałyby piramidy, gdyby przy ich budowie nie zastosowano takich elementów zarządzania, jak planowanie, organizowanie i kontrolowanie zaplanowanych prac. Aleksander Wielki, zwany Macedońskim (356–323 p.n.e.), król Macedonii i wychowanek Arystotelesa, posługiwał się sztabową organizacją w koordynacji działań w toku swoich kampanii wojennych, co zapewniło mu w historii miano znakomitego stratega i taktyka oraz administratora. Cesarstwo rzymskie rozwinęło dobrze wykształconą strukturę organizacyjną, której podstawą był system komunikacji (*wszystkie drogi prowadzą do Rzymu*) i kontroli. Na temat praktyk i koncepcji zarządzania wypowiedział się Sokrates w 400 roku p.n.e.; Platon w 350 roku p.n.e. opisał specjalizację pracy; Farabi, Alfarabi, właściwie Muhammad ibn Takhan abu Nasr al.-Farabi, podał kilka cech przywództwa w 900 roku n.e. Kształtowanie się dokonań w dziedzinie zarządzania przedstawiono na rysunku 1.1.

Współcześni prekursorzy zarządzania rozwinęli swą działalność dopiero w XIX wieku. Robert Owen (1771–1858), angielski ekonomista, filozof, polityk, przemysłowiec i reformator, był jednym z pierwszych menedżerów, który dostrzegł znaczenie zasobów ludzkich organizacji. Do jego czasów na robotników fabrycznych patrzono w sposób bardzo podobny jak na maszyny czy sprzęt. Owen, który sam był właścicielem fabryk, był przekonany, że robotnicy mają prawo do



poszanowania i godności, dlatego w kierowanej przez siebie przędzalni w New Lamark wdrożył unikatowy wówczas program socjalny (budowa mieszkań dla robotników, podwyżka płac, skrócenie czasu pracy). Wychodził z założenia, że większa troska o robotnika zaowocuje zwiększoną wydajnością pracy. Jego koncepcjami był zachwycony car Mikołaj I, od którego otrzymał propozycję osiedlenia się w Rosji wraz z 2 mln angielskich robotników. Mimo że nie znalazł naśladowców wśród siebie współczesnych, jego idee zostały później rozwinięte w behawioralnym podejściu do zarządzania.

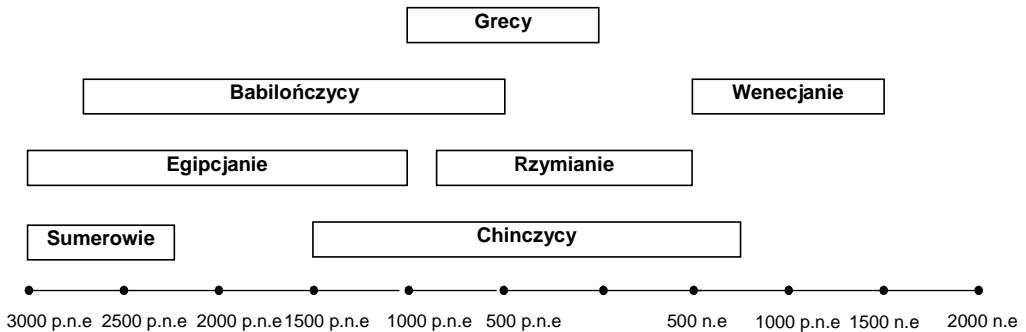
Charles Babbage (1792–1871), angielski matematyk, pionier informatyki, profesor uniwersytetu Cambridge (1828–1839), koncentrował uwagę na efektywności produkcji. Babbage wiązał wielkie nadzieje z podziałem pracy i był orędownikiem zastosowania matematyki do takich problemów, jak efektywne wykorzystanie maszyn, pomieszczeń i materiałów. W tym sensie jego praca wyprzedziła zarówno klasyczne, jak i ilościowe spojrzenie na zarządzanie. Babbage dostrzegał jednak również człowieka. Rozumiał korzyści płynące z współdziałania pracodawcy i robotników, i rozumiał korzyści płynące ze współdziałania w zyskach tych ostatnich. Współcześni prekursorzy naukowego zarządzania to między innymi Federic W. Taylor (1865–1915), Frank Gilbreth (1868–1924), Henry Gantt (1861–1919) czy Harrington Emerson (1853–1931). Pionierem w dziedzinie wydajności pracy był niewątpliwie Taylor, który wprowadził liczne innowacje w sposobie projektowania stanowiska pracy, szkolenia pracowników, którzy mieli pracę wykonywać i użytkować wyższą jakością produkowanych wyrobów.

## Prekursorzy zarządzania

Współcześnie zarządzanie kojarzy nam się przede wszystkim z dużymi przedsiębiorstwami, korporacjami, organizacjami, które są kierowane przez zarządy, prezesów, rady nadzorcze. Zanim do tego doszliśmy, na przestrzeni wielu stuleci żyły narody i ich wybitni przedstawiciele, którzy stworzyli podwaliny naszej cywilizacji.

- Sumerowie – znani są z metod zarządzania opartych na spisanych przepisach,
- Egipcjanie – stworzyli reguły i praktyki w budowie piramid,
- Babilończycy – zaczęli stosować szeroki zestaw aktów prawnych i środków politycznych w zarządzaniu,
- Grecy – specjalizowali się w różnych systemach zarządzania miastami i państwami,
- Rzymianie – używali struktury organizacyjnej w celu komunikacji i kontroli,
- Chińczycy – wprowadzili rozległą strukturę organizacyjną dla agencji rządowych oraz w dziedzinie sztuki,
- Wenecjanie – zastosowali projektowanie organizacji i planowanie do osiągnięć

cia celu, jakim było panowanie na morzu.



Rys. 1.1. Prekursorzy zarządzania i przełomowe ich dokonania, według Griffina Ricky [22]

Chciałbym podać tylko skromny wybór spośród znamienitych postaci, które wywarły olbrzymi wpływ na postęp w zarządzaniu. Ze względu na charakter tej książki przytaczam tylko bardzo skróconą ich charakterystykę:

Sokrates	400 p.n.e.	– autor koncepcji i praktyki zarządzania,
Platon	350 p.n.e.	– podkreślał rolę specjalizacji w pracy,
Alfarbi	900 p.n.e.	– wyróżniał znaczenie posiadania cechy przywództwa w zarządzaniu,
Robert Owen	1771–1858	– zmienił sposób traktowania i nadał duże znaczenie warunkom ludzkiej pracy,
Charles Babage	1752–1871	– koncentrował się na efektywności produkcji,
Frank Gilberth	1868–1924	– projektant wydajnych stanowisk pracy,
Lilian Gilbert	1879–1972	– prace nad optymalizacją np. optymalizacja sposobu układania cegły, psychologia przemysłu, „z dwunastką taniej” (była matką 12 dzieci),
Henry Gantt	1861–1912	– autor powszechnie stosowanego wykresu Gantta,
Harrington Emerson	1853–1931	– doradca organizacyjny rządu USA,
Henry Ford	1863–1947	– wynalazca taśmy produkcyjnej,
Eliyahu Goldratt		– łańcuch krytyczny, tak silna organizacja, jak silne jej najsłabsze ogniwo,
Frederick W. Taylor	1861–1919	– pionier w dziedzinie wydajności pracy.

## 1.2. Podstawowe pojęcia i definicje stosowane w zarządzaniu projektami

### Zarządzanie

Definicji zarządzania jest wiele, tak jak książek na ten temat. Większość z nich dąży do zwięzłego i precyzyjnego zdefiniowania istoty zagadnienia, jednak to czym jest zarządzanie zależy od szerokości, obszaru i organizacji, gdzie następuje proces zarządzania. Jedną z definicji mówi, że zarządzanie to *dokładne poznanie tego, czego się oczekuje od ludzi, a następnie dopilnowanie, aby wykonali to w najlepszy i najtańszy sposób* [F.W. Taylor. Shop Manager, Harper & Row, New York 1903, s. 21].

Gdy przedmiotem zarządzania będą projekty informatyczne oraz zakres kompetencji i czynności należący do kierownika projektu, wówczas *zarządzanie* możemy zdefiniować jako *ogół działań zmierzających do efektywnego wykorzystania zespołów ludzkich, środków materialnych i czasu w celu osiągnięcia wcześniej sformułowanego celu projektu informatycznego w określonej technologii zakresie i jakości*.

W procesie zarządzania można wyróżnić pięć podstawowych funkcji: planowanie, organizowanie, przekazywanie poleceń, koordynację i kontrolowanie.

W ramach każdej z tych funkcji zarządzający może wykorzystywać określone środki organizacyjne, motywacyjne i techniczne, służące do ich realizacji. Zarządzanie to także nauka o metodach, zasadach i instrumentach dotyczących realizacji podanych założeń.

### Projekt

Projekt (ang. *Project*) jest nowym przedsięwzięciem, nie mającym wzorca, nie realizowanym wcześniej. Dotyczy nowej sytuacji, wymaga nierutynowego podejścia. Nie możemy polegać na historycznych sposobach postępowania z danym problemem. Projekt jest to przedsięwzięcie, na które składa się zespół czynności, które są charakterystyczne przez to, że mają datę rozpoczęcia, specyficzne cele i limity, ustalone odpowiedzialności (obowiązki) realizatorów, budżet, rozkład czynności oraz datę ich ukończenia (gdy celem projektu jest rozwinięcie systemu oprogramowania, wtedy jest to projekt rozwoju oprogramowania lub projekt inżynierii oprogramowania). Podane cechy decydują o tym, że jest to nowe przedsięwzięcie nie mające wzorca, nie będące rutynowymi działaniami, nie realizowane wcześniej. Projekt, projektowanie – twórcza działalność związana

z powstawaniem produktu, powodująca jego zróżnicowanie wynikające z cech, parametrów użytkowych, zgodności ze standardami, trwałości, niezawodności, łatwości naprawy i dającego się odróżnić od innych projektów stylu. Projekt ma wizję, najczęściej zawiera rysunki, schematy, obliczenia, opisy, kosztorysy itp. dotyczące wykonania da-

nego urządzenia, przedmiotu, systemu informatycznego lub obiektu budowlanego.

### **Przykład**

Projekt systemu Rejestr Zakładów Opieki Zdrowotnej (RZOZ), którego celem jest wsparcie mechanizmów planowania i zaspokajania potrzeb na świadczenia zdrowotne przez ewidencję i bieżącą aktualizację informacji rejestrowych o wszystkich zakładach medycznych w Polsce, z udziałem wojewódzkich organów rejestrowych oraz udostępnianie tych informacji przez serwis informacyjny [www.rejestrzoz.gov.pl](http://www.rejestrzoz.gov.pl)

Projekty mogą dotyczyć różnych przedsięwzięć informatycznych, dlatego zostały opracowane różne techniki realizacji projektów, w tym uwzględniające obszar i zakres, takie jak:

**Projekt „od dołu do góry”** (ang. *Bottom-up design*) – proces projektowania systemu przez identyfikację składowych niższego poziomu, projektowanie struktury w celu zintegrowania składowych niższego poziomu w coraz większe podsystemy, aż projekt będzie ukończony.

**Projekt „od góry do dołu”** (ang. *Top down design*) – proces projektowania systemu poprzez identyfikację jego większych składników, rozłożenie ich na składniki niższego poziomu oraz rozdzielanie dopóki pożądaną poziom szczegółowości nie zostanie osiągnięty, jest to działanie przeciwne do projektu „od dołu do góry”.

**Projekt inżynierii oprogramowania** (ang. *Software engineering project*) – zestaw wszystkich czynności, funkcji i zadań zarówno technicznych, jak i menedżerskich, wymaganych do zaspokojenia terminów i warunków realizacji projektu. Projekt inżynierii oprogramowania może być częścią większego projektu. Projekt inżynierii oprogramowania jest czynnością charakteryzującą się datą startu, specyficznymi celami i limitami, ustanowionymi obowiązkami, budżetem i planem oraz datą ukończenia. Projekt inżynierii oprogramowania zużywa zasoby, które spełniają wymagania projektu, wyszczególnione w uzgodnieniach projektowych. W niektórych przypadkach projekt może obejmować tylko porcję produktu oprogramowania, może trwać wiele lat i składać się z licznych podprojektów.

**Projekt inżynierii systemu** (ang. *System engineering project*) – zestaw wszystkich czynności, funkcji zarówno technicznych, jak i zarządczych, wymaganych do zaspokojenia terminów i warunków realizacji projektu. Projekt inżynierii systemu jest czynnością charakteryzującą się datą startu, specyficznymi celami i limitami, ustanowionymi obowiązkami, budżetem i planem oraz datą ukończenia. Zużywa zasoby i ma na celu wytworzenie produktu lub zestawu produktów, które zaspokajają wymagania

projektu wyszczególnione w specyfikacji projektu. W niektórych przypadkach może obejmować tylko porcję produktu oprogramowania, trwać wiele lat i składać się z licznych podprojektów.

**Projekt oprogramowania** (ang. *Software desing*) – proces definiowania architektury oprogramowania składników, modułów, interfejsów, podejścia testowego oraz danych dla systemu oprogramowania.

**Projekt rozwoju oprogramowania** (ang. *Software development process*) – patrz projekt inżynierii oprogramowania.

**Projekt systemu** (ang. *System design*) – 1. Proces (patrz p. 1.4) definiowania architektury, jej składników, modułów funkcjonalnych, interfejsu, danych, sprzętu/oprogramowania dla systemu w celu zaspokojenia wyszczególnionego wymagania systemu. 2. Wynik przebiegu procesów projektowania systemu. Bliskoznaczny projektowi architektury. Patrz też projekt oprogramowania.

**Projekt wstępny** (ang. *Preliminary desing*) – 1. Proces analizowania alternatyw projektu oraz definiowania architektury sprzętu/systemu oprogramowania. W inżynierii oprogramowania wstępny projekt zwykle zawiera definicję oraz strukturę komputerowych składników programów i danych, definicje interfejsów oraz przygotowanie rozmieszczenia w czasie i oszacowania kosztów. 2. Wynik przebiegu procesów projektowania wstępnego. Niekiedy rozumiany jako opis projektu wstępnego.

**Projektowanie-do-kosztu** (ang. *Desing-to-cost*) – podejście w zarządzania projektem, polegające na utrzymaniu projektu w granicach kosztu przewidzianego w harmonogramie. To znaczy, że przebieg projektowania jest oceniany (oszacowywany) poprzez monitorowanie jednostkowych wymagań w kolejności zależnej od ważności oraz ustanowienie rygorystycznych celów kosztowych do projektowania i wykonania każdego zadania. Aby to osiągnąć, rezerwuje się zapas na przypadki odstępstwa kosztów (zwykle 15–20%), szukając praktycznego kompromisu między operacyjnymi możliwościami wykonawczymi, zakresem i harmonogramem.

**Projektowanie szczegółowe** (ang. *Detalied design*) – 1. W inżynierii oprogramowania proces weryfikacji polegający na usuwaniu błędów, rozszerzaniu projektu wstępnego oprogramowania w celu zawarcia bardziej szczegółowych opisów logiki przetwarzania, struktur danych oraz definicji danych do tego stopnia, gdzie projekt jest wystarczająco szczegółowy, aby mógł zostać wdrożony. 2. Wynik szczegółowego procesu projektowania. Niekiedy bliskoznaczny z opisem specyfikacji projektu szczegółowego.

### 1.3. Czynniki charakterystyczne projektu

Jak już wcześniej wspomniano, projekt informatyczny charakteryzuje się innowacyjnością, zakresem, ryzykiem, zapotrzebowaniem na zasoby ludzkie i materialne oraz niezbędnym czasem na realizację. Wykonawca projektu-realizator (firma, zespół), przystępując do jego wykonania, najczęściej zmienia swój dotychczasowy model pracy (choć dobry do realizacji codziennych zadań), ze względu na **cel projektu** [14, 24, 46, 50]. Jednym z głównych błędów jest niedocenienie złożoności i wpływu na projekt kontekstu organizacyjnego firm zaangażowanych w jego realizację i niezrozumienie celu. Główne czynniki, które charakteryzują projekt:

- działania nastawione na dokonanie zmian,
- ocena możliwych zysków i strat,
- zakres,
- strategia,
- ewolucja modelu prac w wyniku doświadczenia,
- wykorzystanie „bazy wiedzy” do tworzenia nowych jakości,
- cel (biznesowy, organizacyjny, jakościowy, inny), misja (przesłanie),
- oryginalność,
- działanie niepowtarzalne,
- dotyczy elementów rozwoju, ma cechy ewolucji,
- działanie nastawione na nowatorską obsługę procesów biznesowych związanych z produktem dla określonego podmiotu, dla którego jest realizowany projekt,
- metoda „racjonalnego działania” jako czynnik sprawczy inicjacji projektów,
- inne czynniki w zależności od charakteru projektu.

Są to współczesne wyznaczniki związane z projektem, ale czy jest to uniwersalna recepta na generowanie projektów? W XIX wielu C. Bernard, francuski patolog zdefiniował *czynniki, które sprzyjają powstawaniu rzeczy nowych – projektów*. Według autora należą do nich:

- wyraźne ustalenie celu działania,
- ustalenie szczegółowo wszystkich kierunków działań i środków, za pomocą których można osiągnąć założony cel,
- ułożenie dokładnego planu działania, zmierzającego do osiągnięcia celu przy zastosowaniu najlepszych w obecnych warunkach środków,
- wykonanie skrupulatnie założonego planu,
- skontrolowanie osiągniętych wyników i porównanie z zamierzonym celem – wyciągnięcie wniosków na przyszłość (do następnego planu projektu).

Można zatem wnioskować, że wymienione elementy *racjonalnego działania* były podstawą współczesnego pojmowania realizacji projektów.

## 1.4. Proces

Proces jest to ściśle zdefiniowany ciąg działań nastawionych na ustabilizowanie i optymalizację stanowiących podstawę technologii wytwarzania wielu identycznych lub podobnych produktów o określonych parametrach użytkowych lub świadczonych usługach. Przez technologię należy rozumieć przetwarzanie w sposób celowy i ekonomiczny z zastosowaniem nowej techniki [31, 48].

### Projekt a proces – czynniki różnicujące

Właściwością projektów jest ukierunkowanie na zmiany [7, 21]. Wprowadzenie nowości, zmiana stanu obecnego jest podstawową cechą każdego projektu [39]. Projektem jest na przykład zbudowanie nowego połączenia kolejowego między miejscowością A i B, wprowadzenie nowej usługi dostarczania poczty (listy, paczki) lub zmiana organizacji pracy (część pracowników dostaje komputery do domu i przez Internet przekazuje rezultaty pracy). Projekty są najczęściej realizacją wytworów ludzkiej wyobraźni, która praktycznie adaptuje zdobycze nauki i przez przemyślane działanie tworzy nową jakość. Powtarzalne wykonywanie czynności, które są realizowane według określonej technologii, a ich rezultatem jest jasno zdefiniowany produkt, to typowe działania procesowe, jak np. produkcja kolejnego z serii motocykla, roweru, krzesła, płyty CD; wykonanie usługi, np. zdjęcie RTG, sprzedaż TV przez kasjera, przelew bankowy i tym podobne czynności, które mają charakter powtarzalny i są realizowane według opracowanych zasad (na podstawie wcześniej zrealizowanych projektów). Sytuacje rutynowe, powtarzające się z dużą częstotliwością (do czasu wprowadzenia zmiany przez projekt) w jednostce czasu oraz przez dowolnie długi okres, to procesy [20, 48].

Na rynku mamy do czynienia z podmiotami gospodarczymi, których cechą szczególną jest realizacja *procesów* lub *projektów*. Prowadzenie działalności procesowej powinno być zdefiniowane i zweryfikowane w wielu konkretnych typowych realizacjach. Zespoły wykonawców dysponują opisanym ciągiem działań stanowiącym elementy w realizacji całego procesu. Zidentyfikowane są również problemy zarządzania zespołem realizującym proces i algorytmy współdziałania, komunikacji, kompetencji oraz funkcji z podziałem zakresu prac członków zespołu. Projekt kreuje specyficzne, niepowtarzalne podejście, adekwatne do osiągnięcia celu projektu, przez opracowanie procedur zarządzania i realizacji, które tworzą proces realizacji.

Czas, zarówno opracowywania procesów, jak i projektów, jest czynnikiem wymuszającym ich modyfikacje. Obserwujemy zamiany w technologii, powstają nowe urządzenia, materiały, rodzaje energii, doświadczenia i obserwacje z realizacji stosowanych *procesów* itp., co najczęściej skutkuje potrzebą ich wprowadzenia do funkcjonujących *procesów* podyktowaną względami ekonomicznymi – sprostaniu konkuren-

cji. Zmiany w procesach najczęściej są wprowadzane sukcesywnie w długim okresie, mogą np. dotyczyć wymiany narzędzia lub urządzenia na taśmie produkcyjnej, które stanowiło wąskie gardło procesu lub zmiany kolejności czynności operacji cząstkowych. W przypadku systemu informatycznego, w którym występuje *proces* generowania cyklicznych raportów z bazy danych zastosowanie szybszego procesora lub pamięci masowej o krótszym czasie dostępu, co może spowodować skrócenie czasu niezbędnego na dostarczenie użytkownikowi wymaganego raportu. W projektach zmiany mają szczególny wymiar i skalę, najczęściej są głębokie, gruntowne, niekiedy rewolucyjne. Skala projektów, ich charakter powoduje, że ich wprowadzenie jest związane z poruszaniem się w obszarze czynników niepewnych oraz są zagrożone różną skalą ryzyka.

Rola, wymagania, niezbędne predyspozycje i kwalifikacje kierownika są inne w przypadku realizacji procesów, a inne w przypadku prowadzenia projektów [17, 18, 45]. Kierownictwo firmy, w której są realizowane procesy, np. fabryka samochodów, sprzętu AGD, banku, ingeruje w procesy, gdy np. obniża się jakość produkcji, tj. zwiększa się koszt reklamacji, a w przypadku banku – przy kasie pojawia się nietypowa sytuacja, np. klient zagubił dokumenty, a chce podjąć gotówkę; ktoś chce zrealizować fałszywy czek itp. W projekcie prawie wszystkie działania są nietypowe i nigdy nie wiadomo, kiedy będzie potrzeba konsultacji czy bezpośrednich działań kierownictwa firmy, więc założyć należy, że udział kierownictwa jest wskazany przez cały okres prowadzenia projektu.

W gospodarce występują takie podmioty, które są nastawione na sprawną realizację procesów, są to małe i duże firmy wytwarzające dobra użytku codziennego w skali masowej (artykuły tzw. przemysłowe oraz do zaspokojenia codziennych potrzeb, np. spożywcze, higieniczne itp.), biura administracji publicznej, banki, sektor rozrywki itp. Istnieją także firmy realizujące jednostkowe przedsięwzięcia w dłuższym czasie, np. biura projektów (mostów, dróg, okrętów itp.), jednostki odpowiedzialne za opracowanie i wprowadzenie nowej usługi bankowej, np. e-konto, podpis elektroniczny. W działalności firm, których podstawową działalność stanowią projekty mogą występować procesy, np. w księgowości czy obsłudze płatności. W działalności firm informatycznych, zwłaszcza dużych, część działalności, np. produkcja komputerów, podzespołów itp., to działalność procesowa, ale faza opracowania nowego produktu, np. nowego procesora, innego rodzaju nośnika informacji, to działanie projektowe. Na naszym rynku informatycznym są firmy informatyczne, które specjalizują się we wdrożeniach systemów (niekoniecznie własnej produkcji) i tu występują działania zarówno projektowe, jak i procesowe (np. zdefiniowana technologia wdrożenia systemu SAP). Firma informatyczna, która sprzedaje tylko komputery czy akcesoria niewiele się różni od firmy, która sprzedaje sprzęt AGD, ale firma, która przyjmuje zlecenia wygrywa przetargi na opracowanie i dostarczenie systemu do obsługi telekomunikacji, np. billingu, czy rozliczenia wszystkich podmiotów gospodarczych z obowiązku podatkowego z urzędem skarbowym, to na pewno firma realizująca pro-



jekty.

## 2. Planowanie projektu informatycznego

*Jeśli nie potrafisz czegoś zaplanować,  
to na jakiej podstawie sądzisz, że potrafisz to zrobić.*

KF

Planowanie rozumiane jest najczęściej jako zespół działań pomocnych w wytyczaniu celów organizacji i określaniu sposobu ich najlepszej realizacji. W procesie planowania występują takie elementy, jak podejmowanie decyzji, wybór kierunków działań oraz sprawność zarządzania. Planowanie jest również immanentną częścią projektu informatycznego, którego zadaniem jest osiągnięcie celu projektu z uwzględnieniem ograniczeń projektu.

Trudno ustalić listę priorytetów uniwersalną dla wszystkich projektów, która uzasadniałaby czy wskazywałaby na cele i zadania związane z szacowaniem planowania projektu [2, 11, 42, 43]. Do najważniejszych należą:

- określenie założeń projektowych (cel, zakres, ograniczenia),
- oszacowanie kosztów przedsięwzięcia i jego użyteczności,
- pomoc w identyfikacji obszarów ryzyka,
- utworzenie harmonogramu, którego cechy to:
  - możliwość koordynacji i integracji prac tworzących przedsięwzięcie,
  - podstawowe narzędzie do kontroli realizacji projektu,
  - wspieranie motywacji zespołów przez określenie celów,
  - miara postępu prac,
  - tworzenie wiedzy dla przyszłych projektów.

*Nie tylko nieudane projekty są źródłem postępu.*

KF

### **Planowanie jest procesem realizowanym w całym cyklu życia projektu**

Pytania, jakie najczęściej pojawiają się przed rozpoczęciem planowania to:

1. Jak daleko planować?
2. Jak szczegółowo (głęboko) planować?

3. Jak dużo czasu poświęcić na planowanie?
4. Skąd czerpać dane?

*Mówi się, że planowanie zabiera co najmniej 10% czasu, ale czasem dochodzi nawet do 90%.*

## 2.1. Cele planowania projektu

Planowanie projektu w znacznej mierze jest to szacowanie czasu, nakładów pracy i innych zasobów niezbędnych do realizacji projektu. Różne elementy planu mogą być pożądane w zależności od tego kto jest odbiorcą planu. W projektach, których realizacji chcemy się podjąć w ramach kontraktów zewnętrznych, np. przetargów publicznych, istotną sprawą jest oszacowanie kosztów oraz czasu niezbędnego na realizację w postaci raportu (biznes planu) dla zarządu firmy, który ma podjąć decyzje w sprawie zdobycia kontraktu. W takim przypadku planowanie odbywa się na podstawie specyfikacji wymagań systemu przedstawionej przez klienta.

Gdy mamy do czynienia z projektami wewnętrznymi – najczęściej zanim powstanie specyfikacja, wówczas od zespołu wykonawczego zarząd firmy oczekuje oszacowania projektu na podstawie zdefiniowanego problemu lub pomysłu na wytworzenie produktu, który będzie miał wartość rynkową. W takich projektach wiedza na ich temat ewoluje w miarę rozwoju prac koncepcyjnych i pozyskiwania wiedzy z otoczenia, co umożliwi w kolejnych fazach projektu uszczegółowienie szacowania. Zaleceniem w takich projektach jest iteracyjne szacowanie ponownie po opracowaniu specyfikacji oraz po zakończeniu projektowania i wytworzeniu prototypu systemu. Po każdorazowym przeglądzie harmonogramu i budżetu, jeśli następują rozbieżności z planem wcześniejszym, to o tym fakcie kierownik projektu powinien powiadomić sponsora projektu. Sponsorem projektu w przypadku projektów zewnętrznych jest podmiot zamawiający projekt, w projektach wewnętrznych natomiast – zarząd firmy lub uprawniona osoba funkcyjna.

*Planujemy również po to, aby było wiadomo co musimy lub możemy zmieniać.*

KF

Na ogół planowanie rozumiane jest jako wytyczenie celów organizacji i określenie sposobu ich najlepszej realizacji. W procesie planowania występują takie elementy, jak podejmowanie decyzji, wybór kierunków działań oraz sprawność zarządzania. Planowanie jest również immanentną częścią projektu informatycznego, którego zadaniem jest osiągnięcie celu projektu z uwzględnieniem ograniczeń projektu, takich jak:

- **koszt**, czyli środki finansowe, które możemy wykorzystać do realizacji projektu – *budżet projektu*,

- **czas**, w jakim należy wykonać projekt – *harmonogram projektu*,

- **zakres**, czyli jaką postać finalną ma przedstawiać zrealizowany projekt w sensie funkcjonalności, użytych technologii, jakości, obszaru zastosowania itp. przekłada się bezpośrednio na *pracę*, jaką trzeba wykonać, aby osiągnąć oczekiwany cel projektu.

W wielu rozważaniach związanych z planowaniem projektów wyróżnia się często czwarty parametr, którym jest **jakość** [8, 12, 15, 28, 37, 40]. Korekta jednego z tych elementów wpływa na pozostałe dwa. Mimo że wszystkie trzy są ważne, zwykle jeden z wymienionych elementów będzie miał największy wpływ na projekt.

Związek między tymi elementami jest różny w różnych projektach i określa rodzaje problemów, jakie możemy napotkać oraz rozwiązania, jakie można zastosować. Wiedząc o ograniczeniach lub dopuszczalnej elastyczności, można łatwiej planować projekt i nim zarządzać. Należy również podkreślić, że duży wpływ na planowanie projektu informatycznego ma tzw. pula zasobów projektów (patrz, co to jest pula zasobów p. 2.3).

Plan jest najczęściej projekcją wyobraźni przyszłego kierownika projektu co do sposobu osiągnięcia celu.

## 2.2. Definiowanie celów projektu

Opis celów projektu jest bardzo ważnym i niekiedy trudnym zadaniem, wymagającym rozważenia wielu czynników, które w sposób mierzalny przedstawiają produkt (produkty), przez które osiągamy cele:

- biznesowe,
- jakościowe,
- technologiczne,
- konkurencyjne,
- inne cele.

Najczęściej mało się poświęca czasu na wyspecyfikowanie mierzalnych i porównywalnych efektów, które powinien wnieść projekt. Bardzo często jako cel projektu wskazuje się np. budowę systemu informatycznego, wdrożenie oprogramowania czy budowę sieci komputerowej i instalację oprogramowania. Wiadomo, że są to jedynie środki techniczne – narzędzia, która mogą być elementem być może podstawowym w realizacji projektu, ale celem zasadniczym projektu jest uzyskanie nowej jakości w organizacji, która jest beneficjentem projektu, wyeliminowanie procesów, które są przyczyną hamowania rozwoju lub braku konkurencyjności. Celem takich projektów może być usprawnienie procesów zarządczych, których wymiernym efektem może być zmniejszenie zapasów, kosztów transportu, jednostkowych kosztów obsługi klienta itp.

## 2.3. Zasoby projektu

W rozdziale tym przyjęto nomenklaturę i sposób zarządzania zasobami zaimplementowanymi w programie MS Projekt 2000.

We współczesnych firmach informatycznych jest rzadkością przydzielenie osoby do jednego projektu od jego rozpoczęcia aż do zakończenia, bez nakładania na nią dodatkowych, wykraczających poza jeden projekt, zobowiązań. Współużytkowanie zasobów między projektami pozwala na większą elastyczność i kontrolę w zarządzaniu zasobami. Współużytkowanie zasobów należy rozważyć, jeżeli spełniony jest któryś z podanych warunków:

1. **Nakładanie się projektów.** Może się zdarzyć, że konieczne będzie rozpoczęcie nowego projektu przed zakończeniem projektu wykonywanego aktualnie. W razie potrzeby dzielenia czasu między projektami, współużytkowanie zasobów między plikami tych projektów może pomóc w zapobieżeniu nadmiernej alokacji zasobów. Chcąc przenieść dane zasobów, takie jak stawki kosztów, z dotychczasowego projektu do nowego projektu, można utworzyć *pulę zasobów*, aby zawrzeć w niej zasoby oraz informacje o nich dla obu plików. Utworzenie puli zasobów i następujące po nim przeniesienie informacji o zasobach ułatwi przenoszenie danych ze starego do nowego pliku projektu.

2. **Organizowanie zasobów w obszary funkcjonalne.** Jeżeli trzeba przydzielić zasoby, które pracują nad wieloma projektami w ramach procesu zarządzania, na przykład rewidentów i księgowych, uzasadnione jest utworzenie dla nich puli zasobów. Następnie menedżer grupy funkcjonalnej może zbilansować ich obciążenie pracą i zastąpić lub ponownie przydzielić zasoby, aby zachować zgodność z harmonogramem. Jeżeli nie ma znaczenia, który zasób wykonuje dane zadanie, pula zasobów może być zarządzana poza zakresem projektu, w celu zapewnienia optymalnej efektywności harmonogramu pracy zasobów. Jeżeli natomiast należy zachować kontrolę nad tym, kto wykonuje jakie zadania, można skonfigurować proces zmian przydziałów zasobów, który umożliwi zatwierdzanie przydziałów zasobów przed dokonaniem zmian przydziałów w pliku projektu.

3. **Przewidywanie obciążenia pracą w wielu projektach.** Wykorzystanie puli zasobów może być bardzo wydajne w przewidywaniu obciążenia pracą osób, których zadania mają podobne opisy. Można przydzielić zasoby o nazwach ogólnych, jak choćby Architekt I i Architekt II odpowiednio dla młodszych i starszych członków personelu, lub oznaczyć różne poziomy doświadczenia zawodowego niezbędne w realizacji zadania. Po przypisaniu zadaniom w każdym zbliżającym się projekcie odpowiednich opisów prac, można współużytkować zasoby za pomocą nowej puli zasobów i można zobaczyć całkowitą pracę, przydzieloną do każdego opisu prac. Wartość nadmiernej alokacji każdego z opisów prac stanowi informację, jak wiele określonych zasobów potrzeba do wykonania pracy nad projektami zgodnie z bieżącymi harmonogramami projektów. Na

przykład nadmierna alokacja na poziomie 300% dla Architekta I oznacza, że do wykonania pracy potrzeba trzech młodszych architektów. Następnie, po uściśleniu listy zasobów projektu, można wprowadzić konkretne nazwy do każdego opisu prac i ponownie przydzielić pracę rzeczywistym osobom, którą ją wykonają.

### **Co to jest pula zasobów**

Pula zasobów umożliwia współużytkowanie zasobów przez wiele projektów. Używanie puli zasobów umożliwia sporządzanie harmonogramów dla zasobów pracy we wszystkich projektach, identyfikację konfliktów między przydziałami w różnych projektach i monitorowanie, wykorzystywania czasu zasobu w wielu projektach. Jeżeli informacje o ludziach lub sprzęcie pracującym nad zadaniami znajdują się w wielu plikach projektów, można użyć puli zasobów do centralizacji informacji o zasobach, takich jak nazwa zasobu, używany kalendarz, jednostki zasobu i tabele stawek kosztów, co ułatwi zarządzanie projektem. Każdy projekt, który używa zasobów z puli zasobów, jest nazywany plikiem współużytkującym. Jako puli zasobów można używać dowolnego, istniejącego pliku projektu, ale zaleca się utworzenie nowego pliku projektu tylko na informacje o zasobach, by jak najbardziej ułatwić zarządzanie informacjami o zasobach i przydziałami zadań między plikami współużytkującymi a pulą zasobów.

## **2.4. Definiowanie ograniczeń w projekcie**

Ograniczeniami w projekcie są czynniki, które mają podstawowy wpływ na opcje działań kierownika projektu. Typowe trzy główne ograniczenia to:

- Harmonogram – ograniczenia, takie jak stała data zakończenia lub termin ostateczny w przypadku głównych punktów kontrolnych.
- Zasoby – (materiał, wyposażenie, sprzęt i ludzie oraz skojarzone z nimi koszty) – ograniczenie, takie jak uprzednio zdefiniowany budżet.
- Zakres – ograniczenie, takie jak zakładana funkcjonalność, technologia, produkty itp.

Zmiana jednego z wymienionych ograniczeń zwykle wpływa na dwa pozostałe, a także na jakość projektu. Na przykład zmniejszenie czasu trwania projektu (harmonogram) może zwiększyć liczbę pracowników potrzebnych do realizacji planu (zasoby) oraz zmniejszyć liczbę właściwości cechujących produkt (zakres). Menedżer projektu musi określić, czy można zaakceptować taką degradację. Taki związek jest nazywany potrójnym ograniczeniem zarządzania projektem lub trójkątem ograniczeń projektu. Podczas procesu planowania należy sporządzić listę ograniczeń projektu, aby upewnić się, że wszyscy wykonawcy projektu zostali o niej powiadomieni i mogą

się do niej odnieść. Właściwym dla wykonawców jest także uzgodnienie sposobu ich reakcji na niespodziewane ograniczenia, które mogą ujawnić się w czasie trwania projektu. Na przykład, jeżeli koszty pracy okażą się wyższe od przewidywanych, to wykonawcy mogą zażądać zmniejszenia zakresu projektu.

Główne czynniki, które również należy uwzględnić w planowaniu projektu:

- pieniądze/budżet, jakim dysponujemy,
- czas, w którym projekt należy zrealizować,
- ludzie/nakład pracy, jaki wymaga realizacja projektu,
- miejsce, w którym projekt będzie realizowany.
- wyposażenie/warunki pracy oraz środki techniczne i narzędzia, którymi dysponujemy,
- komunikacja/lokalizacja zespołu, poczta, telefon, videokonferencje itp.,
- logistyka,
- wykształcenie członków zespołu,
- kompetencje posiadane,
- wiedza (praktyczna i teoretyczna),
- zdolności,
- umiejętności,
- outsourcing niektórych prac, zadań, zasobów.

Ponieważ przy realizacji projektów informatycznych część czynników jest stosunkowo łatwo mierzalna i porównywalna, jak pieniądze, czas, wyposażenie stanowiska pracy czy warunki pracy, tzw. standardy, więc o sukcesie projektu mogą zdecydować pozostałe czynniki, które wymagają większego doświadczenia i uwagi.

Etapy i czynności przygotowawcze związane z planowaniem projektu:

**Studium wykonalności projektu (SWP)** – stwierdza, czy dany projekt przy danych zasobach ma szansę wykonania (zakończenia się sukcesem).

**Inicjowanie projektu (IP)** – zbiór czynności, które należy podjąć przed formalnym uruchomieniem cyklu prac nad projektem:

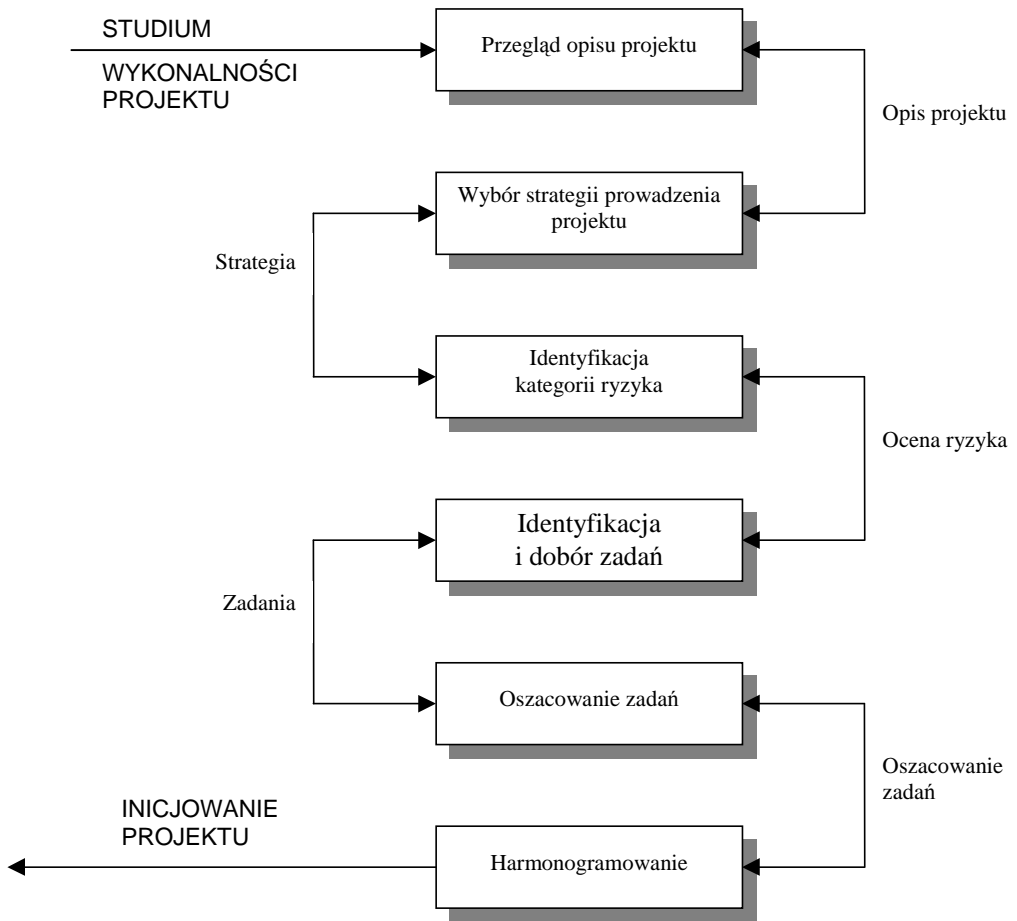
- opis rozwiązania technicznego,
- opis (wstępny plan) projektu (ang. *Business Case*),
- ustanowienie projektu.

Działania w obrębie inicjacji projektu zależą od punktu jego startu [11, 22, 26, 29].

Rozpoczynając prace nad SWP, zmierzamy do IP, ale zanim to ewentualnie nastąpi, należy wykonać prace przez wyspecjalizowane zespoły, które przedkładają dokumenty wymagane w danej firmie. Przykładowy schemat postępowania podano na rys. 2.2.

## Opis projektu

Opis projektu może być wykonywany według różnych, wcześniej przygotowanych formularzy, wzorców. Ich postać jest zależna od doświadczenia i obowiązujących



Rys. 2.2. Etapy i czynności przygotowawcze związane z planowaniem projektu

norm, zarządzeń czy ustaleń związanych z realizacją projektu przez dany podmiot. Przytoczono najczęściej spotykaną specyfikację zawartości opisu projektu:

- opis celów projektu,
- określenie zakresu, ogólna charakterystyka, jego otoczenie i umiejscowienie (fizyczne),
- określenie granic projektu i punktów kontrolnych, ograniczeń i założeń,
- zależności od innych projektów i powiązania z nimi,
- określenie strategii budowy (wydania, wersje, podprojekty – patrz dalej),
- oszacowanie ryzyka projektu,
- podział prac i oszacowanie nakładów (w cyklu budowy i działania),
- wstępny harmonogramu projektu,



- preliminarz kosztów,
- określenie struktury uczestników projektu (klienta, zespołu projektowego, innych),
- określenie wymaganych metryk jakości produktu,
- rozwiązania prawne,
- ustanowienie i rozpoczęcie projektu,
- mierniki sukcesu projektu.

### **Analiza opisu projektu**

Jedną z głównych przyczyn porażek projektów to niewłaściwa identyfikacja i brak zgodności celów między wykonawcą a klientem (patrz rozdz. 9) [52, 56]. Dlatego analiza opisu projektu powinna dotyczyć obydwu stron przedsięwzięcia i uwzględniać potrzebę uzyskania odpowiedzi na pytania:

- Czy cele projektu jasno odzwierciedlają potrzeby klienta?
- Czy projekt ma zdefiniowane produkty końcowe?
- Czy są sprecyzowane mierniki sukcesu?
- Czy cele projektu są perspektywiczne i osiągalne?
- Czy cele nie są wzajemnie sprzeczne (czy możliwy jest kompromis)?
- Czy cele wyznaczają w przybliżeniu przedział czasu dla ich osiągnięcia?
- Czy cele są zgodne z oczekiwaniami klienta i czy główne kierownictwo klienta jest zaangażowane w działania dla ich osiągnięcia?
- Czy cele zostały uzgodnione ze sponsorem projektu?

Szczególnie istotne jest uświadomienie zespołowi realizującemu projekt, że kluczową sprawą jest pamiętanie o zdefiniowanych i uzgodnionych celach projektu oraz prowadzenie okresowej weryfikacji ich zrozumienia przez poszczególnych wykonawców.

## **2.5. Strategia realizacji projektu**

Wybór strategii rozwoju projektu jest poprzedzony analizą wielkości projektu, horyzontu czasowego jego realizacji, poziomem dopuszczalnego ryzyka i innymi czynnikami. Wyróżnia się kilka strategii, które mają swoje nazwy:

- fazowa, monolityczna – sekwencja kolejno wykonywanych faz, dobra dla projektów o niskim poziomie ryzyka,
- wydania, wersje – wytwarzane są fragmenty (podsystemy, komponenty), inkrementalne podsystemy mogą powstawać w sekwencji lub równolegle, ich oddzielne wytwarzanie zmniejsza ryzyko ich uruchomienia
- szybkiej ścieżki, prototypowania – wytwarzany jest prototyp, następnie wyko-

nywana jest jego ocena, po której wytwarzany jest system, zalecana przy wysokim ryzyku,

- mieszana – fragmenty (podprojekty) powstają według różnych strategii, szczególnie przydatna dla dużych projektów obciążonych dużym ryzykiem

Tabela 2.1. Wybór strategii realizacji projektu w zależności od rozmiaru projektu i oceny ryzyka

Rozmiar projektu	Ocena ryzyka	Strategia
< 3 mies.	niskie	fazowa
	średnie	wydania
	wysokie	prototypowanie
3-6 mies.	niskie	fazowa lub wydania
	średnie	wydania
	wysokie	wydania lub prototypowanie
> 6 mies.	niskie	wydania
	średnie	wydania
	wysokie	mieszana lub prototypowanie

Każda strategia realizacji projektu powinna uwzględniać pryncypia w zakresie zarządzania, takie jak:

- zarządzanie wymaganiami – zapewnienie jednoznacznego, obiektywnego określania cech tworzonego rozwiązania oraz zapewnienie weryfikacji zgodności docelowego produktu z wymaganiami,
- kontrolę przebiegu projektu – możliwość bieżącego śledzenia faktycznego postępu prac i wczesne wykrywanie zagrożeń dla harmonogramu, budżetu i jakości tworzonego systemu,
- kontrolę kosztów utrzymania – możliwość realistycznego przewidywania kosztów przyszłych modyfikacji wdrożonego systemu.

## 2.6. Ocena ryzyka

Najczęściej zagrożenia (ryzyko projektu) ocenia się w dwóch głównych obszarach dotyczących uzasadnienie biznesowego projektu, tj. ryzyko biznesowe i ryzyko projektu. Czynniki, jakie należy brać pod uwagę w cenie można pogrupować te, które dotyczą:

### 1. Złożoności systemu lub produktu:

- funkcje i algorytmy,
- złożoność sterowania, wyjątków i/lub operacji matematycznych,
- procedury współdziałania z użytkownikiem,

- znaczący wpływ na pracę ludzi,
  - wymagania jakościowe i efektywnościowe,
  - duża ilość danych, żądania krótkiego czasu odpowiedzi,
  - wymagania technologii,
  - istotne zastosowanie specyficznego sprzętu/oprogramowania.
2. Klienta i środowiska docelowego:
    - liczba węzłów i użytkowników,
    - poziom wiedzy użytkowników i ich udział w projekcie,
    - priorytetowość systemu i jego znaczenie dla zamawiającego,
    - konieczność wprowadzenia zmian w biurach, oddziałach, procedurach.
  3. Środowiska budowy systemu.
  4. Harmonogramów, ich niezmienności bądź elastyczności.
  5. Poziomu wiedzy i doświadczenia zespołu projektowego, stabilności.
  6. Oszacowania ram czasowych.
  7. Korzystania z zewnętrznych dostawców i podwykonawców.
  8. Fizycznego i technologicznego środowiska realizacji projektu.
- W przypadku oceny ryzyka jako wysokiego:
9. Wskazania do obniżenia złożoności projektu.
  10. Udokumentowania obszarów wysokiego ryzyka.
  11. Formalnego memorandum.
- Patrz także: [22, 28, 31, 32, 33].

## 2.7. Struktura projektu – dekompozycja projektu na zadania – WBS

W każdym projekcie PM dekomponuje cały projekt na WBS (ang. *Work Break-down Structure*) do poziomu *zadania* (ang. *task*), które definiują czynności, jakie należy zrealizować w celu wyprodukowania określonego produktu, usługi, dokumentacji itp. w zależności od tego do czego ma służyć realizacja zdefiniowanego zadania. Zadanie może się dzielić na *czynności*. Przyjmuje się, że zadanie powinno być przypisane w realizacji dla pojedynczej osoby i czas jego trwania od 1 do kilku dni, ponadto na być mierzalny i dać się skontrolować co do wykonania oraz jakości.

### Definicja struktury WBS

Głównym zadaniem kierownika projektu jest właściwe określenie elementów składowych prac, które należy zrealizować w projekcie. Struktura WBS reprezentuje pracę nad wytworzeniem indywidualnych komponentów i pracę nad integracją

komponentów w projekt. Głównym celem struktury WBS jest przejrzysta i adekwatna do rodzaju projektu organizacja powiązań i współdziałania wytwarzanych produktów zmierzających do osiągnięcia celu projektu. Umożliwia graficzne wyobrażenie i sprawdzenie czy dany projekt ma szanse realizacji. Wszystko to, co znajduje się w projekcie musi się znajdować w strukturze WBS. Jeśli czegoś nie ma uwzględnionego w WBS, to oznacza, że nie ma tego w projekcie. WBS może być strukturą hierarchiczną drzewa. Począwszy od korzenia do liści, wzrasta stopień szczegółowości opisu WBS. Komponentami WBS mogą być zarówno produkty, jak i usługi [3, 26, 32]. Plan projektu powinien być dokładny, zawierać wszystkie zadania, czynności niezbędne do osiągnięcia zamierzonych celów projektu. Struktura WBS powinna to umożliwiać.

### WBS i jego rodzaje

**WBS produktowy** stanowi perspektywę produktów. Fazy WBS produktowego to najbardziej ogólne komponenty, które muszą być zrealizowane w projekcie.

**WBS fazowy** opiera się na modelu fazowym cyklu życia oprogramowania – tzw. *faz*, które składają się z kilku działań, a te z kolei z aktywności. Faza (ang. *Phase*) – faza rozwoju w produkcie lub czynności, jedna z faz modelu cyklu życia oprogramowania, np. faza analizy (ang. *Analysis phase*) – jest to jedna z dodatkowych faz modelu kaskadowego cyklu życia oprogramowania, w której budowany jest logiczny model systemu. Celem fazy analizy jest udzielenie odpowiedzi na pytanie: jak system ma działać? *Działaniem* w tej fazie *i* jest udzielenie odpowiedzi na pytanie: jak system ma działać?, to następuje przez *aktywność*, wynikiem której jest logiczny model systemu, opisujący sposób realizacji przez system postawionych wymagań, lecz abstrahujący od szczegółów implementacyjnych. WBS jest strukturą podziału pracy, jaką należy wykonać, aby osiągnąć zamierzone cele projektu. WBS stanowi hierarchiczną strukturę drzewa. Na poziomie zerowym jest umieszczona nazwa projektu. Jednakże nie oznacza to, że projekt jest częścią WBS. Stanowi on raczej opis, jakiego projektu dotyczy dany WBS. Idea tworzenia WBS: ogół pracy dzielimy na fazy, następnie fazy dzielimy na zadania, a te na aktywności. WBS: Faza → zadanie → aktywność [3]. Fazy tworzą najbardziej ogólny podział pracy. Znajdują się one na pierwszym poziomie WBS, zaraz po nazwie projektu. Zadania znajdują się na niższym poziomie, poniżej faz. Zadania mogą być dekomponowane na aktywności lub na inne zadania. Składowe danego zadania znajdują się na niższym poziomie. Tak więc zadania mogą znajdować się na poziomie drugim, trzecim itd. Ważne jest, że na ostatnim poziomie dekompozycji danego zadania muszą znajdować się aktywności (dostarczające produktów). Aktywności znajdują się na najniższym poziomie WBS. Reprezentują one produkty, które są przydzielane konkretnym osobom. Osoby te wykonują pracę potrzebną do stworzenia produktu. Tak więc aktywność jest kombinacją produktu i procesu. Tworząc strukturę WBS, należy zwrócić uwagę na numerowanie kolejnych

komponentów WBS. Każdy element WBS jest unikatowy. Na poziomie zerowym znajduje się jeden komponent – nazwa projektu o numerze 1.0, na poziomie pierwszym znajdują się elementy numerowane od 1.1 do  $1n$ , gdzie  $n$  jest liczbą faz, na poziomie drugim znajdują się składowe poszczególnych faz, np. 1.1.1 do 1.1 $x$ , gdzie  $x$  – liczba komponentów składowych pierwszej fazy, 1.2.1 do 1.2 $y$ , gdzie  $y$  – liczba komponentów drugiej fazy.

**WBS strukturalny** tworzymy w celu przedstawienia organizacji zaangażowanej w realizację projektu. Należy uwzględnić takie elementy współdziałania, które wynikają z umowy na tworzenie produktów oraz relacji z wykonawcami, która ma zabezpieczyć wykonanie projektu.

### Etapy tworzenia WBS produktowego

1. Pierwszy poziom tworzymy z produktów, co do których jesteśmy zobowiązani w umowie z klientem. Fazy w WBS produktowym są produktami. Produkty te są wyszczególnione w umowie, dokumentacji, specyfikacji projektu. Ich nazwy powinny być parą rzeczowników lub rzeczownika z przymiotnikiem, np. „Specyfikacja”, „Specyfikacja projektu”.

2. Dla każdego produktu na najwyższym poziomie należy dokonać dekompozycji do części składowych. Każda część składowa staje się komponentem – częścią WBS. WBS powinien być tak tworzony, aby osoba postronna mogła zrozumieć cele wykonania zadania związanego z danym produktem. Podział produktu na wyższym poziomie na produkty na niższym poziomie musi mieć sens. Każdy z produktów na niższym poziomie powinien odróżniać się od pozostałych oraz być odrębną częścią produktu wyższego poziomu.

3. Kontynuacja dekompozycji powinna trwać do momentu osiągnięcia odpowiedniego poziomu szczegółowości.

Zadania na najniższym poziomie – aktywności powinny spełniać następujące warunki:

- powinny być możliwe do wykonania od jednego do dziesięciu dni,
- czas ich wykonania nie krótszy od sporządzenia raportu,
- dla każdego zadania – aktywności można oszacować koszty i czas oraz przydzielić odpowiednie osoby.

Zadania te powinny być nazwane w formie bezokolicznika: np. „Tworzenie specyfikacji projektu”.

4. W WBS powinny być opisane najważniejsze czynności prowadzące do powstania produktu. W tworzeniu np. oprogramowania głównymi etapami tworzenia jest system, podsystem i funkcja. Należy tutaj uwzględnić wyniki testów, kompilację kodu i wymaganą dokumentację.

Każdy wymagany produkt należy zdekomponować do czynności, które są wymagane do jego utworzenia.

5. W miarę rozwoju WBS może mieć kilka aktywności na poziomie drugim, trzecim itd. Jednakże należy szczególnie zwrócić uwagę, aby liczba aktywności związana z danym produktem – zadaniem nie była ani za duża, ani za mała. Zaleca się, aby każdy produkt – zadanie składało się z  $7 \pm 2$  aktywności. Jeżeli jest ich od trzech do pięciu, to należy się zastanowić czy nie można ich dołączyć do innego produktu – zadania. Jeżeli jest ich więcej niż dziesięć należy spróbować podzielić dany produkt – zadanie (związane z tymi aktywnościami) na mniejsze produkty. Jednakże są to tylko zalecenia, jeśli do konkretnego WBS trudno jest zastosować powyższe wskazówki, należy je zaniechać. Struktura WBS powinna być logiczna, tzn. produkty składowe znajdujące się w WBS powinny tworzyć produkt na wyższym poziomie oraz powinny być znacząco różne, nie pokrywać się.

6. Sprawdzenie poprawności WBS rozpoczyna się od przechodzenia z najniższego poziomu do najwyższego, inaczej od dołu do góry – od liści do korzenia. Przechodząc z niższego poziomu do wyższego, należy sumować produkty (aktywności lub zadania) i sprawdzać czy tworzą one produkt na wyższym poziomie. W ten sposób sprawdzamy czy WBS jest kompletny, czy czegoś w nim nie brakuje. Jeżeli okaże się, że są jakieś luki w danym WBS, to należy go uzupełnić.

7. Ostatni etap – to sprawdzenie zgodności powstałego WBS z celami projektu, czy przez utworzone produkty można zrealizować cele projektu [8].

#### **Etapy tworzenia WBS fazowego:**

1. Na poziomie pierwszym znajdują się fazy cyklu oprogramowania.
2. Następnie należy zidentyfikować produkty, co do których jesteśmy zobowiązani w umowie z klientem. Produkty te umiejscawiamy pod odpowiednią fazą.
3. Dane produkty dekomponujemy podobnie jak w WBS produktowym.

#### **Etapy tworzenia WBS strukturalnego:**

1. Na pierwszym poziomie znajduje się firma klienta oraz firmy, które deklarują się do wykonania projektu.
2. Drugi poziom odzwierciedla zawarcie umowy. Połączenie firmy klienta z konkretnym wykonawcą. Tworzona jest organizacja, która zajmuje się realizacją projektu.
3. Dla danej organizacji wykonującej projekt jest tworzona struktura podziału pracy według WBS produktowego lub WBS fazowego.

#### **Inne struktury podziału pracy**

Struktura podziału pracy kontraktowa (*Contractual WBS – CWBS*)

Jest tworzona w celu przedstawienia klientowi. Wykonawca projektu używa kontraktowej struktury podziału pracy do zdefiniowania raportów, jakie będzie przedstawiał klientowi po zakończeniu realizacji prac wyspecyfikowanych w kontrakcie.

CWBS zawiera więcej szczegółów związanych z zarządzaniem pracą nad projektem, po zakończeniu której następują zobowiązania stron projektu. Struktura ta jest pomocna w egzekwowaniu terminów płatności przez klienta za projekt.

Struktura podziału pracy organizacyjna (*Organizational Breakdown Structure – OBS*)

Przedstawia strukturę organizacji realizującej projekt. Pokazuje, które elementy pracy są przydzielone, do których jednostek organizacji. Taka struktura jest stosowana w przypadku rozproszonych zespołów lub ma specyficzną wiedzę dziedzinową.

**Zasoby** (*Resource Breakdown Structure – RBS*)

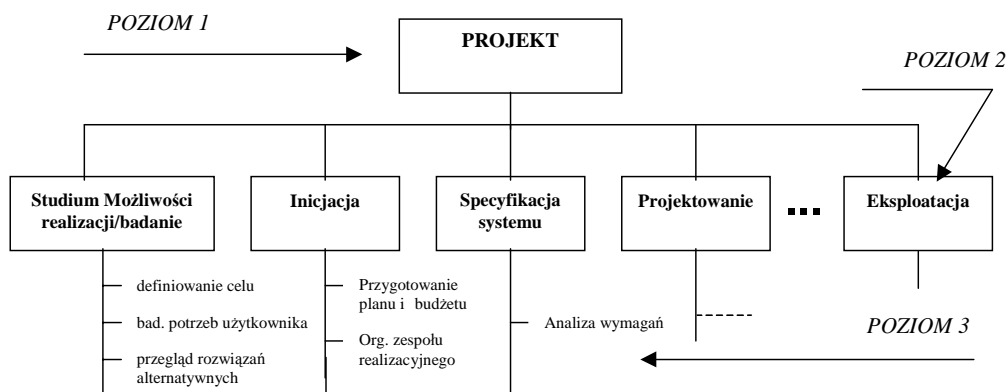
Odmiana organizacyjnej struktury podziału pracy. Jest używana, kiedy zadania są przydzielone konkretnym osobom.

**Koszty** (*Bill of Materials*)

Prezentuje hierarchiczną strukturę zadań, podzadań, komponentów potrzebnych do wyprodukowania produktu.

**Projektowa struktura** (*Project Breakdown Structure – PBS*)

Projektowa struktura podziału pracy jest wykorzystywana w aplikacjach, gdzie pojęcie struktury WBS jest niepoprawne w powiązaniu ze strukturą zasobów (RBS) [2].



Rys. 2.3. Przykład WBS fazowego, prace poziomu 3 można dzielić na działania (czynności) (ang. *activities*)

Aby praktycznie zilustrować etapy, jak również sposób tworzenia diagramów WBS, możemy korzystać z opisu faz według Coopers & Lybrand (tabela 2.2), gdzie jest kolumna z typowymi fazami projektu, kolumna opisująca zespół czynności, które realizujemy w ramach fazy oraz kolumna produktów końcowych, które powinny powstać na zakończenie danej fazy. Tak „rozpisany projekt” umożliwi łatwe tworzenie zarówno WBS produktowego, jak również fazowego (rys. 2.3).

Dla pełnego udokumentowania wszystkich produktów możemy zdefiniować POZIOM 4, na którym wyspecyfikujemy wytworzone w projekcie np. dokumentację, oprogramowanie, instrukcje, zainstalowany sprzęt, uruchomione oprogramowanie itd.

Tabela 2.2. WBS: Standardowe fazy cyklu życia projektu (według Coopers &amp; Lybrand)

Faza	Czynności	Produkt końcowy
Studium możliwości realizacji/badanie	Zdefiniuj problem. Zbadaj wymagania użytkownika. Oceń rozwiązania alternatywne. Zaleć kierunek działania.	Sprawozdanie studialne
Inicjacja	Przygotuj plan i budżet. Przygotuj opis działalności firmy.	Projekt planu technicznego. Projekt planu zasobów. Projekt uzasadnienia przedsięwzięcia. Szczegółowy plan dla następnej fazy. Aprobata dalszych działań.
Specyfikowanie	Analizuj szczegółowo wymagania użytkownika. Określ kryteria akceptacji. Wymyśl strategię implementacji. Opracuj plany.	Specyfikacja wymagań użytkownika. Kryteria akceptacji. Strategia instalacji i przejścia. Strategia szkolenia. Szczegółowy plan dla następnej fazy. Akceptacja dalszych działań.
Projektowanie	Stwórz projekt systemu. Wymyśl strategię. Opracuj plan.	Projekt systemu. Strategia budowy systemu. Strategia testowania. Szczegółowy plan dla następnej fazy. Akceptacja dalszych działań.
Realizacja	Projektuj, pisz i testuj programy. Skompletuj dokumentację. Przeprowadź testy systemu. Opracuj plany.	Moduły. Programy. Procedury. Dokumentacja systemu. Materiały do szkoleń. Szczegółowy plan dla następnej fazy. Akceptacja dalszych działań.
Instalowanie	Opracuj zasady konwersji. Przeprowadź testy akceptujące. Opracuj plany.	System zatwierdzony przez użytkownika. Szczegółowy plan dla następnej fazy. Akceptacja dalszych działań.
Eksploatacja	Przegląd po implementacyjny.	System nadający się do eksploatacji i utrzymania. Raport końcowy.

## 2.8. Szacowania w projekcie

*Policz to co można policzyć, zmierz to co można zmierzyć,  
a to co niemierzalne uczynь mierzalnym.*

Galileo Galilei

Wymiarowanie systemów informatycznych, w tym szacowanie poszczególnych elementów projektu, takich jak czas realizacji, pracochłonność, koszty, wydajność, zużycie materiałów i inne, są przedsięwzięciem złożonym. Szczególnym przedmiotem szacowania jest ta część projektu informatycznego, która dotyczy oprogramowania. W przypadku takich nauk, jak fizyka, elektronika, ekonomia sprawa jest dość oczywista i uwaga badaczy skoncentrowana jest wokół jednostki miary i metody powtarzalności pomiaru.



Tabela 2.3. Fazy cyklu życia projektu obiektowego dostosowane na potrzeby projektu grupowego

Faza	Czynności	Produkty
Planowanie wstępne (założenia)	Poznanie celów, odpowiedzialności i harmonogramu. Analiza problemu. Określenie osób pełniących rolę klientów	Powiązanie grupy z tematem projektu
Studium wykonalności	Identyfikacja podstawowych wymagań. Analiza wykonalności. Przygotowanie raportu wykonalności.	Raport wykonalności Raporty spotkań
Planowanie projektu (ang. <i>Preliminary project plan</i> )	Organizacja grupy, przypisanie ról. Określenie planu działań, oczekiwanych produktów, zasobów, przydziału prac. Określenie ryzyka, określenie strategii. Przyjęcie planu kontroli jakości. Opracowanie harmonogramu. Przygotowanie wymaganych raportów.	Plan projektu. Założenia strategii minimalizacji ryzyka. Plan nadzoru jakości. Szczegółowy plan fazy następnej. Raport przebiegu prac (w tym spotkań)
Specyfikacja wymagań systemowych	Identyfikacja wymagań w oparciu o analizę dokumentów, wywiady, pytania, etc. Specyfikacja wymagań. Weryfikacja i akceptacja. Działania zmierzające do zapewnienia jakości. Przygotowanie wymaganych raportów.	Dokument specyfikacji wymagań użytkowych. Plan (projekt) testów akceptacyjnych. Słownik danych (wstępny). Szczegółowy plan fazy następnej. Raport zmian. Raport przebiegu prac (w tym także spotkań i działań pro jakościowych)
Specyfikacja wymagań na oprogramowanie (modelowanie)	Analiza wymagań. Modelowanie i specyfikacja. Uściślenie słownika danych. Weryfikacja i akceptacja. Działania zmierzające do zapewnienia jakości. Uściślenie założeń projektowych i implementacyjnych. Przygotowanie wymaganych raportów i dokumentacji.	Modele systemu: Model klas Model funkcjonalny Model dynamiczny Słownik danych Projekt testowania funkcjonalnego. Podręcznik użytkownika (szkic). Szczegółowy plan fazy następnej. Raport zmian. Raport przebiegu prac
Projektowanie systemu	Modelowanie i specyfikacja. Uściślenie słownika danych. Działania zmierzające do zapewnienia jakości. Uściślenie założeń projektowych i implementacyjnych. Przygotowanie wymaganych raportów i dokumentacji.	Dokumentacja projektu systemu. Słownik danych. Projekt testowania integracyjnego. Szczegółowy plan fazy następnej. Raport zmian. Raport przebiegu prac
Projektowanie klas	Projektowanie klas. Uściślenie słownika danych. Działania zmierzające do zapewnienia jakości. Uściślenie założeń implementacyjnych. Przygotowanie wymaganych raportów.	Dokumentacja projektu klas. Słownik danych. Projekt testowania obiektów. Szczegółowy plan fazy następnej. Raport zmian. Raport przebiegu prac
Implementacja, integracja i testowanie	Implementacja obiektów. Testowanie. Uzupełnienie słownika danych. Działania zmierzające do zapewnienia jakości. Przygotowanie wymaganych raportów i dokumentacji.	Oprogramowanie projektu. Słownik danych. Raport testowania obiektowego, integracyjnego, funkcjonalnego. Dokumentacja (szkic). Szczegółowy plan fazy następnej. Raport zmian. Raport przebiegu prac
Finalizacja	Dyskusja testów akceptacyjnych. Dokumentowanie. Raport.	Raport testowania akceptacyjnego. Dokumentacja (szkic). Raport finalny

W przypadku informatyki problem dotyczy złożoności oprogramowania, czyli na ogół relacji, jakie występują między dwoma, trzema programami, który jest bardziej złożony, trudniejszy w pielęgnacji, implementacji algorytmów, testowaniu, wdrożeniu itd.

Szacowanie zadań związanych z implementacją oprogramowania jest zagadnieniem trudnym, wymagającym współdziałania kierownika projektu z zespołami przewidzianymi do realizacji projektu, dostępu do informacji na temat podobnych przedsięwzięć lub jego fragmentów, aby można się posłużyć np. techniką analogii zamiast regułą „palca i sufitu”. Podstawą prac nad szacowaniem zadań jest opracowanie w miarę dokładnej struktury projektu, którą należy dekomponować do zadań, które realizują pojedynczy wykonawcy lub specjalizowane zespoły w stosunkowo krótkim czasie, np. kilku dni. Wprowadza się tu takie pojęcia, jak:

- nakład pracy (ang. *effort*) – (MM – *man-months*, PM – *preson-months*),
- czas trwania (ang. *duration*) – (Mo – *months*),
- obciążenie ludzi (ang. *manpower loading*) – liczba wymaganych pracowników przydzielonych do projektu w funkcji czasu.

Dla oszacowania kosztu całego projektu, pojedynczej fazy, aktywności, są potrzebne kosztowe informacje [47, 51, 52]:

- przed startem projektu (dla analizy koszty/zysk, negocjacji kontraktu),
- podczas realizacji projektu (w celu zarządzania projektem, planowania, monitorowania i kontroli),
- muszą być aktualizowane w trakcie prowadzenia projektu.

### Metody szacowania zadań

Przez szacowanie zadań należy rozumieć różne obszary mierzenia zadań, które należy wykonać, aby wytworzyć oprogramowanie, między innymi:

- prognozowana pracochołność,
- koszty,
- niezawodność,
- złożoność,
- złożoność implementacji algorytmów,
- jakość,
- przenaszalność.

Nie ma uniwersalnej metody, która by w zadowalający sposób określała wszystkie obszary oprogramowania, i to niezależnie od jego charakteru, wielkości itd.

- Metoda punktów funkcyjnych (PF) jest stosowana przede wszystkim do szacowania pracochołności i jakości oprogramowania.
- Modele parametryczne, np. COCOMO [Boehm B. W., Software Engineering Economics, Prentice Hall, 1981, Putnam L. H., A General Empirical Solution to

the Macro Software Sizing and Estimating Problem, IEEE Transaction of Software Engineering, SE-4, July 1978] znane jako najlepsze do szacowania kosztów [4].

- Miary niezawodności oprogramowania opierają się na określeniu średnich czasów bezawaryjnej pracy oprogramowania, są to głównie modele niezawodnościowe.

Należy wspomnieć o mikrotechnice szacowania zadań, faz, Wide-Band Delphi, szacowania całkowitego wysiłku przedsiębiorstwa oraz przez eksperta lub oprogramowanie. Z innymi technikami szacowania projektów można się zapoznać w pracy Z. Szyjewskiego [47]. W dalszej części pracy szczegółowo omówiono metodę PF.

## 2.9. Metoda punktów funkcyjnych

W późnych latach siedemdziesiątych IBM potrzebował wynaleźć metodę oceny kosztów rozwoju aplikacji niezależną od języka, w którym dana aplikacja ma być stworzona. Zlecił realizację takiego projektu jednemu ze swoich pracowników Allanowi Albrechtowi, który w 1979 roku zaprezentował wyniki swoich prac jako *metodę punktów funkcyjnych* podczas konferencji w Monterey w Kalifornii [Albrecht A.J., Measuring Applications Development Productivity, Proceedings of IBM applications Division Joint SHARE/GUID Symposium, Monterey, CA, 1979], [1, 26]. W początkowym okresie pojawiło się sporo zarzutów wobec tej metody, ze względu na bardzo ograniczoną liczbę parametrów wejściowych, które wykorzystywano, ale była rozwijana i w 1984 opublikowano wersję, która uwzględniała 14 współczynników mających wpływ na projekt. Ta wersja metody zaczęła zdobywać coraz większą liczbę zwolenników.

Punkty funkcyjne (PF) są rozumiane jako miara wielkości aplikacji komputerowych i projektu, które należy stworzyć. Jest to miara stworzona głównie na użytek szacowania wielkości i kosztów projektu, które np. negocjujemy we wstępnej fazie projektu z klientem. Podstawą mierzenia jest planowanie funkcjonalności (inaczej specyfikowanie potrzeb użytkownika co do funkcjonalności, interfejsu, wielkości i ilości zbiorów danych itd.). Jest ona niezależna od języka programowania, metodologii rozwoju, technologii lub zdolności grup projektowych użytych do wytworzenia aplikacji. Fakt iż Albrecht po raz pierwszy użył jej do szacowania pracochłonności (wysiłku) jest prostą konsekwencją tego, że wielkość projektu jest podstawowym czynnikiem wpływającym na pracochłonność projektu.

Metoda PF nie jest doskonałym miernikiem pracochłonności stworzenia aplikacji lub wyceny jej wartości biznesowej, aczkolwiek wielkość projektu podana w punktach funkcyjnych jest ważnym czynnikiem w mierzeniu każdej z tych dwu wartości. Ilustruje to prosta analogia w handlu nieruchomościami. Koszt wybudowania budynku A o powierzchni 100 m<sup>2</sup> jest zwykle mniejszy od kosztu wybudowania budynku B o

powierzchni 400 m<sup>2</sup>. Jednakże wiele atrybutów, takich jak marmurowe łazienki, armatura i podłogi może wpłynąć na to, iż mniejszy dom może być bardziej kosztowny. Inne czynniki, takie jak lokalizacja i liczba sypialni może także uczynić mniejszy dom bardziej wartościowym miejscem zamieszkania. Tym samym koszt wybudowania 1 m<sup>2</sup> w budynku A i B mogą znacznie się różnić oraz ich wartość rynkowa niekoniecznie musi odzwierciedlać poniesione nakłady finansowe.

Można nadmienić dlaczego punkty funkcyjne nie mierzą wartości projektu oraz wskazać powody, które decydują, że warto używać punktów funkcyjnych:

1. Miara produktywności – wiele wykonawców projektów doszło do wniosku, że pomimo prowadzenia szerszej działalności informatycznej znaczną część zaangażowanych zasobów bazowych lokują w produkcję oprogramowania. Policzenie kilku wariantów rozwiązania tematu za pomocą punktów funkcyjnych daje im możliwość oceny jak dobrze sobie radzą w tej dziedzinie.

2. Wspomaganie szacowania rozwoju – od początku punkty funkcyjne używane były jako technika do szacowania. Szacowanie wielkości projektu jest oczywiście potrzebne do szacowania kosztów aplikacji, co daje nam pojęcie o sposobie jej wytwarzania. Nawet dla strategicznych projektów, które nie potrzebują żadnej ilościowej oceny, właściwe szacowanie jest potrzebne w celu właściwego przydziału pracowników.

3. Monitorowanie umów zewnętrznych (ang. *Outsourcing*) – firmy zlecające komuś na zewnątrz znaczące części swoich zadań oczekują, że dostawca dostarczy produkt według specyfikacji, na odpowiednim poziomie jakości, oczekują zatem związku z kosztami, które mają ponieść firmy zewnętrzne. Użycie metody PF w celu zademonstrowania zgodności swych szacunków, adekwatną do skali produkcji oprogramowania, jest podstawą negocjowania ceny usługi.

4. Pomoc w decyzjach biznesowych – firmy muszą analizować każdy pakiet aplikacji w projekcie. Wielkość w punktach funkcyjnych jest atrybutem, który musi być śledzony dla ilości aplikacji w projekcie. Wraz z innymi danymi pozwoli na decyzje dotyczące ponownego wykorzystania, wycofania lub zmodyfikowania aplikacji.

5. Normalizacja innych miar – aby pokazać właściwy sens niektórych danych, należy je porównać z punktami funkcyjnymi. Na przykład: informacje, że aplikacja o rozmiarze 100 punktów funkcyjnych, posiadająca 100 defektów, jest niezbyt dobrą wiadomością. Ta sama ilość dostarczanych defektów dla aplikacji o rozmiarze 10 000 punktów funkcyjnych jest już znacznie lepszym wskaźnikiem jakości oprogramowania.

Podstawowe pojęcia i wzory stosowane w metodzie PF

**Granice systemu** (ang. *System Boundaries*) – jawnie określone granice systemu poddawanego mierzeniu. Należy wyodrębnić granicę pomiędzy projektem lub aplikacją z punktu widzenia użytkownika.

**ILF** (ang. *Internal Logical File*) – wewnętrzny plik logiczny – grupa danych i rekordów powiązanych ze sobą i utrzymywanych wewnątrz granic sytemu podtrzymywana przez zewnętrzne wejście EI (*External Omput*).

**EIF** (ang. *External Interface File*) – zewnętrzny plik interfejsowy – grupa danych i rekordów powiązanych ze sobą i utrzymywana na zewnątrz granic systemu, która jest używana tylko jako referencje wewnątrz systemu.

**RET** (ang. *Record Element Type*) – rekord, zbiór powiązanych ze sobą logicznie danych identyfikowalny przez użytkownika znajdujący się wewnątrz ILF lub EIF.

**FTR** (ang. *File Type Referenced*) – plik, do którego odwołują się transakcje. Musi być to ILF lub EIF.

**DET** (ang. *Data Element Type*) – pojedyncza dana identyfikowalna z punktu widzenia użytkownika. Niepowtarzalne pole DET jest informacją, która jest zmienną, a nie stałą. Zmienne pole jest odczytywane z pliku lub stworzone za pomocą DET-ów zawartych w FTR. Dodatkowo DET może wywoływać transakcje lub może być dodatkową informacją dotyczącą danej transakcji. Jeśli DET ma wiele wystąpień (jest rekursywny), to tylko jego pierwsze wystąpienie powinno być brane pod uwagę. IFPUG (*International Function Point Group*) [26] podaje szczegółowe sposoby rozpoznawania i liczenia DET dla systemów z GUI oraz systemów czasu rzeczywistego.

**EO** (ang. *External Output*) – zewnętrzne wyjście – proces, w czasie którego przetworzona grupa danych przekracza granice systemu z wewnątrz na zewnątrz systemu. Dodatkowo może uaktualniać ILF. Dane tworzą raporty lub pliki wyjściowe wysyłane do innych aplikacji. Są one tworzone na podstawie jednego lub więcej ILF oraz EIF.

**EI** (ang. *External Inputs*) – zewnętrzne wejście – proces, w czasie którego dane przekraczają granice systemu z zewnątrz do wewnątrz. Mogą one pochodzić z ekranu (klawiatury), przez które wprowadzamy dane lub inne aplikacje. Dane te mogą służyć do uaktualnienia jednego lub więcej ILF. Dane te mogą być danymi kontrolnymi lub operacyjnymi. Jeśli są to dane kontrolne, to nie muszą uaktualniać ILF.

**EQ** (ang. *External Inquiries*) – informacje zewnętrzne – proces, w którym dane wychodzą poza granice systemu. Nie może zawierać przetworzonych lub obliczonych danych wewnątrz modułu. To ilość danych, które otrzymujemy w wyniku zewnętrznych zapytań do systemu.

Wyróżniamy 3 podstawowe typy liczenia :

- Dla projektu (ang. *development*),
- Dla aplikacji (ang. *application*),
- Dla aplikacji modyfikowanej (ang. *enhancement*).

Trzeci z wymienionych typów nie różni się zbyt wiele od drugiego typu. W procesie liczenia dla aplikacji modyfikowanej badamy wpływ dokonanych zmian (zliczamy usunięte zmodyfikowane i dodane funkcjonalności), korzystając z bazowej liczby punktów obliczonej dla aplikacji przed zmianami.

**VAF** (ang. *Value Adjustment Factor*) – czynnik modyfikujący wartość punktów funkcyjnych. Ma on za zadanie modyfikację liczby punktów otrzymanych z bazowego

liczenia przez uwzględnienie wiadomości o realizowanym projekcie. Uzyskuje się go przez odpowiedzenie na 14 pytań związanych z projektem. Odpowiedzią jest ustalenie ważności podanego współczynnika dla naszego systemu (w skali 0–5). Końcową wartość otrzymujemy za pomocą wzoru :

$$VAF = 0,65 + \left[ \left( \sum_{i=1}^{14} Ci \right) / 100 \right]$$

gdzie:  $Ci$  – 14 współczynników mających wpływ na projekt.

Przedstawiono kroki, które należy wykonać w celu kalkulacji projektu według metody PF, zgodne z podręcznikami opublikowanymi przez IFPUG [26]:

**1. Zaplanuj liczenie.** Ten krok obejmuje wybór rodzaju liczenia oraz zdefiniowanie granic liczenia. Obejmuje także bardzo ważny krok związany z identyfikacją i przydzieleniem eksperta systemowego.

**2. Wytlumacz proces liczenia.** Jeśli korzystamy z pomocy eksperta systemowego, będziemy potrzebowali wytłumaczyć mu, co zamierzamy robić, po co liczymy, co zamierzamy zrobić z uzyskanymi informacjami i inne podobne sprawy. Nie będziemy przecież co chwilę przerywać liczenia, po to by tłumaczyć, że użycie punktów funkcyjnych jest konieczne.

**3. Policz VAF.** IFPUG poleca zrobić to na końcu, lecz w czasie liczenia eksperci często narzekają, że poszczególne procesy są zbyt słabo ocenione. Można powiedzieć, że VAF weźmie to później pod uwagę i poprawi ich ocenę. Podzielenie się wiadomością o złożoności systemu utrzymuje osoby związane z liczeniem w ciekawości.

**4. Policz typy danych funkcyjnych.** Krok ten obejmuje identyfikację ILF oraz EIF. Zadając pytania ekspertowi o główne kategorie danych oraz studiując model danych, uzyskamy podstawę do ustalenia grup danych powiązanych logicznie, co następnie ułatwi liczenie złożoności transakcji.

**5. Zidentyfikuj transakcje.** Obejmuje to identyfikację EI, EO, EQ. Jest to zwykle najdłuższa część. Identyfikacja transakcji oraz ocenianie ich złożoności na podstawie liczby danych, z których korzysta.

**6. Wykonaj obliczenia.** Obejmuje zsumowanie punktów oraz przemnożenie otrzymanego wyniku przez VAF

**7. Weryfikacja wyników.** Po zakończeniu procesu liczenia powinno się wyniki przekazać ekspertowi w celu weryfikacji tego, czy jakaś funkcjonalność zawarta w systemie nie została pominięta.

**8. Pokazanie wyników.** Jednym z podstawowych powodów niepowodzeń przygotowywanych metryk projektów jest to, iż klient musi zbyt długo czekać na wyniki obliczeń. W dzień lub dwa po zakończeniu liczenia wyniki powinny być przedstawione pracownikom biorącym udział w projekcie oraz sponsorom.

## Co zrobić z wyznaczonymi punktami funkcyjnymi?

Cel liczenia powinien być już ustalony przed liczeniem, gdyż liczenie dla samego liczenia nie ma sensu. Oto kilka zastosowań punktów funkcyjnych:

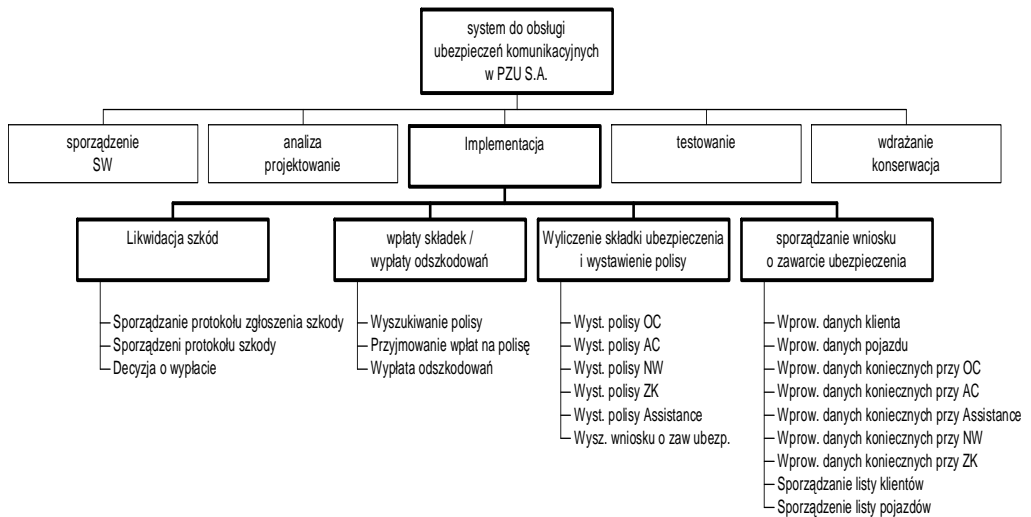
- mierzenie pożądaney produktywności zatrudnionych ludzi, poszukanie outsourcera, a nawet oszacowanie niezbędnego zaangażowania czasowego kierownika projektu, następnie śledzenie zmian w czasie,
- przewidywanie kosztów i budowanie harmonogramu projektu,
- porównywanie produktywności z innymi firmami,
- do oceny czy aplikacja nadaje się do ponownego użytku, powinna zostać odrzucona lub przerobiona.

## Przykład

Oszacowanie kosztów projektu **Systemu do ewidencjonowania polis ubezpieczeniowych w ubezpieczeniach komunikacyjnych dla PZU S.A. – POLISA** metodą punktów funkcyjnych.

Celem projektu jest ewidencjonowanie polis i obliczanie wysokości składek ubezpieczeniowych w ubezpieczeniach komunikacyjnych (OC, NW, AC, Assistance i Zielona Karta) oraz likwidacji szkód komunikacyjnych.

Funkcjonalność systemu POLISA została skrótowo przedstawiona na rysunku 2.4.



Rys. 2.4. WBS dla projektu POLISA (uszczegółowiono tylko fazę implementacji)

Szacowanie wielkości projektu POLISA metodą punktów funkcyjnych dzielimy na następujące etapy:

## 1. Etap wstępny:

- wybieramy typ liczenia – liczymy w fazie projektu (ang. *Development*),
- ekspert systemowy – projektant.

## 2. Obliczenie VAF dla projektu.

W tabeli 2.4 przedstawiono 14 czynników, które szacuje projektant w skali od 1 do 5 jako współczynniki mające istotne znaczenie dla złożoności, wielkości i problemów napotykanym przy budowie oprogramowania.

Tabela 2.4. Cechy aplikacji

Nazwa cechy	Charakterystyka cechy aplikacji	Wartość $C_i$
Wymiana danych	Aplikacja jest oparta na lokalnym przetwarzaniu plików ale możliwe jest zdalne wprowadzanie danych i wykorzystuje zdalną drukarkę.	2
Przetwarzanie danych rozproszonych	Aplikacja przygotowuje dane dla końcowego użytkownika procesu w innym komponencie systemu, takim jak arkusz kalkulacyjny lub system zarządzania bazą danych.	1
Wymagana wydajność systemu	Czas odpowiedzi i wymagania dotyczące przepustowości są wymaganiem kluczowym przez cały czas pracy systemu. Wymagania wydajnościowe dotyczące współpracy mierzonego systemu z innymi systemami są ograniczone.	3
Wymagania sprzętowe systemu	W systemie należy uwzględnić bezpieczeństwo i zagadnienia dotyczące czasu.	2
Częstotliwość transakcji	Wymagania ustanowione przez użytkownika dotyczące przetwarzania transakcji w aplikacji są na tyle duże, że uwzględniane są już w etapie analizy zadań.	4
Wprowadzanie danych w czasie działania systemu	Więcej niż 30% transakcji polega na interaktywnym wprowadzaniu danych.	5
Efektywność dla użytkownika	Aplikacja uwzględnia następujące czynniki: <ul style="list-style-type: none"> <li><input type="checkbox"/> ułatwienia w nawigacji (klawisze funkcyjne, dynamicznie generowane menu, przechodzenie pomiędzy elementami interfejsu za pomocą tabulatora),</li> <li><input type="checkbox"/> menu,</li> <li><input type="checkbox"/> pomoc online i dokumentacja,</li> <li><input type="checkbox"/> automatyczne przesuwanie kursora,</li> <li><input type="checkbox"/> przewijanie okna (w poziomie i w pionie),</li> <li><input type="checkbox"/> zdalne drukowanie,</li> <li><input type="checkbox"/> predefiniowane klawisze funkcyjne,</li> <li><input type="checkbox"/> transakcje online.</li> </ul>	4
Modyfikacja plików logicznych w czasie działania systemu	Możliwa jest aktualizacja większości wewnętrznych plików logicznych.	3
Złożoność przetwarzania	Aplikacja nie zawiera zaawansowanych funkcji matematycznych i logicznych.	0
Ponowne wyko-	Nie ma kodu nadającego się do ponownego użycia.	0



rzystanie pakietów z kodu programu		
Łatwość instalacji	Nie wyspecyfikowano specjalnych wymagań użytkownika oraz nie wymagany jest program ułatwiający instalację.	1
Łatwość administracji	Nie wyspecyfikowano specjalnych wymagań oprócz standardowych procedur archiwizacji.	0
Wielokrotna lokalizacja	Uwzględniono w projekcie potrzebę instalacji aplikacji w więcej niż jednej lokalizacji. Aplikacja jest zaprojektowana tak, by mogła pracować w każdej z tych lokalizacji przy założeniu <i>podobnej</i> konfiguracji sprzętowej i/lub programowej (np. pod kontrolą tego samego systemu operacyjnego).	2
Łatwość dostosowania	Dane kontrolne dotyczące reguł rządzących aplikacją są utrzymywane w tabelach. Zmiany mogą być wprowadzane online przez użytkownika, ale skutki są widoczne natychmiast	2

Suma  $C_i = 29$ 

$$\sum_{i=1}^{14} C_i / 100 = 29 / 100 = 0,29$$

$$VAF = 0,65 + 0,29 = 0,94$$

### Identyfikacja wewnętrznych plików logicznych i zewnętrznych plików interfejsowych

W projekcie wyodrębniono następujące pliki ILF i EIF:

**Wewnętrzne pliki logiczne (ILF):** Klient, Pojazd, Ubezpieczenie OC, Ubezpieczenie AC, Ubezpieczenie NW, Ubezpieczenie Assistance, Ubezpieczenie Zielona Karta, Szkoda, Część.

**Zewnętrzne pliki interfejsowe (EIF):** Wycena

Aby obliczyć liczbę punktów funkcyjnych dla zewnętrznych plików interfejsowych (EIF) i wewnętrznych plików logicznych (ILF), trzeba znać:

- liczbę rekordów tworzących plik (RET),
- liczbę danych (DET) rozróżnialnych dla przyszłego użytkownika tworzących plik.

Odpowiadającą im złożoność (liczba punktów funkcyjnych) odczytujemy z tabeli 2.5 i 2.6.

Tabela 2.5. Liczba punktów funkcyjnych dla ILF

Liczba RET	Liczba DET		
	1–19	20–50	51 lub więcej
1 RET	Niska (7)	Niska (7)	Średnia (10)
2 do 5 RET	Niska (7)	Średnia (10)	Wysoka (15)
6 lub więcej RET	Średnia (10)	Wysoka (15)	Wysoka (15)

Tabela 2.6. Liczba punktów funkcyjnych dla EIF

Liczba RET	Liczba DET		
	1–19	20–50	51 lub więcej
1 RET	Niska (5)	Niska (5)	Średnia (7)
2 do 5 RET	Niska (5)	Średnia (7)	Wysoka (10)
6 lub więcej RET	Średnia (7)	Wysoka (10)	Wysoka (10)

Przykładowy wewnętrzny plik logiczny ILF: – Klient

- liczba RET: 3 (dane osobowe, adres, inne dane),
- liczba DET: 23 (PESEL, imię, nazwisko, ulica itd.),
- złożoność: średnia(10).

Plik interfejsowy – wycena

- liczba RET: 1 (wycena),
- liczba DET: 2 (wartość, marka),
- złożoność: niska (5).

Całkowita liczba punktów funkcyjnych dla danych zapisanych w plikach ILF i EIF:

Tabela 2.7

Wewnętrzny plik logiczny ILF	DT	RET	Złożoność	Punkty funkcyjne (UFP)
Klient	24	3	Średnia(10)	10
Pojazd	21	4	Średnia(10)	10
Ubezpieczenie NW	5	1	Mała(7)	7
Ubezpieczenia AC	55	3	Wysoka(15)	15
Ubezpieczenie OC	12	1	Mała(7)	7
Zielona Karta	15	2	Mała(7)	7
Szkoda	ponad 100	15	Wysoka(15)	15
Części	4	1	Mała(7)	7
Wycena	2	1	Mała (5)	5

Suma punktów funkcyjnych dla plików wynosi **83**

Przystępujemy do identyfikacji transakcji. Aby obliczyć FP dla zewnętrznych wejść (EI), trzeba znać:

- liczbę plików związanych z transakcją (FTR),
- liczbę danych rozróżnialnych dla przyszłego użytkownika wykorzystywanych przez transakcję (DET).

Odpowiadającą im złożoność wyznaczoną w PF odczytujemy z tabeli 2.8.

Tabela 2.8

Liczba plików związanych (FTR)	Liczba DET		
	1–4	5–15	więcej niż 15
Mniej niż 2	Niska (3)	Niska (3)	Średnia (4)
2	Niska (3)	Średnia (4)	Wysoka (6)
Więcej niż 2	Średnia (4)	Wysoka (6)	Wysoka (6)

Aby obliczyć PF dla zewnętrznych wyjść (EO), trzeba znać:

- liczbę plików związanych z transakcją (FTR),
- liczbę danych rozróżnialnych dla przyszłego użytkownika (DET) wykorzystywanych przez transakcję.

Odpowiadającą im złożoność w PF odczytujemy z tabeli 2.9.

Tabela 2.9

Liczba plików związanych (FTR)	Liczba DET		
	1–5	6–19	Więcej niż 19
9	Niska (4)	Niska (4)	Średnia (5)
Mniej niż 2	Niska (4)	Średnia (5)	Wysoka (7)
2 lub 3	Średnia (5)	Wysoka (7)	Wysoka (7)
Więcej niż 3	Średnia (5)	Wysoka (7)	Wysoka (7)

Aby obliczyć PF dla zewnętrznych zapytań (EQ), trzeba znać:

- liczbę plików związanych z transakcją (FTR),
- liczbę danych rozróżnialnych dla przyszłego użytkownika (DET) wykorzystywanych przez transakcję.

Odpowiadającą im złożoność w PF odczytujemy z tabeli 2.10.

Tabela 2.10

Liczba plików związanych (FTR)	Liczba DET		
	1–5	6–19	Więcej niż 19
Mniej niż 2	Niska (3)	Niska (3)	Średnia (4)
2 lub 3	Niska (3)	Średnia (4)	Wysoka (6)
Więcej niż 3	Średnia (4)	Wysoka (6)	Wysoka (6)

Przykładowy moduł „sporządzanie wniosku o zawarcie ubezpieczenia” – tabela 2.11.

Tabela 2.11

Moduł		Sporządzanie wniosku o zawarcie ubezpieczenia				
		FTR	DET	Złożoność	UPF	
Typ Danych	Wejścia EI	Wprowadzenie danych klienta	1	23	Średnia	4
		Wprowadzenie danych pojazdu	2	22	Wysoka	6
		Wprowadzenie danych koniecznych przy OC	3	14	Wysoka	6
		Wprowadzenie danych koniecznych przy AC	3	48	Wysoka	6
		Wprowadzenie danych koniecznych przy Assistance	3	4	Średnia	4
		Wprowadzenie danych koniecznych przy NW	3	4	Średnia	4
		Wprowadzenie danych koniecznych przy ZK	3	4	Średnia	4
	Zapytania EQ	Sporządzanie listy klientów	1	5	Niska	3
	Sporządzanie listy pojazdów	1	2	Niska	3	
Zbiór danych	Klient, Pojazd, Ubezpieczenie NW, Ubezpieczenia AC, Ubezpieczenie OC, Zielona Karta					

Obliczenie sumy nieskorelowanych punktów funkcyjnych (UPF). Sumujemy wartości UPF dla wszystkich transakcji i zbiorów danych (tabela 2.12).

Tabela 2.12

Rodzaj komponentu	Złożoność komponentów			
	Niska	Średnia	Wysoka	Razem
Wejście EI	1 x 3 = 3	4 x 4 = 16	6 x 6 = 36	55
Wyjście EO	0 x 4 = 0	0 x 5 = 0	6 x 7 = 42	42
Zapytania EQ	3 x 3 = 9	1 x 4 = 4	0 x 6 = 0	13
Wewnętrzne pliki logiczne ILF	4 x 7 = 28	2 x 10 = 20	2 x 15 = 30	78
Zewnętrzne pliki interfejsów EIF	1 x 5 = 5	0 x 7 = 0	0 x 10 = 0	5
Całkowita liczba nieskorygowanych punktów funkcyjnych				193

### Wyznaczenie PF

Na podstawie: UPF

- obliczonej liczby = 193,
  - wartości VAF = 0,94.
- otrzymujemy:

$$FP = UPF \cdot VAF,$$

$$PF = 193 \cdot 0,94 = 181.$$

### Szacowanie złożoności implementacji projektu POLISA

Szacowanie linii kodu w zależności od stosowanego języka programowania [1, 26]:

Liczba linii kodu odpowiadająca 1 PF:

Dla języka ADA 95 : 49,

Dla języka C++ : 53,

Dla języka Visual C++ : 34.

Szacowanie liczby linii kodu dla omawianego projektu POLISA:

Dla języka ADA 95 :  $49 \cdot 181 = 8869$ ,

Dla języka C++ :  $53 \cdot 181 = 9593$ ,

Dla języka Visual C++ :  $34 \cdot 181 = 6154$ .

Szacowanie czasu potrzebnego do wytworzenia aplikacji w osobomiesiącach:

VC++ ma poziom języka – 9,5.

Dla języka o takim poziomie jedna osoba jest w stanie przez miesiąc oprogramować od 16 do 23 punktów funkcyjnych.

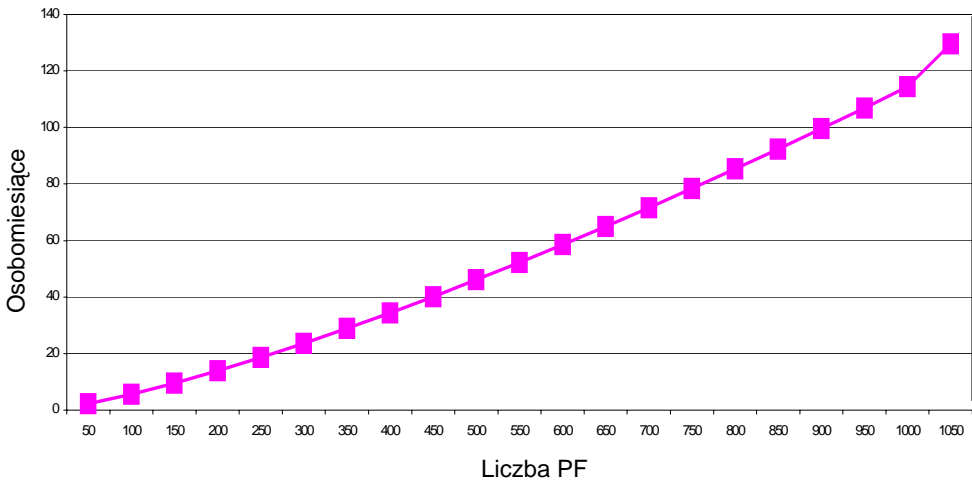
Minimalny czas potrzeby do implementacji projektu

$$181/23 = 7,9 \text{ miesiąca}$$

Maksymalny czas potrzeby do implementacji projektu

$$181/16 = 11,3 \text{ miesiąca}$$

Implementacja projekt POLISA będzie realizowany przez jedną osobę przez 8–11 miesięcy.



Rys. 2.5. Wykres zależności nakładu pracy od liczby PF

Podane szacowanie nie uwzględnia innych czynności technologicznych związanych z wytwarzaniem oprogramowania, jak studium wymagań, projektowanie, analizy, testowania wdrożenia itd.

Z wykresu przedstawionego na rys. 2.5 widać, że wraz ze wzrostem wielkości projektu liczonego liczbą PF, wykonanie kolejnego przyrostu w projekcie (modyfikacje, rozszerzenie funkcjonalności) o np. 100 PF będzie wymagało coraz większej pracochłonności, czyli zmiany w dużych projektach mogą być wielokrotnie kosztowniejsze od zmian w małym projekcie, choć sam produkt liczony w PF jest podobny.

## 2.10. Harmonogram

**Harmonogram** to określony w czasie porządek realizacji zadań w projekcie. Głównymi składowymi harmonogramu są zadania, zależności między nimi, czas trwania oraz alokacja zasobów do poszczególnych zadań.

Jednym z trzech podstawowych parametrów, który definiuje i jednocześnie ogranicza projekt jest czas, któremu w planowaniu projektu i jego monitorowaniu poświęca się szczególną uwagę. Najczęściej mamy do czynienia z sytuacją dysponowania określonymi (najczęściej ograniczonymi) zasobami ludzkimi lub mamy narzucony czas na wykonanie projektu. Charakter projektu i technologia jego realizacji wpływa na związki oraz kolejność realizacji zadań.

### Proces tworzenia harmonogramu

Harmonogram jest to określony w czasie porządek realizacji zadań w projekcie. Głównymi składowymi harmonogramu są zadania, zależności między nimi, czas trwania oraz alokacja zasobów do poszczególnych zadań. Czas trwania realizacji zadania obliczamy według następującego wzoru:

$$\text{czas trwania zadania} = \frac{\text{wymagana praca/nakład pracy zasobu}}{\text{pracochłonność zasobu}}$$

gdzie:

- czas trwania zadania jest rzeczywistą wielkością czasu, który jest planowany na wykonanie zadania (np. 5 dni),
- wymagana praca jest wielkością mierzoną w jednostkach czasochłonności niezbędnej do wykonania zadania (np. 4 osobogodziny),
- nakład pracy zasobu jest wielkością wyrażoną w jednostkach pracochłonności z uwzględnieniem tylko tego czasu, w którym zasób pracuje na rzecz danego zadania – jest alokowany.

### Przykład

- Trzej programiści pracują nad zadaniem przez dwa dni przy nakładzie pracy

8 godzin dziennie, praca każdego zasobu wynosi 16 godzin: (2 dni · 8 godzin).

- Całkowity nakład pracy zasobów wynosi 24 godziny dziennie: (3 programistów · 8 godzin).
- Całkowita praca w zadaniu wynosi 48 godzin: (2 dni · 8 godzin · 3 programistów).
- Czas trwania zadania wynosi 2 dni: 48 godzin / (3 programistów · 8 godzin).

Zrozumienie powyższego wzoru jest ważne do oszacowania, w jaki sposób zmiany dokonywane w zadaniach wpływają na harmonogram projektu.

Prace nad harmonogramem związane są z wykonaniem następujących kroków:

- tworzenie hierarchicznej struktury zadań – WBS,
- specyfikacja zadań na podstawie WBS,
- szeregowanie zadań,
- tworzenie powiązań i zależności między zadaniami,
- określenie wymaganych zasobów,
- szacowanie pracochłonności,
- określenie czasu trwania zadania,
- stworzenie wstępnego harmonogramu projektu,
- stworzenie harmonogramu projektu,
- weryfikacja i korekta harmonogramu.

Patrz: [2, 3, 5, 10, 16, 51].

## Pojęcia i technika tworzenia harmonogramu

Stosuje się najczęściej dwa podejścia co do przyjmowanego czasu trwania zadania:

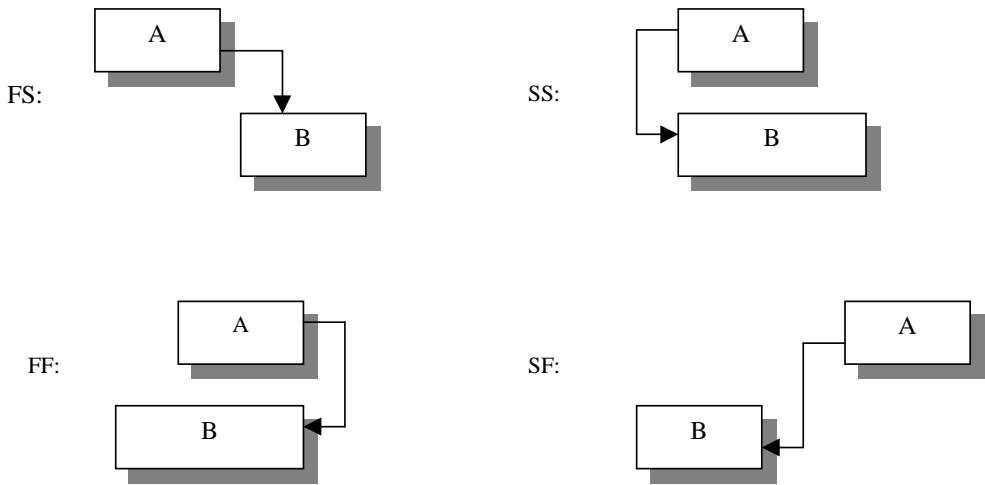
- zależne od posiadanych zasobów (ang. *resource-driven scheduling*),
- o ustalonym czasie minimalnym, w którym zadanie może być wykonane.

Wspomniana technologia realizacji i charakter projektu bezpośrednio wpływa na zależności między zadaniami, które wyspecyfikowano, aby zrealizować projekt.

## Opis powiązań między zadaniami

Graficzne rozmieszczenie zadań na osi czasu oraz ich wzajemne powiązania przedstawia się na ogół w sposób, jak na rys. 2.6.

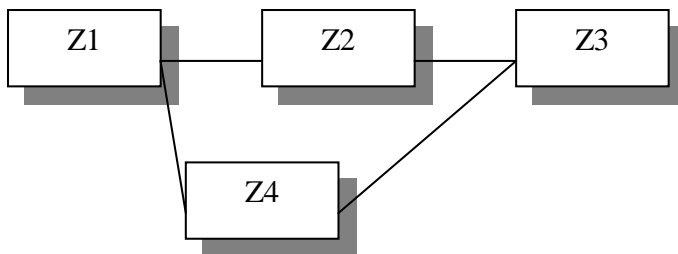
- |  |   |
|--|---|
| Koniec – Start (ang. <i>Finish-to-Start FS</i> )   | – zadanie B nie może rozpocząć się przed ukończeniem zadania A,         |
| Start – Start (ang. <i>Start-to-Start SS</i> )     | – zadanie B nie może rozpocząć się przed rozpoczęciem zadania A,        |
| Koniec – Koniec (ang. <i>Finish-to-Finish FF</i> ) | – zadanie B nie może zakończyć się dopóki nie zakończy się zadanie A,   |
| Start – Koniec (ang. <i>Start-to-Finish SF</i> )   | – zadanie B nie może zakończyć się dopóki nie rozpocznie się zadanie A. |



Rys. 2.6. Typy powiązań między zadaniami w projekcie

Standardowo przyjmuje się, że zadania rozpoczynają się, gdy tylko jest to możliwe. Zadania, których liczba w projekcie zwykle jest znaczna i tworzą harmonogram, mają takie atrybuty, jak:

- sekwencja,
- powiązanie,
- nakładanie się,
- ograniczenia (czasowe), data startu i zakończenia.



Rys. 2.7. Przykład zadań Z1, Z2, Z3, Z4, z których składa się projekt

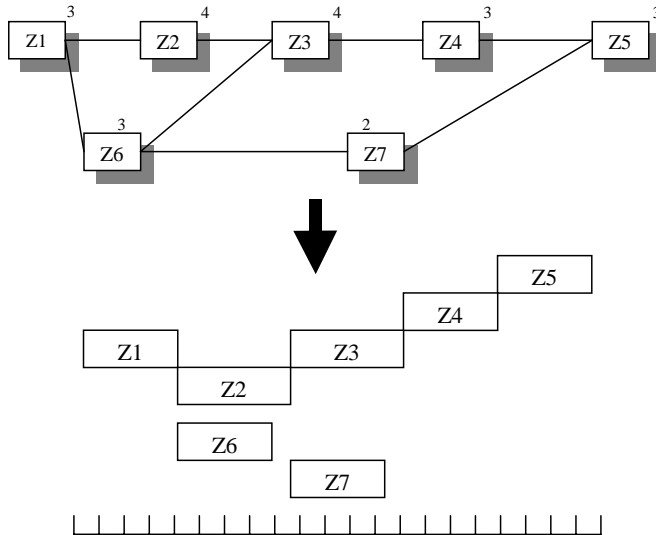
Zadania Z2 i Z3 są realizowane sekwencyjnie po wykonaniu zadania Z1, a zadanie Z3 po zrealizowaniu zadania Z2 i Z4. Z takiego graficznego przedstawienia zadań jak na rys. 2.7 nic nie możemy wnioskować o ograniczeniach czasowych ani o oczekiwanych zasobach przewidzianych do ich realizacji.



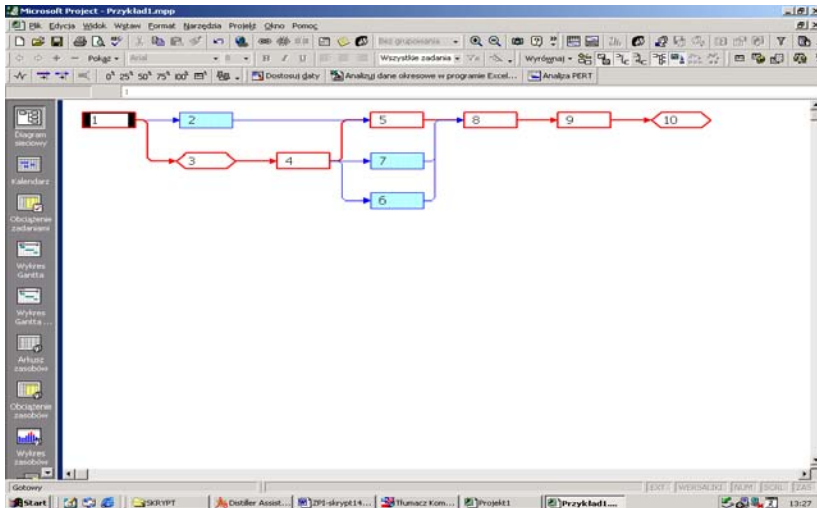
## 2.11. Sieciowe diagramy zależności

**Diagram PERT** (ang. *Program Evaluation and Review Technique* – PERT)

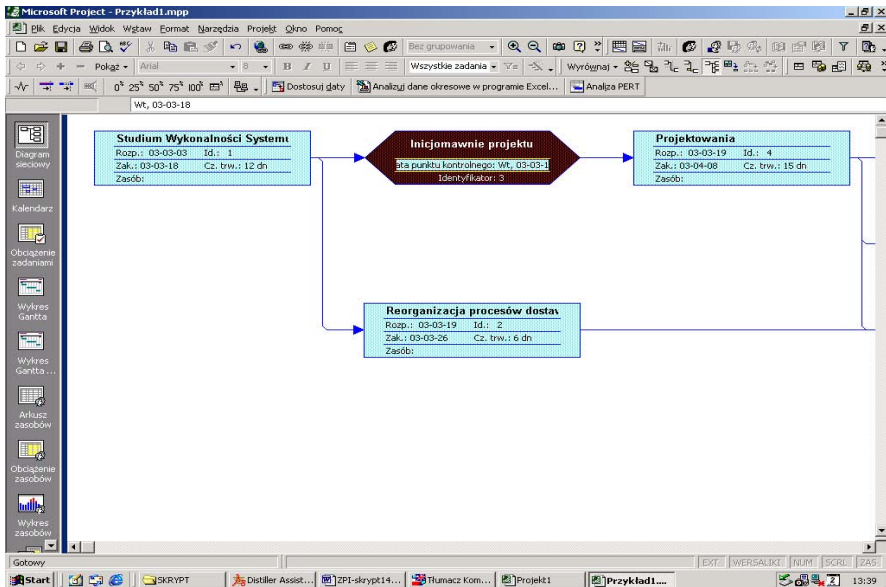
W programie MS Project 2000 możemy przedstawić zadania i ich relacje (poprzednik, następnik oraz które zadania wykonują się równolegle) (rys. 2.8).



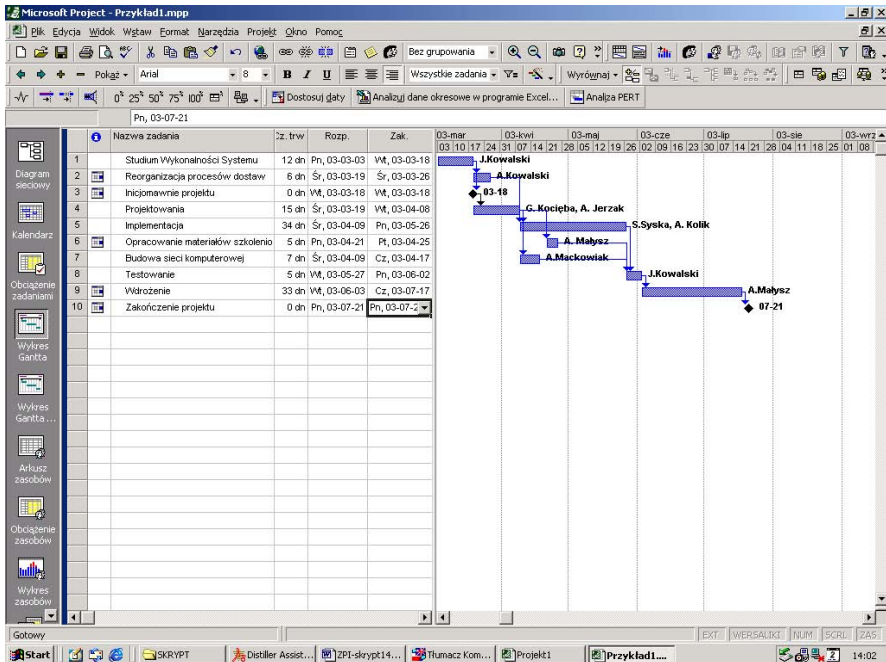
Rys. 2.8. Graficzne przedstawienie zadań oraz ich dekompozycja w sieć powiązań oraz czas realizacji



Rys. 2.9. Widok zadań oraz ich powiązania z wyróżnionymi kamieniami milowymi sieci PERT za pomocą MS Projekt 2000



Rys. 2.10. Fragment projektu z opisami atrybutów sieci PERT za pomocą MS Projekt 2000



Rys. 2.11. Przedstawienie zadań oraz przypisanie zasobów do zadań na schemacie Gantta

Aby zobaczyć nazwy zadań, daty ich rozpoczęcia i zakończenia, czas trwania oraz jakimi zasobami planujemy wykonać zadania, możemy to zobrazować za pomocą diagramu PERT oraz wykresu Gantta (rys. 2.10 i 2.11).

Przypisanie zasobów do zadań polega na oszacowaniu:

Jakie zasoby są potrzebne do realizacji zadania?

Ile jednostek danego zasobu należy przydzielić do zadania?

W jakim czasie od rozpoczęcia zadania zasób będzie potrzebny i do kiedy?

Zasoby = ludzie

Odmiana diagramu, w którym każde zadanie jest opisane przez „punkt startu” i „punkt ukończenia”, jest połączona linią. Innymi liniami zaznaczono zależności między zadaniami.

**Ścieżka krytyczna** (ang. *Critical Path Method* – CPM) – bazuje na PERT i często nazywane jest PERT/CPM. CPM obejmuje ciąg zadań w projekcie, od których zależy zakończenie projektu w terminie. Zadania te wymagają szczególnej uwagi PM, na ogół częstszego monitorowania, niekiedy specjalnego raportowania przez zespół, który realizuje dane zadanie *krytyczne*. Dla tych zadań w zarządzaniu ryzykiem powinno uwzględnić się tzw. akcje zapobiegawcze w przypadku zagrożenia terminu realizacji .

## 2.12. Inicjowanie projektu

Uruchomienie projektu najczęściej następuje w sposób formalny i według przytoczonej listy czynności, które towarzyszą temu zdarzeniu. Jednak biorąc pod uwagę, że nie tylko projekt jest niepowtarzalny, ale również okoliczności i warunki jego uruchomienia, to bywają sytuacje, że prace nad projektem i jego poszczególnymi fazami wyprzedzają oficjalne uruchomienie projektu. Dotyczy to szczególnie sytuacji, kiedy firma uczestniczy w przetargu publicznym i chce zaoferować realizację projektu. Aby móc wycenić wartość projektu trzeba oszacować potrzebne zasoby, czas, koszty stałe, koszty zmienne itp. Należy podjąć wtedy takie działania, które w przypadku wygranej wyprzedzałyby niektóre działania, aby nie podejmować ich powtórnie już w trakcie prac nad projektem. Oczywiście inna jest ich skala i dokładność. Innym podejściem jest szacowanie projektu „dla wygranej” i tu decydują głównie czynniki strategii firmy i jej polityki [23, 24]. Dobrą praktyką jest zatem:

- uzyskanie formalnej aprobaty sponsora,
- przydzielenie budżetu z podziałem na poszczególne fundusze,
- zdefiniowanie struktury projektu (kierownik projektu, odpowiedzialny zwierzchnik, komitet kierujący, reprezentacja użytkownika, struktura zespołu, założenia, odpowiedzialność osób i zespołów),

- ustanowienie struktury (wyznaczenie kierownika i kluczowych członków zespołu, zasad raportowania, monitorowania i powiązań, zasad zarządzania, komunikacji),
- opublikowanie celów i planu,
- opracowanie planów dla wczesnych etapów projektu,
- zapewnienie zasobów, ustanowienie administracji projektu,
- zebranie i odprawa zespołu przewidzianego do realizacji projektu.

### **Efektywna organizacja**

Podobne projekty mogą być różnie realizowane w zależności od otoczenia zewnętrznego oraz modelu organizacji firmy wykonującej projekt (patrz rozdz. 4 oraz [11, 17, 20, 24, 28, 59]). Istnieje jednak przekonanie, że efektywna organizacja prac nad projektem powinna charakteryzować się następującymi kryteriami:

- pojedynczy kierownik (odpowiedzialny, „rozliczalny”, z autorytetem, doświadczony i uzdolniony),
- środki zarządzania dla wsparcia kierownika projektu (możliwość skutecznego oddziaływania na jakość i terminowość prac przez możliwość nagradzania i karania, a także z występowaniem z wnioskami do właściwych osób funkcyjnych – patrz rozdział 4.4 oraz [17, 18],
  - jasno określone cele i odpowiednie ścieżki raportowania,
  - prosta struktura zespołu,
  - krótkie ścieżki komunikowania,
  - samowystarczalne zespoły,
  - zrównoważone połączenie umiejętności i doświadczenia,
  - dedykowane zasoby,
  - jasno zdefiniowane zadania i zakres odpowiedzialności,
  - odpowiedni czas dla komunikowania się i rozwoju zespołu, szkolenia, podnoszenie motywacji i kompetencji (patrz rozdz. 8.4),
  - kontrolowanie delegowania uprawnień (patrz rozdz. 8.7),
  - zasada – najzdolniejszy wykonuje najtrudniejsze zadania,
  - jedna osoba wykonuje jedno zadanie na raz,
  - zdefiniowanie zadań i zakresu odpowiedzialności dla zarządzających jakością [8, 37].

## 3. Śledzenie oraz zarządzanie zmianami projektu

*Śledzenie projektu to nie śledzenie ludzi,  
lecz zadań i produktów, które powstają w czasie projektu.*

KF

### 3.1. Śledzenie projektu – monitorowanie

Sprawdzenie, czy projekt znajduje się pod kontrolą, czy też wymknął się spod kontroli oraz czy w związku z tym trzeba zmienić plan, specyfikację produktów, opis i inne wymagania użytkownika – to czynności permanentne w trakcie trwania projektu. Ogólny model obiegu informacji związanej ze śledzeniem (kontrolą) projektu przedstawiony jest na rys 3.1. Na wyjściu kontrolujemy wyspecyfikowane parametry projektu, sprzężenie zwrotne jest kanałem zgłaszania odstęp od założonych parametrów realizacji projektu oraz postulowanych zmian w projekcie.



Rys. 3.1. Model obiegu informacji związanej ze śledzeniem (kontrolą) projektu

#### Cel raportowania projektu

Zapewnienie bezpiecznego realizowania projektu przy fluktuacji w zespołach projektowych.

Główne aspekty śledzenia projektu:

- czas,
- jakość,

- funkcjonalność,
- koszty.

Etapy śledzenia projektu:

1. Wstępne rozpoznanie:

Porównanie stanu z opisem projektu (ang. *Feasibility Study Report, Business Case*) w celu sprawdzenia, czy nastąpiły (lub też grożą) jakieś istotne zmiany.

2. Drugie rozpoznanie:

Analiza stanu zaawansowanych zadań, porównanie liczby zadań wykonanych w stosunku do planowanych do zakończenia w danym czasie.

3. Trzecie rozpoznanie:

Zebranie danych potrzebnych do finansowania projektu i tworzenia historii projektu. Dane takie pochodzą od codziennych sprawozdań prowadzonych przez członków zespołu. Sprawozdanie powinno zawierać czas spędzony danego dnia na rzecz projektu oraz zakres wykonanych czynności (nakład pracy).

- Projekt powinien być śledzony okresowo: raz na tydzień lub raz na dwa tygodnie.
- Dla projektów o dużym stopniu ryzyka śledzenie należy wykonywać częściej.
- Jeśli widać, że dane zadanie może nie być wykonane na czas, należy wcześniej powiadomić o tym kierownika projektu.
- Szczególnie jest to istotne dla zadań znajdujących się na ścieżce krytycznej. Dla innych zadań jest to istotne wtedy, gdy może zostać przekroczony dopuszczalny czas trwania zadania.
- Śledzenie projektu dotyczy projektu, a nie ludzi! Może być całkiem uzasadnione, iż jednego dnia jakaś osoba nie zrobiła nic na rzecz projektu.

Priorytety śledzenia projektu:

- zadania ścieżki krytycznej,
- zadania nie mające możliwości manewru czasowego,
- zadania o niewielkiej możliwości manewru czasowego,
- zadania o wysokim ryzyku,
- zadania wykorzystujące krytyczne zasoby (ludzi, sprzęt).

## 3.2. Zarządzanie jakością w projekcie

Jest wiele definicji jakości oprogramowania: „zgodność z wymaganiami”, „przydatność użytkowa”. Normy ISO 9000 przyjęły następującą definicję:

*Jakość – ogół cech i właściwości wyrobu lub usług decydujących o zdolności wyrobu lub usługi do zaspokojenia stwierdzonych lub przewidywanych potrzeb.*

Definicja jakości decyduje o całości procesu tworzenia systemu. Podstawowym zadaniem kierownika projektu i innych osób odpowiedzialnych za projekt jest uzyskanie porozumienia co do wspólnej wizji jakości. Bardzo powszechne jest mylenie

jakości z funkcjonalnością produktu. Lider procesu wdrożeniowego musi troszczyć się nie tylko o to, by oprogramowanie miało wszystkie funkcje, których się oczekuje, ale głównie o to, aby te funkcje, które będą dostępne, były niezawodne, efektywne, bezpieczne itd.

Doskonalenie jakości produktu często ogranicza się tylko do polepszania technik i narzędzi testowania. To tradycyjne podejście, bazujące na testowaniu jakości zamiast jej wytwarzaniu i jest przyczyną niepowodzeń wielu projektów.

### Kryteria jakości

Charakterystyka jakości to zespół cech opisujących jakość produktu lub na podstawie których jego jakość jest oceniana.

Jednym z zestawów minimalnego zestawu kryteriów jakości opisujących oprogramowanie jest model McCalla przedstawiony w tabeli 3.1.

Tabela 3.1

<b>Działanie programu</b>	
Wygodny	Odnosi się do efektywności użytkowania programu i wygodnego interfejsu
Bezpieczeństwo	Odnosi się do bezpieczeństwa użytkowania programu pod kątem kontroli uprawnień do korzystania z niego oraz odporności na skutki nieprawidłowej obsługi
Wydajność	Odnosi się do oceny wydajności systemu i sposobów zarządzania zasobami
Poprawność	Odnosi się do stopnia realizacji wymagań, kompletności i logiczności wdrożenia, zgodności działania programu ze specyfikacją
Niezawodność	Odnosi się do stopnia odporności programu na błędy, jego poprawności formalnej oraz sposobów reakcji na błędne sytuacje
<b>Przystosowanie do modyfikacji</b>	
Pielęgnowalność	Ocenia stopień przystosowania programu do działań zmierzających do jego poprawiania, modyfikacji, rozszerzania, adaptowania itp., według nowych wymagań lub raportów o błędach
Elastyczność	Ocenia możliwości rozbudowania programu o nowe funkcje oraz uniwersalność wdrożonych rozwiązań
Testowalność	Ocenia przystosowanie oprogramowania do procesu testowania, tzn. jego strukturę, dokumentację, specyfikację modułów itp., a także przewidziane mechanizmy wspomagające ten proces
<b>Mobilność oprogramowania</b>	
Przenośność	Ocenia oprogramowanie pod kątem zdolności do łatwego uruchomienia na innych maszynach lub systemach programowania niż środowisko projektowe
Uniwersalność	Odnosi się do możliwości wykorzystania istniejącego oprogramowania lub jego fragmentów do konstrukcji innych programów lub systemów komputerowych
Otwartość	Ocenia stopień przystosowania programu do współpracy lub wymiany informacji z innymi systemami komputerowymi

**Zapewnienie jakości.** Według definicji ISO 9000 – zapewnienie jakości są to wszystkie zaplanowane i systematyczne działania, które są niezbędne do uzyskania i utrzymania odpowiedniego stopnia wiarygodności, że wyrób spełni ustalone wymagania jakościowe.

**Planowanie jakości** – jest to zaplanowanie działań zmierzających do zapewnienia jakości. W planie powinny być wzięte pod uwagę następujące kategorie działań:

- przeglądy kontraktów,
- sterowanie analizą wymagań, projektowaniem, wdrożeniem,
- zaopatrzenie i kontrola kooperantów,
- kontrola i badanie oprogramowania w toku produkcji,
- obsługa produktów projektowych niespełniających wymagań,
- instalacje, wdrożenia,
- serwis,
- szkolenie personelu
- wsparcie organizacyjne projektu,
- audyty wewnętrzne i przeglądy systemu jakości inicjowane przez kierownictwo projektu,
- inne działania.

**Plan jakości** powinien zawierać:

- opis sposobu realizacji polityki jakości firmy,
- opis systemu zapewnienia jakości – jego struktury, podziału, odpowiedzialności, procedur i potrzebnych zasobów,
- zestaw przyjętych kryteriów jakości i metryk, służących do ich monitorowania i oceny,
- przyjęte standardy i normy,
- plan działań weryfikacyjnych i walidacyjnych w trakcie projektu,
- plan audytów,
- ustalenie kryteriów jakościowych dla wszystkich produktów,
- plan i procedurę obsługi sytuacji wyjątkowych,
- opis warunków współpracy z klientem, kooperantami, gwarantującą wysoką jakość.

**Nadzorowanie jakości.** Pozytywne, czy negatywne wyniki kontroli jakości są źródłem decyzji projektowych, które zmierzają do:

- dokumentowania działań,
- podjęcia działań korekcyjnych,
- śledzenia ich realizacji,
- weryfikacji ich skuteczności.

**Doskonalenie jakości.** Do podstawowych narzędzi doskonalenia jakości należą:

- inżynieria wymagań,
- metoda projektowania,
- weryfikacja i walidacja,



- przeglądy techniczne oprogramowania,
- testowanie oprogramowania,
- dowodzenie poprawności,
- symulacje i prototypowanie,
- śledzenie wymagań,
- inne narzędzia.

Do metod sformalizowanych należy również tzw. formalny przegląd techniczny (ang. *Formal Technical Report* – FTR): metoda ustrukturalizowanego działania, podczas którego osoba lub grupa osób poprawia jakość oryginalnego produktu pewnej pracy, jak również jakość samej metody [16, 19].

FTR zapewnia:

- autorowi – dane o usterkach,
- partnerom i konstruktorom – informacje o produkcji,
- testującym – informacje o prawdopodobnych błędach,
- kierownictwu – informację o stanie produktu,
- grupie nadzorowania jakości – informację o stanie procesu.

Aby uczynić punkt rozważań o zarządzaniu jakością mniej abstrakcyjnym i nie usłyszeć komentarza po przeczytaniu *Kto by to wszystko miał robić*, mała uwaga dotycząca tego kiedy i czy wymienione procedury wprowadzać. Trudno o jednoznaczną odpowiedź. Jedno jest chyba pewne, nie warto i chyba się to nie uda, aby wszystko wprowadzać przy nowym projekcie, jeśli dotychczas zespół–organizacja ich nie stosowała. Formalne procesy są dobrym „wynalazkiem”, jeśli się je stosuje ze zrozumieniem, jeśli je stosowano wcześniej oraz najlepiej, jeśli jest ich aprobata. Ale projekt jest próbą zrobienia czegoś, czego jeszcze nigdy nie zrobiono, zatem sposób jego realizacji też może być nowy, kiedyś można i trzeba zacząć wprowadzać procedury zarządzania jakością.

**Nadzorowanie funkcjonalności** dotyczy realizacji celów, jakie przyjęto dla projektu i były przedmiotem specyfikacji funkcji, które mają być realizowane przez system informatyczny w zakresie interfejsu, wydajności, dostępu itd. W wyniku nadzorowania projektu możemy dostrzec uchybienia, których źródło ma pochodzenie:

- wewnętrzne – zmiany wywodzące się z fazy planowania projektu, wynikające z niezrozumienia założeń, ze zmian w oszacowaniu nakładu pracy, zmian w składzie zespołu, z przyczyn technicznych i inne zmiany, których nie można było przewidzieć wcześniej;
- zewnętrzne – pochodzące od klienta lub kierownictwa, wynikające z nowych idei, przeoczeń, wymagań innych projektów i inne zmiany, które nie są związane z początkową specyfikacją projektu.

**Nadzorowanie kosztów.** Śledzenie i nadzorowanie czasu oraz kosztów projektu

przedstawiono w rozdziale 3.4.

### **Kontrolowanie zmian**

1. Żądanie zmiany: musi być dokonane na piśmie. Adresowane do kierownika projektu. Musi zawierać: nazwisko członka zespołu żądającego zmiany, datę, opis problemu, opis proponowanej zmiany i jej uzasadnienie.

2. Ocena postulowanej zmiany:

Czy zmiana jest rzeczywiście uzasadniona? Jeśli tak, to czy musi być wykonana teraz, czy można ją odłożyć na później?

Czy w istotny sposób zmienia opis projektu?

Na jakie zadania (zakończone lub w toku) wpływa ta zmiana?

Jaki jest nakład pracy potrzebny na jej zrealizowanie, koszty i korzyści?

Czy w konsekwencji jej wprowadzenia należy ustalić nowy harmonogram projektu?

Czy wymaga ona nowych zasobów nie przewidzianych w planie projektu?

Czy zmiana zwiększa w istotny sposób złożoność i ryzyko projektu?

Czym ryzykujemy, jeśli zrealizujemy zmianę (lub: jeśli nie)?

Jakie są priorytety przypisane do zmiany?

Jakie są skutki zmiany na innych procesach ?

Jakie będą skutki technicznej stabilności produktu?

3. Decyzja:

Jeśli kierownik projektu nie ma wątpliwości co do konieczności wprowadzenia zmiany i jeśli zmiana:

- może być dokonana w danym momencie realizacji projektu,
- nie wymaga dodatkowych środków,
- nie zmienia ryzyka i złożoności projektu,
- nie zmienia opisu projektu,
- nie przedłuża realizacji projektu.

Kierownik projektu akceptuje zmianę. W przeciwnym razie należy odwołać się do kierownictwa organizacji, komitetu sterującego lub innej struktury kompetentnej do podejmowania decyzji. Decyzja powinna być na piśmie zakomunikowana zgłaszającemu zmianę i traktowana jako ostateczna [3, 7, 16, 36].

### **Sprawozdawczość – raportowanie**

Szablony i zasady oraz obieg dokumentów sprawozdawczych jest zadaniem PM i to należy ustalić na samym początku realizacji projektu, ponieważ:

- sprawozdawczość to więcej niż „wielkie raporty na temat małych sukcesów”,
- problemy należy sygnalizować wcześniej,
- zgłoszenie problemu musi pociągać za sobą odpowiednią reakcję,
- sygnalizowanie problemów jest elementem kultury technologicznej zespo-

tu/organizacji.

Tabela 3.2. Klasy i metody przeglądów

Metoda	Typowe cele	Typowe cechy
Walkthroughs	Minimalny nakład Ćwiczenie dla konstruktora Szybki obieg	Brak przygotowań Nieformalne Nie ma pomiarów Nie jest FTR!
Przeglądy techniczne (ang. <i>technicreviews</i> )	Identyfikacja wymagań Wychwycenie niespójności Ćwiczenie	Proces sformalizowany Prezentacje autorskie Szeroki zakres dyskusji
Inspekcje (ang. <i>inspections</i> )	Efektywne wykrycie i usunięcie wszystkich defektów	Proces sformalizowany Listy kontrole Pomiary Faza weryfikacji

## Raportowanie

Sposób raportowania postępu projektu zależy od organizacji. Zakres informacji zawartych w raporcie dla kierownictwa organizacji obejmuje:

- stan projektu: planowo czy nie?
- jeśli nie, jakie są przyczyny zmian?
- jakie dodatkowe akcje podjął zespół w celu przewyżczenia problemów?
- czy są jakieś alternatywy w dalszej realizacji projektu?
- jak może pomóc kierownictwo organizacji?

Raport powinien zawierać także wymaganą dokumentację projektu, uwzględniającą aspekty finansowe (fundusze już wydatkowane a fundusze planowane, wystawione faktury dla klientów itd.).

**Akcje naprawcze** – działania, które zapobiegają negatywnym następstwom niektórych „przeoczeń” lub błędów popełnionych we wcześniejszych fazach projektu. Należą do nich:

- Detekcja potrzeby akcji naprawczej.
- Wybór właściwej akcji naprawczej.
- Wczesne podjęcie akcji naprawczej.

Nadzorowanie pracy i pisanie raportów na temat jej postępów to za mało! Menedżer projektu musi wносить nową wartość przez wczesne identyfikowanie problemu i podejmowanie akcji naprawczych.

### 3.3. Źródła i rodzaje zmian

*Errare humanum est.*

Seneka

Zarządzanie zmianami, nazywane również zarządzaniem konfiguracją projektu, obejmuje zasady i techniki zmierzające do identyfikacji, śledzenia, oceny, sterowania i autoryzacji zmian we wszelkiej informacji projektowej, która ma być udostępniona różnym osobom (zespołom), związanym z projektem. Głównym celem zarządzania konfiguracją jest kontrolowane wprowadzenie do projektu zmian dotyczących dokumentacji, kodu programu i innych produktów faz projektowych. Sposób ich dokonania nie może mieć negatywnego wpływu na zakładane parametry projektu oraz integralność i jakość wytwarzanego systemu informatycznego. Pojęcie zarządzanie konfiguracją jest tłumaczeniem angielskiego terminu *Configuration Management*. W odniesieniu do systemów informatycznych słowo konfiguracja rozumiemy jako zmienny w czasie zestaw wszelkich produktów projektu i innych informacji, które są istotne do sprawnej jego realizacji.

W przypadku zarządzania konfiguracją koncentrujemy się na tych aspektach systemu informatycznego, które bezpośrednio lub pośrednio dotyczą gromadzenia informacji, przechowywania, aktualizowania i dostarczania odbiorcom informacji w taki sposób, aby zachować integralność informacji, jej dostępność, poprawność, wiarygodność i aktualność.

**Elementy konfiguracji.** Należą do nich przede wszystkim:

- Dokumentacja produktów – wszystkie informacje ujęte w formie dokumentacji, które są podstawą do projektowania.
- Dokumentacja projektowa – dokumenty konstytuujące projekt i opisujące jego historię oraz stan bieżący.
- Standardy, procedury, instrukcje – wszelkie ujęte w formie normatywnej opisy sposobów realizacji procesów projektowych.
- Kod programu.

Zwykle zaleca się stworzenie zespołu ds. zarządzania konfiguracją projektu, który najczęściej jest rozproszony po całym projekcie.

Na każdym poziomie znajduje się osoba, która ma prawo zatwierdzać zmiany w projekcie, stosownie do swoich kompetencji i zakresu prac, za które odpowiada. Osoba ta ma obowiązek dopilnowania, aby wszelkie zmiany były odpowiednio raportowane i archiwizowane, a te, które zostały przyjęte do realizacji, także monitorowane, aż do momentu ich wprowadzenia.

**Zgłoszenia niezgodności:**

- Błędy w wymaganiach, wynikające ze złego rozpoznania wymagań.

- Błędy w projekcie.
- Naruszenie standardów.

**Zgłoszenie zmian.** Są obsługiwane podobnie jak niezgodności, lecz ich powstanie ma inną naturę. Można je podzielić na trzy kategorie:

Wymagania nie realizowalne – np. z przyczyn zbyt małych zasobów bądź błędów w implementacji innych wymagań.

Rozszerzenia – zwykle dotyczą nowych wymagań, powstających jako wynik przemyśleń i analizy twórców i odbiorców przyszłego systemu.

**Usprawnienia.** Każde zgłoszenie powinno być zapisane, a jego status odpowiednio śledzony.

Przed podjęciem decyzji o wprowadzeniu zmian do projektu należy przeanalizować:

- rozmiar i zakres zmian,
- złożoność,
- ograniczenia czasowe,
- wpływ na bieżący stan projektu,
- zasoby wymagane do realizacji,
- koszt,
- ryzyko niepowodzenia,
- politykę firmy, produktu, projektu,
- wymagania udziałowców,
- alternatywne sposoby wprowadzenia zmian.

Jednym z postulatów dobrego zarządzania konfiguracją jest zasada aktualności danych projektowych. Oznacza ona, że w danej chwili można otrzymać pełny obraz zmian w projekcie, dotrzeć do źródła ich powstania i prześledzić losy ich realizacji.

### 3.4. Proces zarządzania zmianami

Zarządzanie zmianami (konfiguracją) to zasady i techniki zmierzające do identyfikacji, śledzenia, oceny, sterowania i autoryzacji wprowadzonych do projektu zmian we wszelkiej informacji projektowej, która ma być udostępniona różnym osobom, związanym z realizacją projektu. Wprowadzenie zarządzania zmianami ma na celu zachowanie spójności i integralności projektu oraz zabezpieczenie osiągnięcia zakładanej jakości [32, 44, 45].

Zadania związane z zarządzaniem zmianami w większych projektach powierza się najczęściej zespołowi ds. zarządzania zmianami, który może być rozproszony po całym projekcie.

**Zmiana** – propozycja modyfikacji czegoś, co zostało już wcześniej uzgodnione

może dotyczyć zarówno projektu, jak i sposobu jego zarządzania.

Zarządzanie zmianami – proces, który umożliwia wprowadzenie wymaganych i uzgodnionych zmian przy minimalnym wpływie na projekt.

#### **Zmiany w projekcie:**

- zmiany modyfikują ustalony (bazowy) plan projektu,
- zmiany mogą (lub nie) modyfikować uzgodnione elementy projektu,
- zmiany źródeł,
- zespołu realizującego projekt,
- użytkowników.

Zmiany nie mogą być ignorowane! Zmiany muszą być formalnie zarządzane!

#### **Ignorowanie a bezkrytycyzm**

Ignorowanie (odmowa wprowadzenia) jakichkolwiek zmian spowoduje, że projekt nie będzie spełniał rzeczywistych potrzeb. Bezkrytyczna akceptacja wszystkich zmian grozi wymknięciem się projektu spod kontroli.

#### **Dlaczego zarządzanie zmianami?**

Ponieważ w każdym projekcie w trakcie jego realizacji występują zmiany:

- Potrzeba oceny zmiany: konieczna czy nie?
- Zespół projektowy musi pracować według takich samych założeń, co do zakresu i produktów projektu

#### **Zarządzanie zmianami – cele:**

- racjonalizacja procesu modyfikacji projektu,
- kwalifikacja zmian: konieczne, warunkowe itd.,
- wprowadzenie zmian do projektu w sposób jak najbardziej łagodny, „bezwstrząsowy”,
- definiowanie i dokumentowanie zmian (zakresu, czasu, kosztu, ryzyka...),
- uzasadnienie konieczności wprowadzenia zmiany.

#### **Brak zarządzania zmianami**

Po zastosowaniu zmian w projekcie bez zarządzania powoduje:

- poślizgi czasowe, będące wynikiem dodatkowej pracy,
- opóźnione punkty węzłowe i terminy realizacji prac,

- oddalanie się od założonych celów, a nawet bankructwo projektu.

### Zarządzanie zmianami – struktura procesu

- Identyfikacja konieczności zmian: dokumentacja żądania zmiany (opis, uzasadnienie, istotność, wpływ).
- Analiza: ocena wpływu zmiany na poszczególne podprojekty.
- Ocena wpływu na projekt (cały).
- Decyzja o wprowadzeniu (tak, nie, na później).
- Wprowadzenie, komunikacja, rozpowszechnienie.

Tabela 3.3

Decyzja/zmiana	W zakresie projektu	Poza zakresem projektu
Odrzucenie	Rozpowszechnienie decyzji wraz z uzasadnieniem	
Akceptacja	<ul style="list-style-type: none"> <li>• dołączenie do planu</li> <li>• zmiana harmonogramu</li> <li>• przydział środków</li> <li>• rozpowszechnianie i realizacja</li> </ul>	<ul style="list-style-type: none"> <li>• przygotowanie propozycji</li> <li>• wstrzymanie realizacji do momentu uzyskania formalnej kontaktowej zgody</li> </ul>
Odłożenie	<ul style="list-style-type: none"> <li>• przeprowadzenie dokładniejszej analizy</li> <li>• zgrupowanie zmiany z innymi o podobnym charakterze</li> </ul>	

### Źródła zmian:

a) wewnętrzne:

- architektura systemu,
- plan realizacji,
- konfiguracja (nie powoduje zmian zakresu projektu);

b) zewnętrzne:

- funkcjonalność,
- otoczenia,
- misji,
- prawne (zwykle powodują zmianę zakresu projektu).

**Grupowanie zmian ma na celu** polepszenie efektywności procesu wdrażania zmian i może być uzyskane przez ich analizę:

- według podobieństwa przyczyn,
- według podobieństwa działań,
- w celu minimalizacji zakresu powtarzania prac,
- w celu minimalizowania zakresu testowania.

Łączenie zmian tak, aby ich wprowadzenie stawało się przedmiotem projektów z wyznaczonymi celami oraz mierzalnymi efektami ich wprowadzenia.

Przykładowy formularz zgłoszenia zmiany, którą autor stosował w fazie nadzoru autorskiego *do zarządzania zmianami projektu*.

Przykład: Formularz zgłoszenia zmian w projekcie

### FORMULARZ – ZGŁOSZENIE ZMIANY

Tytuł Projektu    Serwis Internetowy *Rejestru Zakładów Pracy Chronionej*

Skrót:            *SI RPC*

Numer zmiany:    **39/2003**

Akceptacja oczekiwana w : \*1 tydzień  / 2 tygodnie  / 1 miesiąc  / 3 miesiące

#### Część I : Propozycja zmiany (wypełnia wnioskujący)

##### APLIKACJA Centralna – Zmiana kodów teryt

Propozycja zmiany:

1. *Dokonanie zmian kodów terytorialnych w bazie centralnej zgodnie z załącznikiem*
2. *Wprowadzenie takich zmian w aplikacji centralnej , które pozwolą na automatyczną zmianę „starego” kodu na „nowy” niezależnie, czy operator w aplikacji lokalnej wprowadził zmianę „starego” kodu na „nowy” czy też nie.*

Uzasadnienie zmiany:

*W dniu 1.01.2003 weszła w życie zmiana rozporządzenia Rady Ministrów w sprawie kodów terytorialnych. Ograniczenie się do zmian w aplikacji jedynie do zmian słownikowych spowoduje, że w raportach odnoszących się do poszczególnych powiatów nie będzie prawidłowych danych. Przykładowo w nowo utworzonym powiecie Powiat m.st. Warszawa raporty wykażą 0 (zero) zakładów pracy chronionej.*

Zgłaszający zmianę:

**A. Kowalski**

Data: 20.02.2003

#### Część II – Decyzja

(wypełnia „akceptujący”)

Akceptacja*	<input type="checkbox"/>	Podpis decydującego: Kazimierz Frączkowski  Data: 23.02.2003
Akceptacja warunkowa*	<input checked="" type="checkbox"/>	
Odrzucenie zmiany*		



**Komentarz :**

Akceptacja częściowa

\* zaznacz właściwe pole X

**Część III. Podsumowanie działań**

(wypełnia Kierownik Projektu)

**Koszty:**

3 godz. pracy

**Harmonogram:** w terminie dogodnym dla Zamawiającego z powiadomieniem wybranych organów rejestrowych**Zasoby:** J. Nowak, A. Zawiślak**Efekt w projekcie**

Ad1.

## 1. W CBD

– zmienić słownik teryt (mogą to wykonać: A. Kowalski J. Nowak, A. Kabel)

– wystawić nowy słownik TERYT do pobrania (data w SL) dla dolnośląskiego, mazowieckiego, MZ i podkarpackiego

2. W aplikacji lokalnej nie dokonujemy zmian.

Ad 2. Zmiana 39/2003 p.2 proponuje się odrzucić, nie można „ręcznie” manipulować danymi, które autoryzują lokalne punkty rejestrowe z użyciem podpisu elektronicznego. Zmodyfikowanie BD aplikacji centralnej przez Wykonawcę spowoduje, że powinniśmy też zmodyfikować bazy lokalne. Jest to rozwiązanie tymczasowe na które nie można się zgodzić. Podmieniać kody może również aplikacja lokalna – ale na takie rozwiązanie też nie możemy zgodzić się.

Co zrobimy jeśli przyjdą kolejne zmiany w kodach teryt?

WNIOSEK

Punkt 2 zmiany 39 odrzucamy.

**Ryzyko:**

Gdy FIRMA S.A. będzie modyfikowała „ręcznie” dane w CBD, będzie przejmowała odpowiedzialność za ich jakość.

Zmiana : \*TAK  /NIE

**Komentarz:**

Aplikacje lokalne w ramach aktualnej funkcjonalności mogą poprzez złożenie wniosku o aktualizację – dokonać stosownych zmian kodu.

Działania w zakresie zmian oszacowane przez.....W. Warkomla.....

Data: .....22.02.2003.....

\* – zaznacz X właściwe pole

Załącznik do formularza zgłoszenia zmian

L.p.	Przed zmianą w bazie centralnej		Po zmianie w bazie centralnej	
	Kod	nazwa	Kod	nazwa
1	0263	Powiat M. Wałbrzych	0221	Powiat Wałbrzyski
2	0263011	M. Wałbrzych	0221091	Wałbrzych
3	1805102	Szerzyny	1216162	Szerzyny
4	1406102	Tarczyn	1418063	Tarczyn
5	1431191	Sulejówek	1412151	Sulejówek
6	1413	Powiat warszawski	1465	Powiat m.st.Warszawa
7	1431011	Warszawa-Bemowo	1465028	Bemowo
8	1431021	Warszawa - Białoleka	1465038	Białoleka
9	1431031	Warszawa - Bielany	1465048	Bielany
10	1431041	Warszawa - Centrum	1465	Powiat m.st.Warszawa
11	1431058	Warszawa Mokotów	1465058	Mokotów
12	1431058	Warszawa - Ochota	1465068	Ochota
13	1431078	Warszawa Praga Południe	1465078	Praga Południe
14	1431088	Warszawa Praga Północ	1465088	Praga Północ
15	1431098	Warszawa Śródmieście	1465098	Śródmieście
16	1431108	Warszawa Wola	1465188	Wola
17	1431118	Warszawa Żoliborz	1465198	Żoliborz
18	1431121	Warszawa Rembertów	1465098	Rembertów
19	1431131	Warszawa Targówek	1465118	Targówek
20	1431141	Warszawa Ursus	1465128	Ursus
21	1431151	Warszawa Ursynów	1465138	Ursynów
22	1431161	Warszawa Wawer	1465148	Wawer
23	1431171	Warszawa Wilanów	1465168	Wilanów
24	1431181	Warszawa Włochy	1465178	Włochy
25	1431201	Wesoła	1465158	Wesoła

*Dla większości projektów nie ma czegoś takiego jak „mała zmiana”!*

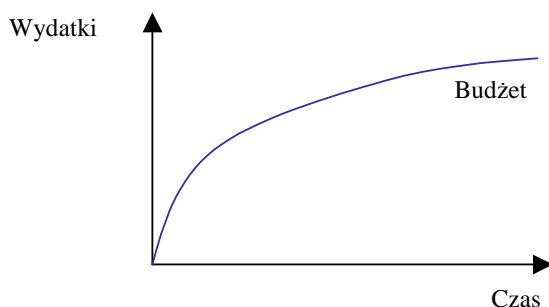
KF

### 3.5. Śledzenie i nadzorowanie czasu oraz kosztów projektu

Konieczność kontroli realizacji projektu zauważono już dawno (patrz rozdz. 1.3 – C. Bernard), wskazując na potrzebę „skontrolowania osiągniętych wyników i porównanie z zamierzonym celem – wyciągnięcie wniosków na przyszłość (do następnego planu projektu)”.

Potrzebna jest ona zarówno osobom odpowiedzialnym za realizację projektu, jak i zleceniodawcom. Choć obydwie wspomniane grupy na realizację projektu patrzą z różnych perspektyw, to jednak można wyciągnąć pewną część wspólną lub próbować opisać stan projektu obiektywnymi współczynnikami. Opisywane metody opierają się właśnie na takich współczynnikach dających pełny i stosunkowo obiektywny obraz.

Źródłem powstania najpopularniejszej metody są Stany Zjednoczone, a więc jak można się spodziewać podstawą obliczeń są pieniądze. Budżet przedsięwzięcia szacowany jest metodą *bottom-up*, tzn. budżet przydzielany jest każdemu małemu zadaniu realizowanemu w ramach projektu. Całkowity budżet jest więc sumą budżetów wszystkich zadań. Harmonogram przedstawia zatem rozkład w czasie planowanych wydatków na realizację zadań, czyli krzywą budżetu w funkcji czasu.



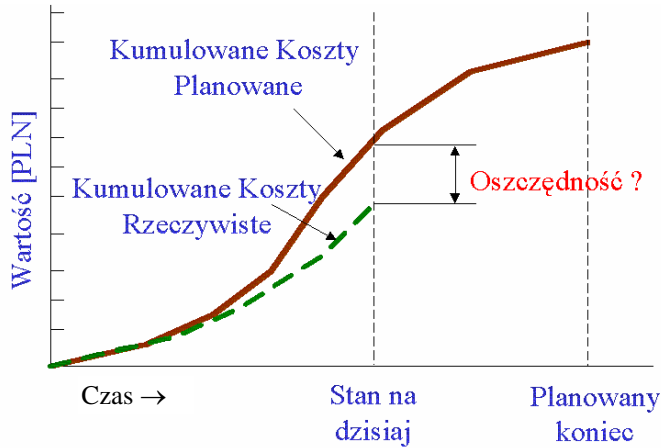
Rys. 3.2. Przebieg wydatków na realizację projektu w czasie

Przykładową krzywą budżetu przedstawia wykres na rysunku 3.2. Informuje nas ona o planowanych wydatkach rozłożonych w czasie. Aby zweryfikować wydajność osiąganą przy realizacji projektu, należy porównać osiągnięte wyniki z tym co zostało zaplanowane.

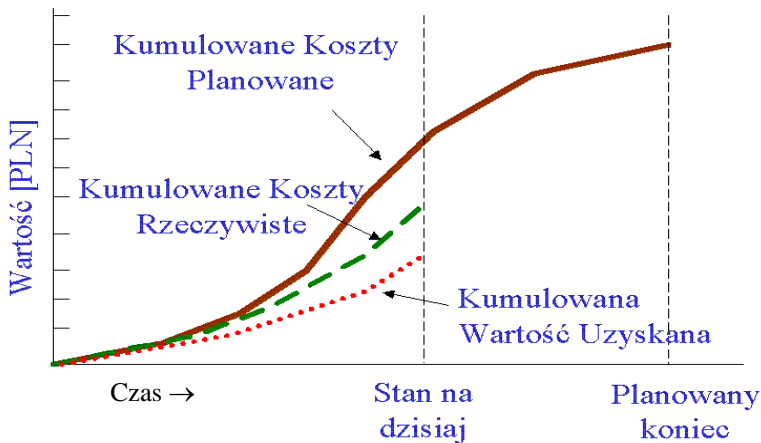
### 3.6. Nadzorowanie projektu metodą wartości uzyskanej (*Earned Value*)

Amerykańscy producenci kierowani potrzebą pomiaru wydajności pracy zastosowali metodę, która stała się początkiem *Earned Value*, czyli obliczania tzw. *wartości uzyskanej*. Prawdziwy początek i rozkwit metody to lata 1960. Departament Obrony Narodowej Stanów Zjednoczonych w 1967 roku przyjął tę metodę jako standard stosowany do pomiaru wydajności projektów prowadzonych na ich zlecenie. Metoda *Earned Value* lub wartości uzyskanej polega na kontroli realizacji projektu przez porównywanie zrealizowanego do danej chwili zakresu prac i terminów ich realizacji oraz rzeczywiście poniesionych kosztów z przyjętymi w planie bazowym harmono-

gramem i budżetem projektu. Metoda ta należy do kategorii metod zarządzania przez pomiar wydajności (ang. *performance based management*).



Rys. 3.3. Tradycyjne podejście do kontroli kosztów



Rys. 3.4. Przedstawienie na wykresie dodatkowej krzywej wartości uzyskanej

W tradycyjnej metodzie kontroli realizacji projektu w punkcie kontrolnym (stan na dzisiaj) porównywano, jakie były faktycznie poniesione wydatki związane z realizacją projektu (od jego początku – kumulowane) w stosunku do zaplanowanych kosztów na realizację (od jego początku – kumulowane). W takim podejściu nie wyceniano wartości prac, które zostały wykonane do chwili kontroli. To czy poniesione nakłady spowodowały wytworzenie określonej wartości, jaką *wartość uzyskana* nie było przedmiotem kontroli. W ten sposób można było mylnie interpretować różnicę między

kumulowanymi kosztami planowanymi i kumulowanymi kosztami rzeczywistymi jako oszczędnościami (rys. 3.3).

Kontrola obejmuje również wycenę wartości prac (produktów, usług itd. od początku realizacji projektu) zrealizowanych do chwili kontroli (stan na dziś) przez obliczenia tzw. kumulowanej wartości uzyskanej, wówczas mamy możliwość porównania w pieniądzech trzech parametrów, tj. kosztów planowanych (KP), kosztów rzeczywistych (KR), wartości uzyskanej (WU) (rys. 3.4) i określenia wskaźników odchyień związanych z realizacją projektu takich, jak planowany czas (harmonogram) i budżet (koszt projektu).

### Odchylenia

Odchylenie kosztów (ang. *cost variance*) jest pierwszym parametrem metody *Earned Value*:

$$OdK = WU - KR$$

gdzie:

- OdK* – odchylenie kosztów,
- WU* – wartość uzyskana,
- KR* – koszty rzeczywiste.

Odchylenie od harmonogramu (ang. *schedule variance*) jest różnicą w danej chwili między planowanym kosztem a wartością uzyskaną:

$$OdH = WU - KP$$

gdzie:

- OdH* – odchylenie od harmonogramu,
- WU* – wartość uzyskana,
- KP* – koszt planowany.

Jak widać z definicji danych odchyień, tj. *OdK* i *OdH* mogą one mieć wartości dodatnie lub ujemne. Wartości dodatnie występują wtedy, kiedy *WU* jest większa zarówno od *KR*, jak i *KP*, jest to sytuacja korzystna, bo inwestując w projekt np. 1 zł wypracowujemy *WU* większą od 1 zł. Również nasz harmonogram nie ma opóźnienia do chwili kontroli, ponieważ uzyskaliśmy większą wartość (więcej zrobiliśmy) niż było zaplanowane. Oczywiście w przypadku ujemnych wartości podanych parametrów należy wysunąć wniosek, że projekt realizujemy za drogo oraz mamy opóźnienie w harmonogramie.

### Wskaźniki

Wskaźnik kosztów (ang. *cost performance index*) daje nam odpowiedź na pytanie

ile uzyskujemy za każdą wydaną jednostkę waluty finansowania projektu, np. złotówkę. Jeśli np. uzyskujemy wartość prac tylko 90 groszy przy wydaniu jednego złotego, to znaczy, że mamy małą wydajność prac w projekcie:

$$WK = WU/KR,$$

gdzie:

$WK$  – wskaźnik kosztów,

$WU$  – wartość uzyskana,

$KR$  – koszt rzeczywisty.

Wskaźnik harmonogramu (ang. *schedule performance index*) jest porównaniem tego co zostało zrobione do tego co miało być zrobione. Jeśli wskaźnik ten wynosi np. 1,05, to znaczy że projekt realizowany się szybciej niż było planowane. Jeśli jednak  $WH$  jest mniejszy od jednego, to znaczy, że się spóźniamy:

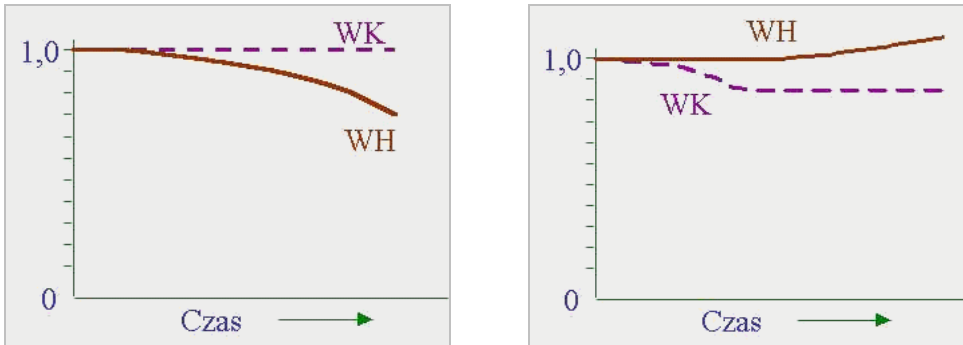
$$WH = WU/KP$$

gdzie:

$WH$  – wskaźnik harmonogramu,

$WU$  – wartość uzyskana,

$KP$  – koszt planowany.



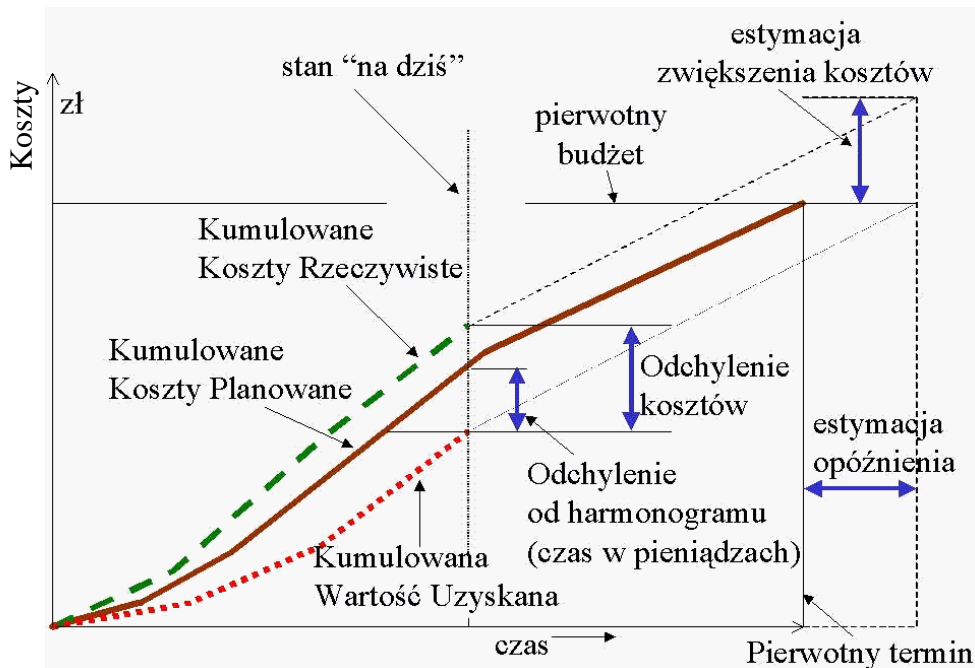
Rys. 3.5. Przedstawienie graficzne  $WK$  i  $WH$  w metodzie *Earned Value*

Analiza i wnioski z danych wskaźników nie zawsze są takie jednoznaczne i jasne, ponieważ mogą dotyczyć nie tylko konieczności zwiększenia wydajności pracy zespołu, ale może się okazać, że wartość prac w projekcie została zaniżona (przydzielenie wartości uzyskanej) lub planowane koszty oszacowano nieprawidłowo.

Gdy mamy możliwość okresowego badania tendencji przez estymację zwiększenia kosztów projektu czy możliwych opóźnień, wówczas PM można wcześniej podjąć akcje naprawcze lub wystąpić z inicjatywą negocjacji terminu zakończenia projektu oraz jego kosztów (np. aneks do umowy) (rys. 3.5).

Z przeprowadzonych badań nad 700 projektami śledzonymi metodą *Earned Value*

wynika, że już po około 15% realizacji widać pewną stabilizację trendów. Odwrócenie trendów (tych negatywnych) jest jednym z najtrudniejszych wyzwań *Project Managementa* [5, 53, 59].



Rys. 3.6. Przedstawienie ostatecznych kosztów projektu i rzeczywistego (szacowanego) terminu zakończenia

## Przydzielanie wartości uzyskanej

### Metoda 0/100

Dogodnym momentem kontroli prac projektowych metodą *Earned Value* są takie fazy harmonogramu projektu, w których zakończono pracę nad produktami (komponentami projektu) po osiągnięciu kamieni milowych. Ale niejednokrotnie osiągnięcie takiej sytuacji byłoby związane ze zbyt długim czasem oczekiwania. W przypadku współbieżnego realizowania wielu zadań ich dynamika przydzielania wartości uzyskanej musi następować w wyznaczonym czasie kontroli projektu, pierwsza kontrola już po ok. 15% wykorzystania zaplanowanego czasu na realizację projektu. Jak już zaznaczono najkorzystniejsza sytuacja jest wtedy, kiedy wykonano zadanie w 100%. Przydzielanie wartości uzyskanej następuje w wysokości 100% *planowanych kosztów*. Jest to dobre rozwiązanie dla krótko-



trwałych zadań.

### **Metoda 50/50**

Metoda 50/50 jest również prosta. Przy rozpoczęciu zadania „uzyskuje” 50% swojego zaplanowanego budżetu, na zakończenie – pozostałe 50%. Gdy operujemy większą liczbą zadań, statystyczny rozkład ich rozpoczęć i zakończeń powoduje, że po pierwszym „skoku”, gdy rozpoczynamy kilka zadań, krzywa *uzyskanych wartości* dość dobrze odpowiada rzeczywistości. Metoda jest dobra, gdy zadania mają zbliżone długości realizacji.

### **Metoda 0-30-70-90-100**

W metodzie tej przyporządkowujemy kolejno 0% wykorzystanego zaplanowanego budżetu na rozpoczęcie aż do 15% zaawansowania prac nad zadaniem, 30% – gdy oceniamy realizację między 15% a 50%, 70% – gdy ocena realizacji mieści się między 50% a 80%; 90% – gdy zrealizowane jest między 80% a 95%; 100% zaś od 95% realizacji do końca zadania. Metoda 0-30-70-90-100 jest oparta na ocenie eksperta. Wkłada on niejako swoją ocenę wartości uzyskanej do jednej z kilku powyższych „przegródek”.

### **Metoda subiektywnej oceny**

Podstawą metody jest ocena postępu prac nad projektem przez eksperta. Metoda ta ma wadę, zwykle oceny eksperta są zbyt optymistyczne. Zwłaszcza gdy ekspert ocenia sam siebie. Najczęściej ekspert ulega deklarowanym przez wykonawców sugestiom, że prace wykonano w 90%, a później przez długi czas są kłopoty ze zrobieniem pozostałych 10% zadania. Szczególnie łatwo o taki błąd w pracach integracyjnych i testowych, gdzie podwyższenie lub zapewnienie wymaganej jakości może istotnie przedłużyć czas zakończenia tych ostatnich 10% nawet do 50% i więcej.

### **Metoda oparta na doświadczeniach z poprzednich inwestycji projektowych**

W metodzie tej porównujemy aktualny projekt z wcześniej wykonanymi projektami. Analizujemy podobne produkty, które były już wykonywane oraz pytamy byłych wykonawców o pracowalność i złożoność prac oraz obciążenie zespołu, jakie było potrzebne, aby produkt wykonać.

### **Estymacja ostatecznych kosztów i terminu zakończenia**

Wybranie jednej z wymienionych metod może ułatwić monitorowanie stanu projektu. Dzięki *wskaźnikowi kosztów* (WK) możemy obliczyć, jaki będzie ostateczny koszt projektu.

Nazywane jest to estymacją ostatecznych kosztów EOK (ang. *estimate et comple-*

tion). Uproszczony wzór na obliczenie tej estymacji wygląda następująco:

$$EOK = (\text{pierwotny budżet projektu})/WK.$$

Podobnie, korzystając ze *wskaźnika harmonogramu*, możemy estymować termin zakończenia projektu. *Estymowany okres realizacji EOR* można obliczyć z uproszczonego wzoru:

$$EOR = (\text{pierwotny planowany czas realizacji})/WH.$$

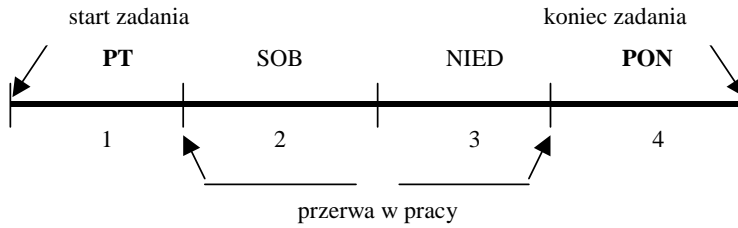
Estymacja opóźnienia to EOR pomniejszony o pierwotny planowany czas realizacji.

### Szacowanie upływu czasu w projekcie

Ze względu na różny stopień wykorzystania czasu pracy, obiektywne przyczyny zewnętrzne angażowania zasobów przydzielonych do projektu, najczęściej nie ma bezpośredniego przełożenia między wysiłkiem a upływającym czasem w harmonogramie projektu.

#### *Wysiłek ≠ Upływ czasu*

- W planach ogólnych należy uwzględniać szkolenia, wakacje, zwolnienia itp., przyjmując około 200 dni roboczych w roku.
- Realnie przeciętny pracownik wykorzystuje na pracę 25 godzin w tygodniu.
- Dla osób odpowiedzialnych za kierowanie ludźmi należy uwzględniać dodatkowo 15–20% na bezpośrednio podległych pracowników.
- Oszacowania powinny być dokumentowane w sposób czytelny – zawsze występuje konieczność korygowania planów.
- Pierwsze przydzielone zadania nie powinny być duże i złożone.
- Należy minimalizować współzależności dla skracania ścieżki krytycznej.
- Ważne jest przydzielanie czasu na przeglądy, aby jakość była uwzględniona w planach projektów.
- Ludzie i miesiące nie mogą być traktowani wymiennie (10 ludzi × 1 mies. ≠ 1 czł. × 10 mies.).
- Czas na przełączenie między zadaniami.
- O około 10% powinno być zwiększone szacowania w celu uwzględnienia działań związanych z zapewnieniem jakości.



Rys. 3.7. Zależność między harmonogramem a terminami według kalendarza

Należy zwrócić szczególną uwagę na kalendarz, dni wolne, święta, urlopy i dostępność zasobów w odniesieniu do kalendarza, jak również szacowaną nieprzerwaną pracę wymaganą do zrealizowania zadania. Na rysunku 3.7 przedstawiono zależność między harmonogramem a terminem według kalendarza:

wysiłek	= praca	= 4 dni kodowania
zasoby	= 2 programistów	
czas trwania	= # nieprzerwany czas wymaganej pracy	= 2 dni
kalendarz	= # ciągły czas kalendarzowy wymagany do pracy	= 4 dni

### Zasoby zużywane na ponoszone koszty rzeczyste projektu

W większości przypadków przydzielone do realizacji projektu zasoby bywają współdzielone np. z innymi projektami lub wykorzystywane są również np. do wspomagania działalności operacyjnej firmy, dotyczy to pomieszczeń, wyposażenia, jak również czasu pracy. Ogólnie zależne jest to od przyjętej w firmie metody ewidencji i rozliczenia kosztów i obciążania nimi ośrodków ich powstawania. Wyróżnić można następujące rodzaje zasobów:

- ludzkie,
- środki materialne (trwałe, nietrwałe),
  - zasoby sprzętu komputerowego: pamięć zewnętrzna, moc obliczeniowa,
  - specyficzny sprzęt biurowy,
  - zasoby programowe, narzędzia, usługi zewnętrzne,
  - infrastruktura,
- środki niematerialne (wiedza, doświadczenie itd.),
- finansowe,
- czas,
- inne.

W ostatnim okresie, ze względu na narastającą konkurencję i postępy technologii informatycznych, które wymuszają skracanie cyklu realizacyjnego projektu oraz ich kosztu, metoda *Earned Value* staje się niewystarczająca i wspomagana jest nową metodą zarządzania zasobami projektu – ludzkimi, materialnymi, sprzętowymi, czasowo-

wymi, tzw. (ang. *Activity Based Costing*) – ABC.

*Activity Based Costing* jest nową koncepcją rachunku kosztów, jest to tzw. rachunek kosztów działań. Zgodnie z tą koncepcją koszty stanowią jedynie pewien miernik zużycia zasobów w trakcie realizacji procesów i działań prowadzących do powstania produktu.

*Kluczowe znaczenie dla zarządzania kosztami ma zrozumienie, że wysokie koszty są jedynie skutkiem, a nie przyczyną niskiej efektywności ekonomicznej przedsiębiorstwa.*

Krzysztof Pniewski [41]

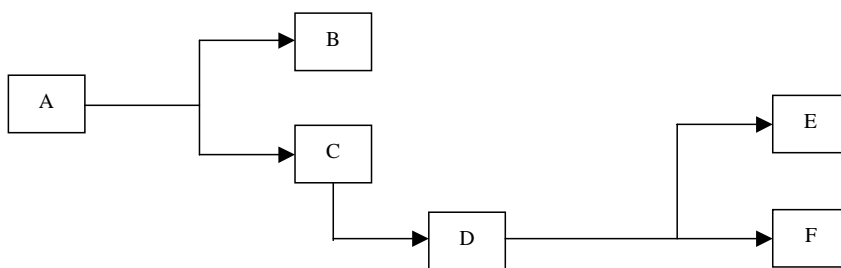
### 3.7. Podsumowanie

*Nie ma tzw. małych zmian w projekcie.*

KF

Skoro nie można uniknąć zmian, więc jak je traktować? Można je ignorować, można też bezkrytycznie akceptować. Ignorowanie zmian, jeśli pojawi się konieczność ich wprowadzenia, nie jest dobrym rozwiązaniem. Zmiana jest pewną akcją korekcyjną wobec naszego projektu i jeśli ją zignorujemy, to możemy doprowadzić do tego, że nasz projekt nie będzie częściowo bądź nawet całkowicie spełniać wymagań klienta. W takim razie akceptujemy wszystkie zmiany, jakie się tylko pojawiają! Niestety nie byłoby to wcale lepsze rozwiązanie. W tym przypadku nagminne i niekontrolowane wprowadzanie zmian do projektu mogłoby spowodować wymknięcie się projektu spod kontroli i minięcie się z jego celem. Dodawanie każdej nowości, np. najnowszej animacji w Flashu, kontrolerek, które stały się hitem na rynku, tylko po to, że są one nowościami nie jest na miejscu. Nasz system ma przede wszystkim funkcjonować, natomiast tworzenie tzw. wodotrysków nieraz było powodem problemów.

Innym aspektem, który tutaj się pojawia jest następujące stwierdzenie: *Do projektu zawsze można wprowadzić później zmiany, więc nie musimy teraz aż tak bardzo się przykladać. Później wszystko się pouklada.* Teraz to znaczy we wczesnych fazach projektu. O jakże zgubne jest ono dla tych, którzy mu zawierzą. Prawda jest taka, że każdy projekt w końcu uda się zakończyć, nawet, jeśli nawet realizowany jest niestannie.



Rys. 3.8. Sekwencja powstawania komponentów projektu oraz ich powiązanie funkcjonalne i czasowe w kontekście postulowanych zmian

Uda się, jeśli dysponujemy nieograniczonymi zasobami i nieograniczonym czasem. W rzeczywistych projektach jednak mamy ograniczony czas, ograniczony budżet, ograniczone zasoby. Z tego wniosek, że im staranniej przykładamy się do realizacji, zwłaszcza we wczesnych fazach projektu, tym lepiej – jest szansa, że w projekcie nie będzie radykalnych zmian. Jeszcze jeden przykład na to, że zwłaszcza wczesne fazy są istotne dla projektu (rys. 3.8). Załóżmy, że podany schemat mówi nam o zależnościach między kolejnymi komponentami i pokazuje czas ich powstawania. Jeśli na początku realizacji projektu trzeba będzie wprowadzić jakieś zmiany do komponentu A, to kolejno powstające komponenty B, C, D, E i F będą już budowane według zmienionego elementu A. Jeśli natomiast wykonamy wszystkie wymienione komponenty i okaże się, że należy dokonać zmian w elemencie A, to pociągnie to za sobą zmiany w pozostałych, czyli dodatkowe zużycie czasu, pieniędzy, środków. Jak widać należy się starać zarządzać problemom, jeszcze zanim się pojawią. Jeśli jednak zmiany trzeba wprowadzić, należy rozważyć czy można je odłożyć, czy też należy je wprowadzić natychmiast, aby uniknąć przedstawionej sytuacji.

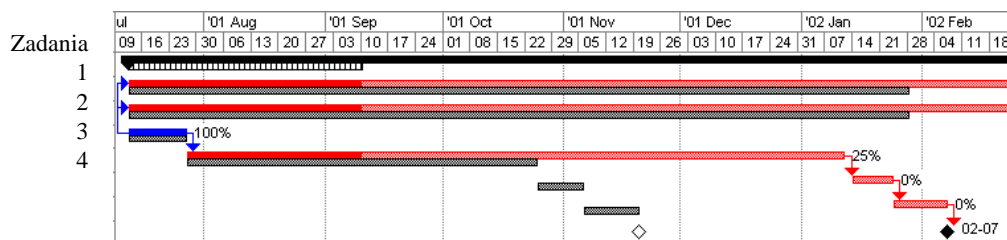
Należy zatem przyjąć, że niezależnie od włożonego wysiłku oraz stopnia kompetencji zespołu, który przygotowywał założenia do projektu, tworzył plan – zmiany są nieuchronnym zjawiskiem związanym z istotą projektu.

*Można by powiedzieć, że stale w projekcie jest to, że nic nie jest stale.*

### Zadanie 3.1

7 września 2003 w dniu kontroli projektu, kierownik programistów poinformował kierownika projektu, że zadanie nr 4 *Implementacja elementów interfejsu* przedłuży się. Jest to dzień, w którym zadanie to powinno być zrealizowane w około 50%. Po oszacowaniu dotychczas zrealizowanych prac kierownik programistów stwierdził, że prace są zrealizowane jedynie w około 25%, co przekazał menedżerowi projektu w postaci raportu. Informacja ta po wprowadzeniu do programu *Microsoft Projekt*

pokazała, że projekt jest opóźniony i skłoniła kierownika projektu do przeprowadzenia analizy metodą *Earned Value*.



Rys. 3.9. Różnice między harmonogramem planowanym a obecnym.

Zad. 1. Implementacja, zad. 2. Konsultacje i doradztwo,  
zad. 3. Nadzór zadań, zad.4. Implementacja elementów interfejsu

Podane wykresy przedstawiają różnice między harmonogramem wcześniej zaplanowanym (szary), a pokazującym obecną sytuację (zaciemniony). Zadanie zaplanowane było na około 2500 h. Ponieważ w połowie jego realizacji kierownik programistów ocenił, że zadanie jest zrealizowane w około 25% – a ich obecny czas realizacji wynosi około 5000 h. Pozostałe zadania 1, 2 trwają odpowiednio 140 dni, zad. 3 – 3 dni.

Przyjmujemy koszt godziny pracy 40 zł dla zadań 1 i 3 oraz 60 zł dla zadań 2 i 4. Obliczyć odstępstwa od planu na podstawie:

1. Wskaźnika harmonogramu *WH*.
2. Odchylenia harmonogramu *OdH*.
3. Wskaźnika kosztów *WK*.
4. Odchylenia od kosztów *OdK*.

## 4. Modele pracy i komunikacji – telepraca

Sytuacje, w których zwykle przychodzi działać PM odbiega znacznie od ideału, który pozwalałby właściwie zaplanować przedsięwzięcie informatyczne. Do rzadkości należą takie projekty, w których na początku PM może zdefiniować jego strukturę (WBS), do zadań i uzgodnionego harmonogramu może przydzielić we właściwej fazie określone zasoby ludzkie, sprzętowe oraz ma przydzielony budżet. To tylko główne elementy, które na samym początku mogą zdecydować czy projekt będzie właściwie zarządzany. Najczęściej mamy do czynienia z sytuacją, że realizujemy projekt w pewnym okresie, np. 1–1,5 rocznym, a w tym czasie zmienia się np. organizacja firmy oraz jej strategia. Do przeszłości należą czasy, kiedy zmiany struktury organizacyjnej w firmie były bardzo powolne i nieznaczne. Przykładem mogą być struktury organizacyjne przedsiębiorstwa w Europie z lat 20. i końca lat 80., aby stwierdzić niewielkie różnice nawet w nazewnictwie komórek organizacyjnych [11, 22, 33, 35, 38].

Dopiero w ostatnich kilkunastu latach nastąpiło „szaleństwo” związane ze zmianą formy zarządzania, organizacji pracy, tzw. *reengineering*. W latach 1979–1995 w Ameryce zlikwidowano blisko 43 miliony miejsc pracy. W Niemczech na początku 1996 roku liczba bezrobotnych osiągnęła największy poziom od czasów drugiej wojny światowej – 4 miliony. W Wielkiej Brytanii, zgodnie z danymi Ministerstwa Pracy z grudnia 1994, od 1990 roku 6,6 miliona mężczyzn, czyli 44% całej populacji siły roboczej było bezrobotnymi

w różnych okresach, w tym czasie ten sam los spotkał ok. 3,9 miliona kobiet, czyli ponad 30% wszystkich pracujących kobiet [11]. Kolejną falę zwolnień obserwuje się w ostatnich miesiącach 2001 r. w USA, w Polsce poziom bezrobocia w maju 2001 r. osiągnął prawie 16% Polaków tj. ok. 3 miliony, powodem zwolnień nie musi być skrajnie zła sytuacja firmy, ale walka konkurencyjna, poszukiwania sposobów obniżania kosztów działalności firm oraz dostępne inne możliwości zatrudniania ludzi – wirtualnie. Jaki zatem jest faktyczny poziom bezrobocia? Czy właściwie je mierzymy w sytuacji, kiedy szacuje się, że obecnie w Europie jest ponad 9 milionów telepracowników, rok wcześniej z tej formy zatrudnienia korzystało 4,6 miliona osób, a w 1997 r. tylko 2 miliony, tak wynika z raportu *Nowe metody pracy*, przygotowanego przez Komisję Europejską [11]. Mamy zatem do czynienia z wirtualnymi firmami, w której pracują wirtualni pracownicy, i praca nadzorowana jest przez sprawnie działające sieci telekomunikacyjno-informatyczne, bez siedzi-

by, a często również osobowości prawnej. Mogą ją tworzyć ludzie znajdujący się w różnych zakątkach świata, lecz pracujący wspólnie nad jednym przedsięwzięciem i indywidualnie świadczący określoną pracę zgodnie z umiejętnościami i potrzebami zleceniodawcy.

Praktyczne wskazówki i informacje w zakresie technik oraz narzędzi wspomagających pracę zespołów zlokalizowanych geograficznie w różnych miejscach są od pewnego czasu coraz częściej poszukiwane. W tej tematyce wyróżniono już odrębne zagadnienia socjologiczne czy implikacje globalnej ekonomiki wirtualnych biur. Nie rzadko podnoszone się problemy kulturowe, których nie można pomijać jako aspektu zwiększenia produktywności, efektywności, elastyczności czy kolokacji zespołu. Mówimy o wirtualnej organizacji, zespole i gdy zapytamy o definicje, czy cechach tych pojęć, możemy usłyszeć dość zróżnicowane poglądy, które koncentrują się wokół następujących zagadnień:

- geograficznej separacji członków zespołu,
- ciągły system pracy,
- struktura przejściowa lub macierzowa,
- zespoły multikorporacyjne.

Obserwuje się dynamiczny rozwój outsourcingu, w warunkach coraz większej konkurencji na rynku, coraz mniej firm stać jest na stworzenie i utrzymanie zespołu projektowego. Zróżnicowanie projektów uniemożliwia wręcz stworzenie dostatecznie elastycznej grupy, mogącej podjąć się coraz trudniejszych zleceń. Dynamika rozwoju informatyki skłania pracowników do specjalizacji w wąskiej dziedzinie, w której mogą ubiegać się o miano eksperta. Trudnym zadaniem jest stworzenie zespołu, zawierającego wszystkie ogniwa konieczne do podjęcia każdego wyzwania. Dostęp do ekspertów staje się coraz bardziej znaczący. Niejednokrotnie możliwe jest to tylko w sposób zdalny. Dzięki nowoczesnym technikom porozumiewania się, ludzkość stoi u progu pokonania barier komunikacyjnych bez względu na odległość, a to daje możliwości tworzenia zespołów ściśle wyspecjalizowanych w dziedzinie problemów, z którymi konwencjonalne zespoły nie mogą wygrać.

Ponieważ dostęp do tego rynku pracy dotyczy jednak niewielu, więc są jego zwolennicy i przeciwnicy, tak jak dla idei globalizacji, która będzie sprzyjała takim tendencjom. Jest to zrozumiałe w sytuacji patrzenia na świat w zmniejszonych proporcjach dzięki rozwojowi komunikacji i telekomunikacji. Dla kogo takie trendy są szansą, a dla kogo zagrożeniem w sytuacji, kiedy uwzględnimy, że świat to „wioska” zamieszkała przez 100 osób, w której: 12 osób to Europejczycy, 60 – Azjaci, 14 – Afrykanie, 1 – Australijczyk, 13 – Amerykanie. Ponadto: 6 osób ma prawie 3/5 dochodów całej wioski i jest 1 komputer.

## 4.1. Wirtualne zarządzanie zasobami - elektroniczny PM



Zróźnicowanie stref czasowych umożliwia niemal ciągłą pracę 24 godziny na dobę, gdzie poszczególne zadania przejmowane są przez grupy ludzi z różnych części globu [18]. Zwiększa to elastyczność, wydajność i dyspozycyjność do świadczenia pracy. Niesie również ze sobą nowe zagrożenia. Różnice występują w sposobie pracy, mentalności pracowników, są też różnice kulturowe. Wymaga to bardzo skrupulatnego przestrzegania zasad komunikacji, definiowania wspólnych pojęć i celów. Jednakowe zrozumienie i interpretacja współdzielonych danych jest ważnym aspektem tworzenia i działania zespołów wirtualnych.

Przed projektem menedżera (*PM*) stoją całkowicie nowe wyzwania. Wiązą się one z budową zespołu, z pokonaniem barier kulturowych, oceną kosztów i złożoności komunikacji, z określeniem jednolitych standardów obowiązujących cały zespół.

Tabela 4.1. Telepracownicy w Unii Europejskiej (w latach 2000–2010)

	2000	2010
Na stałe zatrudnieni w domu	810 000	3 170 000
Pracownicy pracujący w wielu miejscach	3 700 000	14 332 000
Niezatrudnieni na stałe, świadczący usługi	1 450 000	3 040 000
Pracujący w domu, nie świadczący usług	3 080 000	6 580 000
Suma	9 040 000	27 122 000

Źródło: *Emergence analysis*, 2001 [5].

W istocie elektroniczny *PM* to rzeczywisty człowiek specjalista w dziedzinie zarządzania przedsięwzięciami projektowymi, który poza doświadczeniem i wiedzą niezbędną w kierowaniu i zarządzaniu ludźmi jest wyposażony w odpowiednie środki techniczne oraz narzędzia programowe, którymi posługuje się sprawnie, aby zaplanować i monitorować cały cykl życia przedsięwzięcia. Dodatkowym zadaniem *PM* działającego w środowisku wirtualnym jest zorganizowanie komunikacji z zespołem poprzez efektywne wykorzystanie środków technicznych zapewniających dostęp do wiedzy, częściowych wyników prac współdziałających zespołów oraz zarządzania zmianami.

## 4.2. Elektroniczny partner, członek zespołu

Elektroniczny członek zespołu lub zespół to istniejący rzeczywiście specjaliści, którzy pracują niejednokrotnie we własnych mieszkaniach i najczęściej w takich strefach ekonomicznych, gdzie jest duża podaż specjalistów, a ich oczekiwania finansowe za świadczoną pracę są znacznie niższe niż w centrali firmy. Dziś niejednokrotnie bardzo złożone prace mogą realizować wirtualne zespoły na własnym sprzęcie komputerowym (najczęściej) lub udostępnionym, mając zapewniony dostęp do Internetu. Dynamiczny rozwój sieci telekomunikacyjnej oraz telefonia komórkowa praktycznie umożliwia pracę

prawie w każdych warunkach. Nowoczesne miejsce pracy zapewniające pełny komfort obejmuje: telefon, komputer osobisty połączony z siecią, laptop, fax, łącze internetowe i łącze wideo. Taka praca jest pochodną innych znanych modeli organizacyjnych zespołów w zarządzaniu projektami, ponieważ celem ich jest:

- przedstawienie modeli organizacji zespołów projektowych w zróżnicowanych organizacyjnie firmach,
- analiza zadań pod kątem dopasowania – zastosowania adekwatnej metody organizacji zespołu a typem projektu,
- jak wykorzystać zróżnicowaną wiedzę poszczególnych członków zespołu do realizacji wspólnego celu.

**Zespół – grupa**, jest wtedy, gdy ma wspólne cele i zdaje sobie z tego sprawę, że do ich osiągnięcia potrzebne są wysiłki każdego z jej członków. Zespół jest wówczas, kiedy grupa ludzi sama uważa się za zespół, gdy zmierza w zespołowym kierunku i gdy ma własne zespołowe sposoby działania, swoisty sposób zachowania.

### Symptomy tworzenia, powstawania zespołu

Aby można było nazwać grupę ludzi zespołem, muszą oni mieć wspólny cel, współzależne zadania, wspólną odpowiedzialność i wzajemne zaufanie.

Proces tworzenia wirtualnego zespołu projektowego należy do *projektu menedżera* (PM). Jego zadaniem nie jest jedynie stworzenie własnej wizji współpracy. Budowanie zespołu pociąga za sobą również:

- Określenie wspólnej wizji dla zespołu.
- Zbudowanie infrastruktury dotyczącej technologii, nadzoru, ułatwienia komunikacji.
- Utworzenie współdzielonego repozytorium wiedzy.
- Zbudowanie dobrych relacji pomiędzy członkami zespołu.
- Selekcję i ocenę członków zespołu.
- Stworzenie atmosfery satysfakcjonującej pracę w zespole.

W MSI (ang. *Management Strategies, Inc.*) opracowano dwa współpracujące ze sobą modele efektywnego tworzenia zespołów rozproszonych.

• **Model dopasowania** – wspomaga projekt menedżera w selekcji i ocenie kandydatów.

• **Model dojrzewania** – pomaga ocenić istniejącą strukturę zespołu rozproszonego, by umożliwić jego rozwój do postaci umożliwiającej uzyskanie większej wydajności.

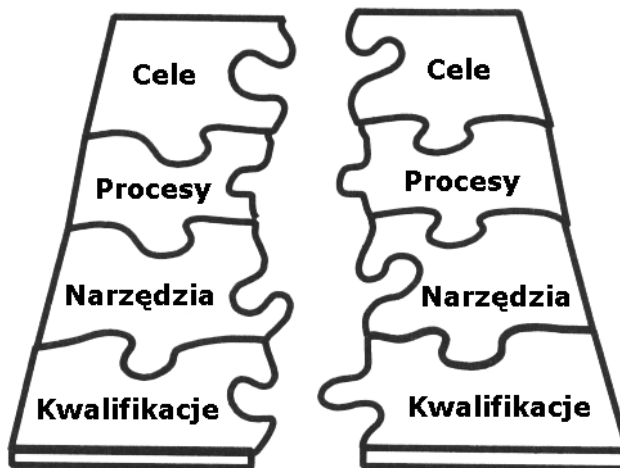
Aby zastosować z powodzeniem podane modele, należy dokładnie zdefiniować główne aspekty współpracy między członkami zespołu wirtualnego. Odnoszą się one do określenia standardów dostępności i potwierdzania otrzymania informacji, utworzenia wspólnego kontekstu wszystkich przesyłanych informacji. Komunikacja synchroniczna, zapewniająca umieszczenie wszystkich stron komunikacji w tej samej „czasoprzestrzeni” umożliwia szybkie zbudowanie korzystnych relacji między człon-

kami zespołu. Wprowadzenie priorytetów w rozsyłaniu informacji, aby zapobiec efektowi przeładowania.

W zespołach rozproszonych duże znaczenie ma *repozytorium* wiedzy, jest to najczęściej dziedzinowa baza danych, która zawiera wszystkie bieżące i historyczne informacje dotyczące projektu. Coraz rzadziej zdarza się, aby w zespołach projektowych osoby uczestniczące w realizacji projektu były tam od samego początku. Cenna wiedza gromadzona przez osoby zaangażowane na różnych etapach, nie może zostać bezpowrotnie utracona. Stworzenie centralnej bazy wiedzy umożliwia korzystanie z repozytoriów, dokumentów, formularzy, opracowań czy kodów źródłowych tworzonych przez zespół. Umożliwia szacowanie rozmiaru kolejnych projektów, ich czasochłonności i złożoności przez podobieństwo do informacji zgromadzonych w repozytorium.

Użycie *modelu dopasowania* koryguje procedurę doboru osób zaangażowanych w realizację projektu. W przypadku klasycznym *Project Manager*, kieruje się przede wszystkim umiejętnościami technicznymi kandydata, zakłada się, że zatrudniona osoba wdroży się w narzucony styl i narzędzia pracy. To podejście nie ma zastosowania w przypadku osób zatrudnianych zdalnie.

Rdzeniem *modelu dopasowania* jest uzmysłowienie sobie, że zatrudnienie osoby z zewnątrz organizacji zatrudniającej pociąga za sobą rozbieżności w stylu komunikacji i pracy między pracodawcą a pracownikiem. Konieczne staje się dopasowanie dwóch części układanki, dającej w efekcie owocną współpracę. Jej elementami są: cele, procesy, narzędzia i kwalifikacje (rys. 4.1.)



Rys. 4.1. Model dopasowania

Pierwszą częścią są cele. W procesie ich definiowania nie wystarczy rozpatrzenie podstawowych elementów. W zespołach rozproszonych o różnych uwarunkowaniach

kulturowych, w dużym rozproszeniu geograficznym konieczne jest dogłębne zdefiniowanie używanych pojęć. Określenia „jakość” czy „na czas” mogą mieć zbyt rozbieżne znaczenia. Wspólne definicje używanych pojęć są konieczne przy definiowaniu wspólnych celów.

Wspólne procesy mają wpływ na dopasowanie telepracownika do zespołu. W przypadku stosowania rygorystycznych procesów wytwarzania oprogramowania ważne jest zatrudnienie pracownika znającego i rozumiejącego obowiązujące procedury postępowania.

Posługiwanie się wspólnymi narzędziami i środowiskami pracy jest bardzo ważnym aspektem pracy w grupach rozproszonych. Komunikacja elektroniczna niejednokrotnie stanowi jedyny pomost łączności między członkami zespołu. Korzystanie z jednolitych rozwiązań technologicznych pozwala na zachowanie spójności przesyłanych informacji. Opóźnienia spowodowane trudnościami w komunikacji mogą stanowić nawet do 25% czasu realizacji projektu.

Ostatnim elementem układanki są umiejętności. Oprócz ściśle związanych z technologią liczą się również umiejętności związane z komunikacją oraz samą organizacją pracy. Dopasowanie tych elementów jest konieczne do stworzenia prężnie działającego wirtualnego zespołu projektowego.

Model dojrzewania opiera się na czterech poziomach struktur zespołu wirtualnego. Na każdym z tych poziomów pojawiają się kluczowe problemy, charakterystyczne dla każdego z nich. Model dojrzewania jest w stanie określić czas, jaki zajmie organizacji przejście z aktualnego poziomu do kolejnego. Miarą wydajności zespołu jest jego zdolność do realizacji projektów w narzuconym czasie i budżecie. Przejście do kolejnych poziomów zwiększa wydajność przez optymalizację poszczególnych aspektów działania zespołu.

Zespół niezależnie od fizycznego usytuowania, sposobu komunikacji oraz realizowanego zadania na pewnym etapie działania nabywa cech pozytywnych oraz negatywnych, do których należą między innymi:

- swoista ideologia,
- normy współdziałania i bycia egzekwowane są w sposób naturalny – nie ma regulaminu,
  - myślenie grupowe,
  - brak krytycznego spojrzenia na efekt pracy grupy,
  - nie rozważanie jakichkolwiek rozwiązań przyjętych poza grupą,
  - następuje spadek jakości pracy,
  - manifestowanie jednolitości i lojalności.

Niektóre ze zjawisk mają przyczynę w istotnych różnicach wynikających z porównania pracy między zespołem tradycyjnym a wirtualnym, takich jak:

- w zespole tradycyjnym:
  - dyskusje prowadzone bezpośrednio, widać reakcje i temperament rozmówcy,
  - często praca samodzielna,

- zasoby w bliskim otoczeniu,
- praca sekwencyjna,
- informacja przechowywana lokalnie;
- w zespole wirtualnym:
  - sesje dyskusyjne wirtualne (czat, gadu-gadu, e-mail),
  - dokumenty elektroniczne,
  - informacja w globalnym – ogólnodostępnym lub selektywnym repozytorium,
  - praca wykonywana wspólnie,
  - dostęp do zasobów poprzez połączenia.

Tabela 4.2. Różnice między tradycyjnym i wirtualnym zespołem projektowym

Tradycyjny zespół	Wirtualny zespół
Członek z tej samej organizacji	Członek z organizacji, firmy przynależnej biznesowo, konkurent
Członek wyuczony, często polecony, ustalający standardy	Członek dobrany z powodu wykazanych przez niego kompetencji
Mała ufność	Żądanie dużej ufności
Procesy pracy sztywne i zdefiniowane, często nieużyteczne	Procesy pracy elastyczne dopasowane do zespołu, dostosowane do projektu
Struktura zespołu hierarchiczna	Redukcja hierarchii, więcej zależności sieciowych
Stabilne środowisko pracy	Ciągłe zmiany środowiska pracy
Ważne pozycja i władza	Ważna wiedza i zdolności

Grupowe produkty zespołów wirtualnych generują rozwój oraz wymogi w kierunku sprawniejszej i multimedialnej komunikacji, sprzedaż produktów grupowych w 1997 r. w USA zwiększyła się w stosunku rocznym prawie o 100% [23], ale jednocześnie w ciągu ok. 30 dni instalowanych jest prawie 1 milion korporacyjnych skrzynek e-mailowych.

#### Przykład:

Przy realizacji projektu PM, realizowanego w okresie od kwietnia 2001 r. do grudnia 2002, w listopadzie 2001 r. do projektu było zaangażowanych 7 osób. W tym czasie harmonogram obejmował 5 wydzielonych zadań między innymi: *modelowanie danych*, *modelowanie procesów*, *opracowanie logiki stanów*. W ciągu miesiąca zespół (3 osoby we Wrocławiu, 2 w Łodzi, 1 w Katowicach, 1 w Gdyni) wymienił pomiędzy sobą ponad 1000 e-maili, czyli przeciętnie każdy z członków zespołu otrzymał lub wysłał łącznie średnio ponad 140 e-maili. Niezależnie od poczty był dostęp do wspólnych zasobów – repozytorium w Lotusie i grupowa praca nad produktami. W repozytorium gromadzono np. wspólne ustalenia, wzory dokumentów, korekty dokumentów, uwagi, polecenia, monitorowanie zmian, opiniowanie itd. Etapy kończyły się spotka-

niem zespołu w trakcie videokonferencji [18, 30].

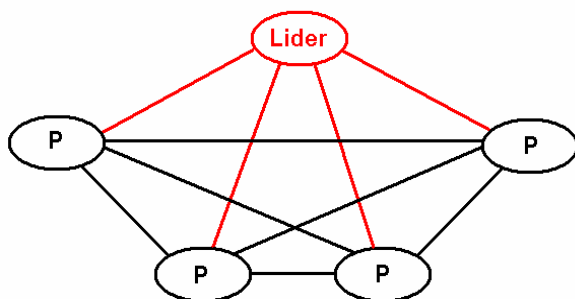
### 4.3. Podział i modele organizacyjne zespołów

Organizacja jest wskaźnikiem stopnia dojrzałości organizacji oraz jej podejścia do tego zagadnienia. Wyróżnia się następujące modele organizacyjne pracy zespołów:

- sieciowy,
- gwiazdzisty,
- izomorficzny,
- specjalizacyjny,
- nieegoistyczny,
- macierzowy
  - zadaniowy,
  - funkcjonalny.

**Zespół sieciowy** (demokratyczny) charakteryzuje się dosyć równym udziałem wszystkich członków w podejmowaniu decyzji (rola lidera może w zasadzie być przechodnia). Uczestnicy komunikują się każdy z każdym (rys. 4.2). Z tego względu liczba członków

w takim zespole nie powinna przekraczać 8–12. Zaletą tej struktury jest łatwa możliwość rekonstrukcji zespołu w razie odejścia lidera, gdyż wszyscy członkowie mają duży wgląd w postęp prac projektowych. Lider ma za zadanie wyłącznie koordynację prac, reprezentowanie zespołu i pełnienie funkcji administracyjnych. W zespole takim nie mogą się pojawiać członkowie nowi, niedoświadczeni, gdyż nie nadążaliby za tempem pracy pozostałych. Model taki dobrze sprawdza się we wczesnych fazach prac nad projektem („burza mózgów”), gdy wymagane jest powstawanie twórczych koncepcji. Może się wraz z upływem czasu okazać, że zespół o strukturze demokratycznej przerodzi się w zespół o bardziej scentralizowanej strukturze (np. zespół o strukturze gwiazdzistej).

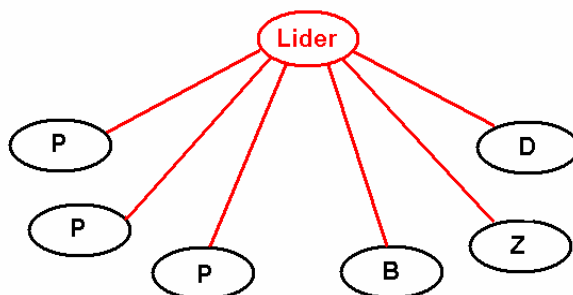


Rys. 4.2. Struktura sieciowa współdziałania członków zespołu

W strukturze sieciowej występują takie cechy, jak:

- istnieje komunikacja między poszczególnymi osobami,
- nie ma podziału ze względu na odległość służbową lub organizacyjną,
- grupy kilkuosobowe (zalecane maksymalnie 8 osób),
- grupa początkująca,
- szybko wypracowuje standardy dokumentowania pracy itp.

**Zespół** o organizacji **gwiazdzystej** charakteryzuje się silną, centralną pozycją lidera, który pośredniczy w wymianie wszystkich informacji między członkami zespołu (rys. 4.3). Lider jest jedyną osobą znającą całość prac projektu i jego odejście jest dużym zagrożeniem dla projektu. Lider przydziela pracę poszczególnym członkom zespołu i nadzoruje wykonanie przez nich zadań. Nie ma tu miejsca na dyskusje i demokratyczne ustalenia. Za to zespół może być liczniejszy (do pewnych rozsądnych granic – wyznaczanych przez możliwości komunikacyjne lidera), jak również mogą w nim znaleźć miejsce osoby o różnym poziomie zaawansowania (zwłaszcza początkujące, którym lider może poświęcić nieco czasu na udzielanie rad). Model ten sprawdza się w dalszych fazach cyklu produkcji oprogramowania, dzięki scentralizowanej władzy. Angielska nazwa tej struktury wzięła się od sposobu organizowania zespołów programistycznych stosowanego w latach siedemdziesiątych przez IBM. Zespół taki składał się z 3–4 stałych członków: głównego programisty (*chief programmer*), programisty zapasowego (*backup programmer*), oraz bibliotekarza (*librarian*). Do nich w razie potrzeby przydzielano pozostałych niezbędnych pracowników.



Rys. 4.3. Struktura gwiazdzysta współdziałania członków zespołu

W strukturze gwiazdzystej można wyspecyfikować takie cechy działania, jak:

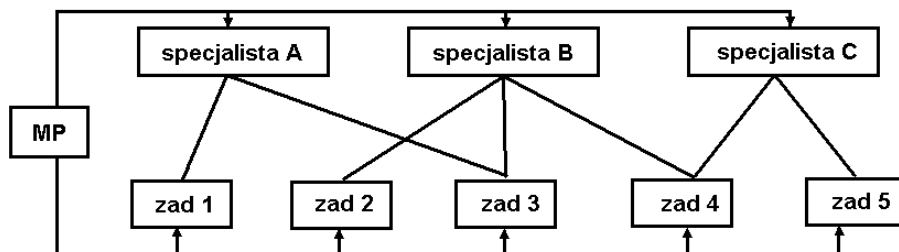
- wszystko skupia się wokół jednej osoby – szefa,
- szef współpracuje ściśle, ale osobno z każdą osobą z grupy,
- wymiana informacji między członkami zespołu jest za pomocą kierownika,
- kierownik przydziela zadania i ocenia efekty pracy,
- zespoły z niedoświadczonymi ludźmi (kierownik prowadzi „za rękę” pomaga),

- liczność większa niż przy sieciowej zależna od możliwości kierownika.

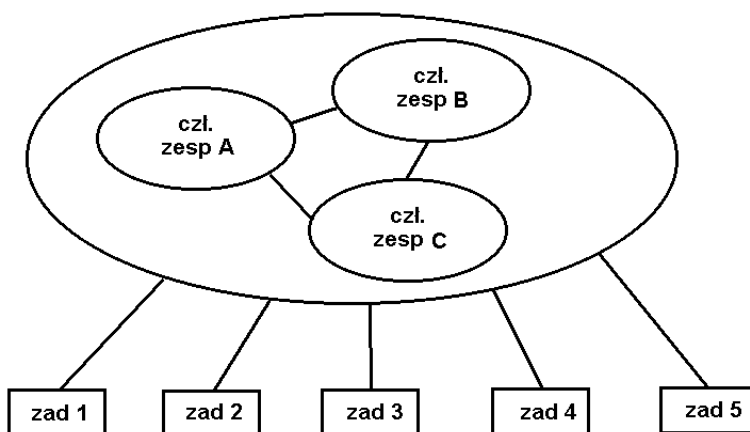
Problemem zasadniczym staje się zmiana kierownika, jak również jego czasowa nieobecność, np. choroba, urlop.

**Struktura izomorficzna** najczęściej charakteryzuje się takimi cechami, jak:

- odwzorowuje struktura projektu,
- oddaje strukturę projektu w strukturę zespołu,
- opracowanie dokumentów projektu zgodnie z kompetencjami zespołu.



Rys. 4.4. Struktura specjalistyczna zespołu



Rys. 4.5. Struktura nieegoistyczna zespołu

**Struktura specjalistyczna** (rys. 4.4) jest dostosowana do możliwości i kwalifikacji zespołu, jak:

- zadania dla osób według ich specjalizacji,
- odpowiedzialny za całość projektu najbardziej doświadczony członek zespołu.

Podstawą pracy zespołu o **strukturze nieegoistycznej** są zadania i ich znaczenie (rys. 4.5). Praca członków zespołu podporządkowana jest ich realizacji i zespoły mają wspólny cel (w różnym czasie lub stopniu) w wykonaniu zadania. Wykonane zadanie



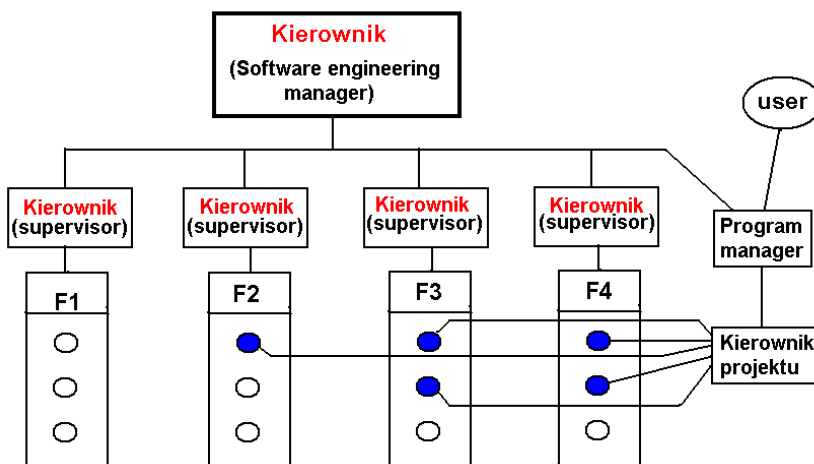
z sukcesem lub jego niepowodzenie to zasługa całego zespołu. Informacja dotycząca „wkładu” w całość projektu i poszczególnych zadań nie jest przedmiotem personalnego identyfikowania się z produktem końcowym czy zadaniem.

#### 4.4. Projekty w firmie o strukturze macierzowej

Rzeczywista firma i jej zdolność do realizacji projektu przez wirtualne tworzenie sieci partnerów jest niejednokrotnie podstawą jej sukcesów. Modele organizacyjne, ich cechy i specyfika omówiona jest w pracach [7, 11, 28, 38]. Najczęściej obserwujemy ewolucje firm o strukturze macierzowej (rys 4.6), które dopasowują swoją organizację pod sprawną obsługę procesów, przez nią przebiegających. Nowe zadania, nietypowe projekty w takiej strukturze wymuszają tworzenie sieci partnerów. W strukturze macierzowej mamy do czynienia:

- z podwójną podległość pracowników,
- konfliktami wobec podwójnego podporządkowania przy próbach elastycznego działania na styku różnych komórek i w przypadku przeciążenia zadaniami,
- podwójny system kontroli i oceny wyników pracy, niejednokrotnie trudności równoważenia obowiązków formalnych i wkładem pracy, co ma bezpośredni wpływ na nagradzanie pracowników.

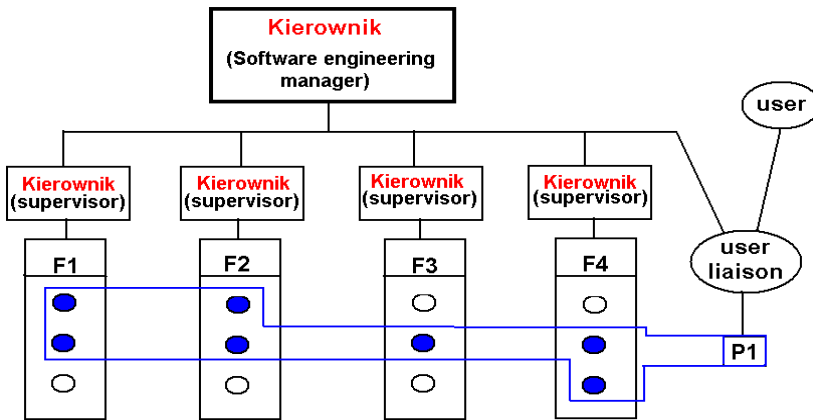
#### Struktura macierzowa



Rys. 4.6. Podwójny łańcuch podporządkowania w realizacji projektów w firmie o strukturze macierzowej – zadaniowej

Możemy wyróżnić dwa rodzaje osób kierujących (zarządzających): kierownika projektu (ang. *software project manager*) i kierownika zespołu funkcyjnego (ang. *supervisor*). W zależności od ich wpływów i siły struktury macierzowe można podzielić na słabe, silne i zbalansowane. Dla struktury macierzowej słabej rola kierownika projektu jest ograniczona bardziej do roli koordynatora i „przyspieszacza” niż zarządzającego. Przy strukturze silnej macierzowej kierownik projektu ma znaczny autorytet (prawo podejmowania decyzji) i uprawnienia administracji załogą. Zazwyczaj kierownik projektu odpowiada za sprawy związane z zarządzaniem, harmonogramowaniem i planowaniem, natomiast kierownik funkcjonalny jest odpowiedzialny za techniczną stronę projektu, zajmuje się szkoleniami i karierą pracowników.

### Struktura funkcjonalna



Rys. 4.7. Podwójny łańcuch podporządkowania w realizacji projektów w firmie o strukturze macierzowej – funkcjonalnej

Załogi są grupowane z uwzględnieniem specjalności, funkcji oprogramowania lub podobnych, takich jak: zespół inżynierów systemowych, zespół testerów, zespół pomocy technicznej (ang. *software support*), zespoły zastosowań inżynierii oprogramowania, zespół analityków branżowych, hurtowni danych itp.

Zasięg projektu ograniczony jest do zespołu funkcyjnego, tzn. komunikacja poszczególnych zespołów odbywa się przez przełożonych, tj. kierowników zespołów dziedzinowych. Zarówno pytania, jak i produkty końcowe przekazywane są od jednego zespołu do drugiego przez przełożonych.

Taka organizacja w dalszym ciągu w wielu firmach zdaje egzamin, ale duży postęp w zakresie stosowania nowoczesnych systemów informatycznych głównie w samych firmach informatycznych oraz zmiana kultury kierowania przez partnerstwo ludzi myślą-

cych, mających dużą swobodę działania – nie tylko czekających na polecenia i ich wykonanie, zmienia model zarządzania. Sieci korporacyjne, intranet, e-mail, połączenia ISDN na potrzeby wideokonferencji, telefon stacjonarny i komórkowy spowodowały, że możemy mówić o rodzącym się zjawisku samokreowania warsztatu i miejsca pracy. Mój pracodawca to ja sam z moimi umiejętnościami świadczenia pracy i przesyłania jej wyników w postaci elektronicznej. Idziemy w kierunku kontraktowania zadaniowego i czasowego swej pracy, a nie zatrudnienia na etat.

Za pomocą środków i narzędzi informatycznych wykonawcy zadań oraz centrala firmy, na każde żądanie i w każdej chwili ma zapewniony dostęp do rezultatów pracy, może jednym e-mailem powiadomić wszystkich lub wybraną grupę pracowników o swoich decyzjach lub zadaniach.

Udostępnić wiedzę i prowadzić edukację przez organizację dziedzinowych repozytoriów, np. z wykorzystaniem Lotus Notes lub WebDB, Visual Source Safe (VSS). Istnieje rodzina programów wspomagająca zarządzania konfiguracją typu RCS, Case Ware, Aide-De-Comp, DSEE, Rational, NSE, które pozwalają na łatwe kontrolowanie procesu zmian w projekcie i zarządzania wersjami (dodatkowe informacje w rozdz. 9). Tym samym szczeble pośrednie stają się zbędne, struktura znacznie się spłaszcza, ale powstaje inny problem potrzeby wykreowania specjalistów, którzy zweryfikują tworzone rozwiązania i wykorzystają je w firmie. Budowane narzędzia i technika komunikacyjna sprzyja budowie systemu scentralizowanego z możliwościami zdecentralizowanych i samodzielnych decyzji w małych zespołach zadaniowych.

## 4.5. Wirtualna sieć partnerów przedsięwzięcia

Choć słowo „wirtualny” znaczy nierzeczywisty, nieprawdopodobny, ale możliwy bez cech fizycznych, jest bytem realnie mogącym działać. W tym nowym otoczeniu dotychczasowe metody zarządzania są niewystarczające. Adekwatnie do potrzeb wirtualna sieć zespołów potrzebuje wirtualnego zarządzania. To pojęcie jeszcze nie jest upowszechnione, łączy się z metodami kreowania zespołów wirtualnych, pobudzania indywidualnej przedsiębiorczości, przekazywanie efektów pracy na odległość, czyli telepraca, efektywne kierowanie specjalistami, zapewnienie ciągłego uczenia się w przedsiębiorstwie i przekształcenie go w uczącą się organizację, która umie zarządzać wiedzą.

### **Kto za co odpowiada i czym dysponuje**

Jeśli głównymi elementami stanowiącymi o sukcesie wirtualnego zarządzania jest elastyczność oraz kreatywna integracja, to dotychczasowe systemy planowania, sterowania kontroli nie spełniają już swojej roli. Pewne elementy kontroli, jak np. czas pracy, dyscyplina formalna, harmonogram w pewnym zakresie jest poza kontrolą i nie

spełniają swojej roli. Pracownicy przejmują w znacznej części funkcje menedżerów – myślenie w zakresie co i jak zrobić, nie otrzymują systematycznych poleceń, które skrupulatnie mają wypełniać. Pracownika nie obserwuje kierownik i nie widzi jego zmęczenia czy patrzenia w „sufit”. Menedżerowie sterują zespołem wirtualnym przez wspólne zdefiniowanie celów i występują w roli konsultantów i trenerów. Kierownictwo firmy dysponuje odpowiednią infrastrukturą i zatrudnia menadżerów zajmujących się zarządzaniem strategicznym. Za działalność operacyjną w większości przypadków odpowiadają rozproszone zespoły czy pojedynczy ich członkowie, co znacznie zwiększa wymagania wobec wszystkich i jest elementem mobilizującym. Każdy kto bierze udział w takim projekcie ma świadomość swojej ważności, że jest specjalistą, a więc pracownikiem dysponującym dużą wiedzą, jest dobrze wykształcony, pomysłowy, odznacza się umiejętnością działania w zespole i dobrze znosi pracę bez żywych kontaktów (atmosfery biurowej) i jest odporny na stres związany z brakiem kontaktów interpersonalnych.

### Gdy mamy zespół i organizację

Gdy mamy infrastrukturę, środki techniczne oraz organizację rozproszoną jak niemal sieć www, wówczas jako firma mająca zdefiniowany cel możemy realizować złożone projekty, ale możemy również oferować na bazie utworzonej organizacji – zespół telepracowników, oferować nowy rodzaj usług np.:

- ICC (*Internet Communication Center*), usługa hostingu serwerów, w której usługodawca udostępnia serwer we własnej serwerowni i odpowiada za stan oraz aktualizację plików.
- ASP (*Application Service Provider*), usługa udostępniania aplikacji dla klientów za miesięczną opłatę.
- Serwisowanie oprogramowania przez Internet.
- E-commerce, inne.

Tabela 4.3. Zalety i wady struktur macierzowych

Struktura zespołu	Zalety	Wady
Funkcjonalna	1. Szybki start- organizacja już istnieje <ul style="list-style-type: none"> <li>• łatwe zarządzanie pracownikami</li> <li>• ustalone są narzędzia, techniki, standardy itp.</li> </ul>	1. Brak centralnej odpowiedzialności za projekt 2. Problemy z interfejsami komunikacyjnymi 3. Trudna kontrola i monitorowanie
Projektowa	1. Centralna odpowiedzialność za projekt 2. Centralna kontrola interfejsów komunikacyjnych 3. Szybkość podejmowania decyzji 4. Wysoka motywacja załogi	1. Konieczność tworzenia nowego zespołu dla każdego projektu 2. Trudniejsze zarządzanie pracownikami 3. Brak ustalonych narzędzi, metod standardów, itp. na starcie projektu

Macierzowa	<ol style="list-style-type: none"> <li>1. Wady powyższych projektów złagodzone</li> <li>2. Bardziej elastyczne zarządzanie ludźmi</li> </ol>	<ol style="list-style-type: none"> <li>1. Dwóch przełożonych</li> <li>2. Trudności w monitorowaniu i przepływie dokumentów, większy nakład czynności koordynujących</li> <li>3. Możliwy konflikt przywódców</li> </ol>
------------	--	--

Wprowadzenie systemów elektronicznej komunikacji wewnętrznej i internetowych baz danych, dostępnych z każdego miejsca na świecie, praktycznie wyeliminowało problemy dostępu do danych związanych z lokalizacją miejsca pracy. Dziś, pracując nad konkretnym projektem, firma może zwerbować zespół najlepszych specjalistów z danej dziedziny, nie przejmując się miejscem ich zamieszkania. Komunikacja przez Internet ułatwia też organizację czasu pracy. Internetowa współpraca nie budzi jednak entuzjazmu u wszystkich. Pewne kraje, jak USA, Wielka Brytania, mają większe doświadczenia w pracy wirtualnej, w krajach tych lansowano tezę, że przyszłość to praca w domu. Ale nie wszyscy pracownicy akceptują taki model pracy, nie wszyscy czują się w nowej sytuacji lepiej. Nadal przyjeżdżamy do firmy, aby zaspokoić nasze potrzeby społeczne: spotkania się z kolegami, rozmowy, śmiechu. Psycholodzy ostrzegają, że jeśli pracujemy w domu, możemy mieć trudności z określeniem, gdzie kończy się życie zawodowe, a zaczyna prywatne. Przy bardziej stresujących zadaniach i napiętych terminach może to doprowadzić do tego, że o firmie będziemy myśleć od porannej toalety aż po wieczorną. Przypominać o niej będzie włączony komputer i sterta dokumentów na biurku i w sypialni. Rozwiązaniem może być wydzielenie odrębnego pokoju do pracy i żelazne przestrzeganie godzin jej zakończenia. Ryzyko realizacji dużych projektów w takiej organizacji może być minimalizowane przez właściwe proporcje między pracownikami pracującymi tradycyjnie i wirtualnie. Główną barierą w upowszechnianiu wirtualnej pracy nie są ograniczenia technologiczne, ale mentalność, zarówno pracowników, jak i ich przełożonych.

## 4.6. Zespoły projektowe – praca grupowa

Rozwój informatyki skłania pracowników do specjalizacji w wąskiej dziedzinie, w której mogą ubiegać się o miano eksperta. Trudnym zadaniem jest stworzenie zespołu, zawierającego wszystkie ogniwa konieczne do podjęcia każdego wyzwania. Zasób wiedzy, umiejętności, jakie należy przyswoić w określonym czasie, jak również ją aktualizować, powoduje, że specjalizacja i podział kompetencji jest dominujący w organizacji, która wytwarza produkty informatyczne. Dostęp do ekspertów staje się coraz bardziej znaczący i niejednokrotnie decyduje o powodzeniu przedsięwzięcia. Nie zawsze jest to możliwe w danej firmie, często należy szukać na zewnątrz lub niektóre prace outsoursować. Zachodzi potrzeba budowy zespołów ściśle wyspecjalizowanych w dziedzinie problemów, z którymi konwencjonalne zespoły nie mogą

wygrać. Czasy kiedy informatyk znał budowę komputera oraz 2 podstawowe języki programowania, np. assembler oraz Coboll i wystarczały mu do zajmowania właściwej pozycji w każdym projekcie należą do przeszłości. Ogromna liczba wykonywanych zadań w projekcie wymaga współpracy kilku, a niekiedy kilkudziesięciu osób, czyli wyspecjalizowanych zespołów oraz pracy grupowej (ang. *collaboration*).

## Rodzaje zespołów projektowych

Typowymi zespołami, które powołuje się na czas realizacji projektu lub na stałe są utrzymywane w firmie to:

- **zespół wytwarzania aplikacji**

- zależność od technologii,
- znajomość narzędzi i technik wytwarzania oprogramowania,
- role:

klient/zamawiający, kierownik, kierownik techniczny, analityk, projektant, implementator, testujący, dokumentujący, administrator

- **zespół identyfikacji i analizy potrzeb/wymagań**

- znajomość dziedziny  $\leftrightarrow$  znajomość metod informatycznych,
- role:

analityk, dokonujący interview, protokołujący

- **zespół serwisu**

- sprzętu informatycznego i infrastruktury,
- oprogramowania systemowego,
- oprogramowania użytkowego

- **zespół utrzymania**

○ funkcje: obsługa, usuwanie usterek, rozwój, dostosowanie do nowych potrzeb, doradztwo,

- role:

użytkownik, operator, inżynier systemów, administrator bazy danych, programista, konfiguracji

- **zespół wdrożeniowy**

- znajomość strategii wdrożenia,
- zespół wytwarzający + przyszli użytkownicy.

Jeśli przyjmimy przykład przemysłowej organizacji procesu spiralnego rozwoju oprogramowania z rozwojem przyrostowym i iteracyjnym, to należy utworzyć:

- zespół zapewniający jakość,
- zespół organizacji wielokrotnego wykorzystania oprogramowania (ang. *reuse*),
- zespół między-projektowy,
- wzorcownia – standaryzacja GUI, komunikacja z bazą danych,
- zespół konsultacyjny.

## Praca grupowa

Termin ten określa współpracę wielu, rozproszonych geograficznie, albo siedzących w jednym pomieszczeniu osób, które korzystają z udogodnień oferowanych przez wspomagane technologią środowisko pracy. Wspomaganie to polega na dostępności mechanizmów przesyłania i współdzielenia dokumentów, mechanizmów planowania i rozdzielania zadań, kontroli postępów w realizacji tych zadań, śledzenia terminowości i kosztów prowadzonych prac. Przykładem jest pisanie raportu. Oprócz głównego autora, wpływ na treść dokumentu ma np. recenzent, konsultant, korekta oraz szef zespołu. Podobnie jest przy tworzeniu koncepcji, projektowaniu oraz pisaniu oprogramowania, gdy odpowiednie rezultaty pracy wprowadzane są przez członków poszczególnych wydziałów. Takich sytuacji, w których potrzebna jest współpraca kilku osób, można wymienić jeszcze co najmniej kilka. Okazuje się, że zadania, jakie wykonują pracownicy w przedsiębiorstwach, to często złożona praca grupowa.

Osiągnięcia technologiczne ostatnich lat w zakresie Internetu oraz intranetu, a także w zakresie rozbudowywania możliwości programów biurowych (pakietu biurowego *Office 2000*, *Word*, *Excel*, *Power Point*, *Outlook*) doprowadziły do tego, że dzisiaj przy odrobinie dobrych chęci praktycznie w każdym biurze możliwe jest zastosowanie zasad pracy grupowej wykonywanej w sposób elektroniczny.

Z badań przeprowadzonych wśród użytkowników *Lotus Notes* przez firmę *Andersen Consulting* najczęściej używaną częścią oprogramowania pracy grupowej jest poczta elektroniczna. Rzadziej używane są grupy dyskusyjne, bazy informacyjne, a najmniej – obieg dokumentów (ang. *workflow*). W dalszym ciągu obserwuje się niechęć do używania bardziej zaawansowanych technik poprawy efektywności pracy w zespole.

Narzędzia stosowane do pracy grupowej umożliwiają wcielenie w życie idei pracy zdalnej – telepraca, centralnego zarządzania i kontroli prac grup projektowych, regularnego zabezpieczania ich wyników, usystematyzowanego rejestrowania i korzystania z wiedzy członków zespołu. Praca grupowa **wymusza na członkach zespołu inicjatywę, obowiązkowość i efektywne gospodarowanie czasem i zasobami** i stanowić może element strategii zarządzania wiedzą i budowy kompetencji.

## 5. Zarządzanie ryzykiem

*W życiu pewne jest tylko to, że umrzemy i że musimy płacić podatki.*

B. Franklin

*Gdyby wszystko wokół było pewne, niepotrzebne byłoby zarządzanie.*

KF

Intuicyjnie *ryzyko* oznacza możliwość obniżenia poziomu sukcesu przedsięwzięcia (do kompletnego braku sukcesu włącznie). Tak więc, na podstawie pojęcia ryzyka, należy odwoływać się do skali wartości określającej sukces przedsięwzięcia (a właściwie brak sukcesu) oraz do określonej (niechcianej) sytuacji z tym związanej. Dla poszczególnych udziałowców projektu informatycznego ta niechciana sytuacja może być różnie zdefiniowana. W przedsięwzięciach projektowych występują również zjawiskami, które mają lub mogą mieć wpływ na powodzenie lub negatywne skutki, ale oprócz uświadomienia sobie potencjalnych zagrożeń nie możemy mieć na niego wpływ – nie można nim aktywnie zarządzać – te czynniki lub zjawiska to *niepewność*.

Przytoczymy zatem dwie definicje, jakimi będziemy się stale posługiwać w dalszej analizie zagadnienia:

**Ryzyko** – to możliwość, szansa wystąpienia niebezpieczeństwa, sytuacja niedeterministyczna, w której są określone prawdopodobieństwa wystąpienia przypadków, zarówno pozytywnych, jak i negatywnych. Ryzyko jest zjawiskiem permanentnym.

**Niepewność** – niemożność uzyskania informacji, charakteryzuje się brakiem wpływu na zmianę sytuacji – niepewność nie podlega ocenie za pomocą prawdopodobieństwa i minimalizacji.

Przykładem ryzyka może być np. przekroczenie budżetu projektu czy nie wykonania projektu w terminie. Niepewność to np. pogoda czy zjawiska globalne, które mogą mieć wpływ na nasz projekt – termin dostawy sprzętu komputerowego, a wymienionymi zjawiskami nie możemy zarządzać ponieważ nie mamy na nie wpływu.

Na to jak ważne jest odpowiednie wstępne oszacowanie ryzyka związanego z projektem mogą wskazywać statystyki dotyczące realizowanych projektów informatycznych [44, 50]:



- 47% projektów nie jest nigdy używana (niezgodna z wymaganiami klienta),
- 29% jest zapłacona, ale nigdy nie wytworzona,
- 19 % jest zarzucona lub całkowicie przerobiona na nowo,
- 3% użyte po zmianach,
- 2% użyte bez zmian.

Statystyki wyraźnie pokazują jak niewielki jest odsetek dobrze zrealizowanych projektów informatycznych oraz jak wiele projektów nie udało się zrealizować wcale. Niepowodzenia w realizacji projektu przekładają się zazwyczaj na ogromne straty finansowe bądź to po stronie zamawiającego, bądź wytwórcy. Straty te rocznie szacowane są w miliardach dolarów (w samym tylko USA). Musimy zatem zadać sobie pytanie, czy w ogóle opłaca się podejmowanie realizacji projektów obarczonych ryzykiem. Jak już wspomnieliśmy wcześniej, każdy projekt obciążony jest ryzykiem i niestety tego faktu nie można uniknąć. Gdybyśmy zatem unikali ryzyka, wówczas żadne projekty nie mogłyby być zrealizowane. Jak widać nie tędy droga, gdyż dojdziemy do pewnego paradoksu. Oczywiście należy podejmować się realizacji projektów, gdyż są one chociażby motorem postępu – tworzymy rzecz nową, ale należy starać się w momencie planowania projektu (tworzenia harmonogramu) możliwie najlepiej oszacować ryzyko związane z projektem oraz zaplanować akcje zapobiegawcze – pozwoli to na sprawną realizację projektu zgodnie ze stworzonym planem.

Jak wiemy, z ryzykiem możemy mieć do czynienia dopiero w momencie, kiedy występują elementy niemożności dokładnego szacowania pewnych wartości związanych z otoczeniem projektu oraz jego „udziałowcami” i tak np. może to być:

- Dla klienta – przekroczenie budżetu lub czasu realizacji.
- Dla wykonawcy – odmowa klienta uznania systemu za zakończony.
- Dla użytkownika – zła funkcjonalność, „trudny” interfejs, nieefektywność czasowa, wysoka zawodność.
- Dla instalatora – niska jakość oprogramowania, trudność w dopasowaniu do środowiska docelowego, skomplikowana parametryzacja.

O praktyce zarządzania, zorientowanej na zarządzanie ryzykiem, więcej można znaleźć w pracach [28, 39, 45, 52].

Największym ryzykiem dla projektu jest to, że z opóźnieniem lub w sytuacji, kiedy już występują niekorzystne sytuacje dla projektu, takie jak: przekraczamy terminy, ponosimy większe koszty od zakładanych czy np. użytkownik naszego systemu nie przyjął etapu zrealizowanych dla niego prac – przystępujemy do usuwania zagrożeń. Obrazowo przedstawia to rysunek 5.1 Toba Gilba. To, że takie sytuacje mogą wystąpić prawie w każdym projekcie jest oczywiste, ale przyjęcie tego jako realne, że może to nastąpić w naszym projekcie i że należy wcześniej zastanowić się co będziemy robili, aby do tego nie doszło lub jakie działania przedsięwziąć wcześniej, nie jest powszechnie akceptowalnym działaniem. Wynika to z faktu, że prace „profilaktyczne” zajmują czas, angażują zasoby i kosztują, czyli najczęściej podrażają realizację projektu w sytuacji, gdy wierzymy, że nastąpią wy-

łącznie korzystne przypadki ryzyka. Często wymagane jest scharakteryzowanie elementu ryzyka za pomocą pojedynczej wartości liczbowej, co pozwala na ich bezpośrednie porównywanie i uszeregowanie. Ryzyko jest funkcją, którego atrybuty możemy zdefiniować przez: zdarzenia, prawdopodobieństwo ich wystąpienia oraz konsekwencje – skutki, które mogą nastąpić w wyniku wystąpienia zdarzenia (korzystnego lub niekorzystnego).

*Jeżeli TY nie zaatakujesz ryzyka...*



*..... ryzyko zaatakuje Ciebie!!! – Tob Gilb*



Rys. 5.1. Różnica w podejściu do ryzyka w zależności od fazy realizacji projektu

Definicja podstawowa ryzyka:

$$\text{Ryzyko} = P(z) \cdot S(z)$$

gdzie:

$z$  – element ryzyka,

$P(z)$  – prawdopodobieństwo wystąpienia  $z$ ,

$S(z)$  – miara skutku wystąpienia  $z$ , wyrażana zazwyczaj szacunkowym kosztem lub wartością  $z$  przyjętej skali.

Celem zarządzania ryzykiem jest utrzymanie odpowiedniego stopnia gwarancji odnośnie do sukcesu przedsięwzięcia. Poziom tego ryzyka, który w projekcie jest dopuszczalny, zależy od wielu czynników zarówno zewnętrznych, jak i wewnętrznych. Są projekty, w których różne jego elementy mogą być krytyczne i poziom ryzyka musi być ograniczony do wartości bliskiej zeru. Takim przykładem może być realizacja np. *projektu do obsługi wyborów lub referendum z wykorzystaniem podpisu elektronicznego na dzień 06.06.2004*. Opóźnienie, wydłużenie czasu realizacji projektu np. z 285 dni do 286, kiedy to może znaczyć oddanie w pełni sprawnego systemu informatycznego na dzień 07.06.2004 (opóźnienie tylko o 1 dzień) jest katastrofą dla tego projektu.

Jak wspomniano ryzyko można redukować (minimalizować) do poziomu oczekiwanego (akceptowalnego), a w niektórych przypadkach określone obszary ryzyka eliminować.

Metoda obliczania wpływu redukcji ryzyka RRL (ang. *Risk Reduction Leverage*):

$$RRL = \frac{RVb - RVa}{RRC}$$

gdzie:

$RVb$  – faktyczna wartość ryzyka,

$RVa$  – oczekiwana wartość ryzyka po akcji redukcji ryzyka,

$RRC$  – koszt redukcji ryzyka.

Metoda ta ma bardzo istotne znaczenie, ponieważ w sposób ilościowy – wymierny szacuje koszty związane z redukcją (obniżeniem) poziomu ryzyka.

Każdy projekt ma swoją specyfikę, a z tym związane ograniczenia i trudności oceny oraz zwymiarowania ryzyka. W projektach informatycznych mamy do czynienia z dużą dynamiką zmian technologii, brakiem danych o podobnych przedsięwzięciach, nieporównywalnymi kwalifikacjami oraz kompetencjami zespołu. Do głównych elementów specyfiki projektów informatycznych należą:

### Specyfika oprogramowania

- zdominowanie przez proces projektowania,
- trudności w wizualizacji,

- oprogramowanie nie zużywa się fizycznie tylko moralnie,
- duża złożoność,
- dowolność struktury (twórczość projektanta, programisty),
- zależność elementów,
- brak naturalnych ograniczeń,
- łatwość zmian.

### **Na czym polega specyfika projektów informatycznych?**

- wymiar projektu (czas, budżet i funkcjonalność),
- interdyscyplinarność,
- brak wcześniejszych doświadczeń,
- zmienność (*Phanta rei*),
- rola i znaczenie defektów (*tylko ten się nie myli, kto nic nie robi*),
- ustalenie celów i zakresu systemu,
- intensywność komunikacji.

Nakład pracy związany z rozmiarem projektu nie jest liniowy (patrz rys. 2.5), ponadto wiele prac, które realizuje zespół projektu informatycznego przez dłuższy czas jest niewidoczny. Dopiero pokazanie użytkownikowi projektu w fazie interfejsu (komunikacji z systemem) powoduje weryfikację lub zmianę założeń.

### **Dlaczego nie potrafimy tworzyć systemów informatycznych?**

*...gdybyśmy budowali domy tak, jak tworzymy systemy informatyczne,  
to nie potrafilibyśmy wybudować domu wyższego niż 50 pięter,  
zaś ponad połowa wieżowców o wysokości ponad 20 pięter  
waliliby się zaraz po wybudowaniu.*

Capers Jones – *Applied Software Measurement*

### **Dlaczego projekty się nie udają, co głównie o tym decyduje?**

- Rozmiar projektu przekraczający umiejętności zarządzania.
- Strategia nieadekwatna do rodzaju projektu.
- Niewystarczające wsparcie projektu przez sponsora i kierownictwo firmy wykonawczej.
- Współpraca z klientem niewystarczająca.
- Analiza systemowa powierzchowna.
- Sztynna infrastruktura pracy.
- Jakość ograniczana z uwagi na czas i koszty.

- Planowanie i nadzorowanie projektu.
- Niekonsekwentne zarządzanie zmianami.
- Zarządzanie ludźmi oraz komunikacja.
- Zarządzanie ryzykiem najczęściej bez wcześniejszej analizy i specyfikacji zadań zapobiegawczych.



Rys. 5.2. Czy wystarczy tylko wskazywać na ryzyko

Jak widać na rysunku 5.2 zarządzanie ryzykiem nie może sprowadzać się wyłącznie to uświadamiania, żeby być „ostrożnym” w działaniach. Sam fakt uświadamiania sobie ryzyka, tego co może nas spotkać, może być niewystarczający. Istotne są działania planowe związane ryzykiem, zakomunikowanie ich wyższemu kierownictwu, współdzielenie odpowiedzialności za sukces projektu ze sponsorem, delegowanie uprawnień i inne.

## 5.1. Czynniki mające wpływ na ryzyko

Harmonogramy prac są zwykle bardzo napięte, budżet szczegółowo rozpisany, a wykorzystanie zasobów wysoce zoptymalizowane. Kiedy ryzyko staje się problemem, trzeba nagle znaleźć środki do jego rozwiązania. Odbija się to na terminowości oraz koszcie projektu. Gdyby przeprowadzono wstępną analizę ryzyka, wówczas może udałoby się oszacować jego skutki i zastosować najprostszą strategię jego eliminacji lub ograniczenia przez dodanie do kosztorysu i harmonogramu „marginesu bezpieczeństwa” na obsługę pojawiających się problemów. Oto jakie cechy projektu mogą zadecydować o jego sukcesie lub klęsce.

- Wspólna wizja produktu (systemu)

Powodzenie projektu powinno być wspólną troską wszystkich udziałowców projektu.

- Praca zespołowa

Bardzo ważnym aspektem pracy zespołowej jest zaangażowanie przyszłego użytkownika systemu. Nikt tak jak on nie zna przyszłego środowiska pracy systemu.

- Myślenie przyszłościowe

Zarządzanie ryzykiem koncentruje się na wczesnym wykryciu możliwego ryzyka i działaniach prewencyjnych, nie zaś na usuwaniu skutków powstałych problemów.

- Komunikacja

Wymiana informacji między wszystkimi poziomami powinna być swobodna. Dotyczy to zarówno informacji o rozpoznanym ryzyku, jak i przyjętej strategii jego minimalizacji. Każdy musi być odpowiedzialny za informowanie o wszystkim, co może zakłócić realizację projektu. Rolą kierownika projektu jest stworzenie struktury komunikacyjnej.

- Zintegrowane zarządzanie

Zarządzanie ryzykiem jest częścią zarządzania projektem, dlatego musi być traktowane integralnie z innymi działaniami wspierającymi. Znalezienie wszelkich możliwych źródeł ryzyka i skuteczna minimalizacja jego skutków powinna stanowić jeden z podcelów projektu.

- Ciągłość procesów

Na każdym etapie projektu ryzyko musi być na nowo rozpoznane i oszacowane. Dobrą praktyką jest wpisanie do harmonogramu projektu częstego przeglądu ryzyka projektu.

### **Stopień ryzyka związany z wdrażaniem systemów informatycznych**

Określa się go najczęściej według osiągniętych rezultatów realizacyjnych:

- przerwanie realizacji działań lub fiasko podjętego przedsięwzięcia,
- uzyskanie rezultatów, które nie spełniają wymagań funkcjonalnych czy jakościowych systemu (np. przedłużenie czasu realizacji, zwiększenie kosztów, zawodność funkcjonowania, pominięcie problemów bezpieczeństwa itp.),
- ujawnienie istotnych wad w trakcie eksploatacja systemu.

### **Kategorie ryzyka**

**1. Ryzyko organizacyjne**, które wynika z :

- niejednoznacznie określonych celów i wizji jednostki,
- niewłaściwych relacji w zakresie uprawnień i odpowiedzialności,

- niezdolności do wprowadzeni zmian organizacyjnych,
  - braku właściwych procedur organizacyjnych,
  - niesprawności mechanizmów kontrolnych,
  - nieumiejętności pracy zespołowej,
  - braku określenia celu informatyzacji,
  - niedostatecznej znajomości możliwości i ograniczeń informatycznego wspomaganie zarządzania w przedsiębiorstwie oraz brak zaangażowania jego kierownictwa,
    - nieadekwatność stosowanej technologii informatycznej do rzeczywistych potrzeb i możliwości przedsiębiorstwa,
    - brak zdolności przedsiębiorstwa do zmiany lub niechęć do jej wprowadzenia i zamiar wykorzystania technologii informatycznej jako jej substytutu,
    - niezdolność do przyswajania nowej technologii informatycznej w zakresie przetwarzania danych,
    - brak dostatecznej motywacji przyszłych użytkowników rozwiązania informatycznego,
      - ograniczone zasoby realizacyjne przedsięwzięcia,
      - niedostateczna umiejętności i doświadczenie realizatorów systemu,
      - niesprawna organizacja zespołu projektowo-wdrożeniowego,
      - wykorzystanie odpowiednich metod, technik i narzędzi informatycznych,
      - pojawienie się nowych, nieprzewidzianych czynników w otoczeniu.
2. **Ryzyko psychospołeczne** określane przez:
- niechęć do wprowadzania zmian organizacyjnych,
  - nieumiejętność celowego zastosowania technologii informatycznej,
  - niska kultura informatyczna.
3. **Ryzyko techniczno-technologiczne** określane przez:
- niski poziom w zakresie infrastruktury informatycznej,
  - wykorzystywanie niewłaściwej technologii informatycznej.

Praktyka ignorowania zagrożeń z pierwszej kategorii powoduje, że w krajowych wdrożeniach zintegrowanych systemów informatycznych następuje wzrost kosztów o 200–300% oraz wydłużenie terminów realizacji projektów o 100–200%.

Właściwa ocena ryzyka sprowadza się do jego identyfikacji, a następnie opisu, samo uświadomienie sobie ryzyka nie wystarcza. Analiza musi dotyczyć opisanych zagrożeń – list kontrolnych oraz prognozowanego rozmiaru, skutków jakie dane zagrożenie będzie miało dla projektu oraz w jakiej jego fazie, jak również jakimi symptomami ryzyko będzie się przejawiać. Ważne jest też skupienie się na istotnych zagrożeniach, aby analiza była pomocna w uruchamianiu działań zapobiegawczych.

## 5.2. Identyfikacja ryzyka – metody analizy ryzyka – kwestionariusze, listy kontrolne

Identyfikacja ryzyka może odbywać się według określonych metod, które są preferowane w danej organizacji, która realizuje projekt. Najczęściej wybrane metody analizy są adaptowane do potrzeb i specyfiki projektu, można je podzielić ogólnie na:

- analizę subiektywną,
- analizę wspomaganą listami kontrolnymi i kwestionariuszami,
- analizę wstępującą,
- analizę zstępującą.

### Kwestionariusz identyfikacji ryzyka projektu

Obszar kontaktów z klientem/użytkownikiem

Umowy	Jak jest skonstruowana umowa? Jaka jest wymagana dokumentacja? Jak określono standardy wykonania? Czy dokonano przeglądów umowy?
Wymagania	Czy użytkownik uczestniczył w ustalaniu wymagań?

Obszar środowiska organizacyjnego projektu

System projektowania	Czy dostępna jest wystarczająca liczba stanowisk pracy? Czy system pozwala na realizację wszystkich kroków cyklu życia projektu? Czy organizacja projektu jest efektywna?
----------------------	---

Obszar zarządzania projektami

Zarządzanie	Czy projekt jest realizowany zgodnie z planem? Czy oszacowania są wiarygodne? Jakie jest doświadczenie kierownika projektu?
-------------	---

Obszar inżynierii oprogramowania

Wymagania	Czy wymagania zmieniają się? Jaki ma to wpływ na jakość, funkcjonalność, projekt ...? Czy są wymagania, których nie ma w specyfikacji?
Projekt	Czy są specyficzne, trudne algorytmy do wdrożenia? Czy projekt opiera się na racjonalnych założeniach? Czy zdefiniowano wszystkie interfejsy wewnętrzne i zewnętrzne?
Testowanie:	Czy jest wystarczająco dużo czasu do przeprowadzania wszystkich testów? Czy przygotowano plany testowania? Jakie jest doświadczenie zespołu testującego?



## Obszar działań testujących

Kontrola jakości:	Czy zdefiniowano atrybuty jakości produktu? Czy istnieje system kontroli jakości? Czy prowadzone są zapisy jakościowe?
Szkolenia:	Czy pracownicy mają odpowiednie kwalifikacje? Czy pracownicy przeszli odpowiednie szkolenia?

W tabeli 5.1 wyspecyfikowano czynniki różnicujące analizę ryzyka metodą wstępującą i zstępującą.

Tabela 5.1. Porównani analizy ryzyka zstępującej i wstępującej

Analiza zstępująca	<ul style="list-style-type: none"> <li>• Przeglądanie list kontrolnych, zawierających spis potencjalnych zagrożeń i odnoszenie tych zagrożeń do obecnej sytuacji</li> <li>• Analiza sytuacji bieżącej pozwala ograniczyć liczbę rozpatrywanych zagrożeń</li> <li>• Wykorzystanie list kontrolnych w celu utrzymania właściwego zakresu analiz</li> </ul>
Analiza wstępująca	<ul style="list-style-type: none"> <li>• Ocena sytuacji obecnej i wskazanie możliwych negatywnych skutków w przyszłości</li> <li>• Możliwość wystąpienia przeoczeń oraz wykroczenia poza przewidziany zakres analiz</li> </ul>

### Czynniki ważne podczas tworzenia list kontrolnych

Listy kontrolne są dość powszechnie stosowaną metodą identyfikacji ryzyka. Wiele firm ma własne listy czynników ryzyka (np. firmy konsultingowe, wytwarzające oprogramowanie, wdrożeniowe), kwestionariusz z 194 pytaniami proponuje firma *Software Engineering Institute*, ale są również listy publikowane w literaturze jak np. 60 czynników ryzyka Capersa Jonesa, *Complete List of Schedule Risks* Steve'a McConnella [37]. Listy kontrolne uzupełnione specjalnymi szablonami, są stosowane w trakcie wspólnych sesji identyfikacji ryzyka przez burzę mózgów. Podstawą tworzenia list kontrolnych identyfikacji ryzyka w zakresie procesu wytwórczego oprogramowania są: aktywność, rola, jaką pełnią w projekcie wykonawcy oraz tworzone produkty (artefakty). Po uwzględnieniu jednak aktywności procesu wytwórczego związanego z oprogramowaniem, listę należy rozszerzyć poprzez specyfikację w następujących obszarach:

- czynników charakterystycznych dla efektywności aplikacji,
- czynników ludzkich,
- metod projektowych,
- czynników programowych/sprzętowych,
- czynników zmiany,
- czynników dostawy,
- czynników środowiskowych,
- zabezpieczenia finansowego,
- odpowiedzialności i stabilnej postawy partnera.

Stworzenie pełnej listy kontrolnej stanu początkowego projektu oraz opracowanie formularzy raportowania parametrów stanowiących zidentyfikowane obszary ryzyka to rutynowe elementy zarządzania ryzykiem.

### Inne czynniki ważne podczas tworzenia list kontrolnych

W tej grupie zagadnień należy uwzględnić warunki i środowisko oraz charakterystykę projektu:

- otoczenie socjologiczno-ekonomiczne,
- otoczenie technologiczne,
- organizacja realizacji projektu,
- lista twórców SI,
- projekt.

### Strategie postępowania z ryzykiem

Podstawowe działania, jakie możemy zastosować w przypadku, gdy zidentyfikowaliśmy ryzyko – tworzyliśmy listy:

- obniżenia ryzyka,
- uniknięcia ryzyka,
- transfer ryzyka,
- zaakceptowania ryzyka.

Wyjaśnienia wymaga tzw. *transfer ryzyka*, chodzi o to, aby w projekcie ustanowić właściwe relacje i współodpowiedzialność sponsora za terminowe dostarczenie, np. założeń, analiz itd., aby zamawiający opracowanie systemu informatycznego wyznaczył osobę (zespół), która będzie akceptowała produkty częściowe powstające w trakcie projektu.

## 5.3. Akcje naprawcze

Akcje bezwarunkowe	Jeśli ryzyko może być obniżone przez natychmiastowe działania podejmowane w stosunku do wpływających na nie czynników
Akcje awaryjne	Jeśli poziom ryzyka wymaga śledzenia i jeżeli zaistnieje taka potrzeba podjęcia odpowiednich działań naprawczych

Przygotowanie planu awaryjnego obejmuje:

- określenie istoty potencjalnego problemu,
- rozważenie alternatywnych sposobów rozwiązania problemu,
- określenie ograniczeń, w ramach których problem będzie rozwiązywany,
- analiza alternatywnych rozwiązań,
- wybór jednej z alternatyw.

Plan postępowania awaryjnego zawiera:

- identyfikację zagrożeń, których dotyczy,
- metodę śledzenia ryzyka związanego z tymi zagrożeniami,
- przypisanie odpowiedzialności za śledzenie ryzyka i realizację planu do członków zespołu,
- warunki uruchomienia planu,
- przydział zasobów do wykonania planu,
- ograniczenia związane z opracowaniem planu.

Największym nieporozumieniem w zarządzaniu ryzykiem jest podejście, które polega jedynie na działaniach zmierzających do uniknięcia ryzyka !

## 5.4. Metoda punktowa szacowania ryzyka

*Policz to co można policzyć, zmierz to co można zmierzyć,  
a to co niemierzalne uczynń mierzalnym.*

Galileo Galilei

Wychodząc naprzeciw tezie, że analiza ryzyka musi być mierzalna, wymagane jest scharakteryzowanie elementu ryzyka za pomocą pojedynczej wartości liczbowej, co umożliwi ich bezpośrednie porównywanie i uszeregowanie.

Ryzyko dla większości projektów można wydzielić ze względu na charakter zadań, które realizowane są w projekcie na kategorie. Najbardziej typowe kategorie ryzyka, ich wagi oraz wartości ryzyka przedstawiono w tabelach 5.2 i 5.3.

Tabela 5.2. Kategorie ryzyka

Kategoria	Waga
techniczna	3
organizacyjna	4
finansowa	5
zewnętrzna	4

Tabela 5.3. Wartości ryzyka

Wartość	Znaczenie
1	pomijalne
2	niskie
3	średnie
4	wysokie
5	katastrofalne

Szacowanie ryzyka metodą punktową polega na identyfikacji zadań zagrożonych ryzykiem niepowodzenia realizacji oraz przydzielenie im ilościowej miary poziomu ryzyka według przyjętej skali. Zadania projektu zakwalifikowane jako ryzykowne są grupowane do określonej kategorii, np. organizacyjne. Za pomocą poniższych zależności są definiowane poszczególne wskaźniki ryzyka.

W tabeli 5.2 przydzielono subiektywnie dla poszczególnych kategorii ryzyka wartości wag. Waga ma stanowić ogólną ocenę „ważności” głównego zagrożenie ocenianego nie z poziomu poszczególnych zadań, lecz całego projektu. Wagi wprowadzamy w celu możliwości „przeszacowania” wartości ryzyk w poszczególnych kategoriach oraz całego projektu. To przeszacowanie nazywane jest normalizacją. Do obliczenia wartości ryzyk dla poszczególnych kategorii przed i po podjęciu akcji zapobiegawczych oraz wartości ryzyka całkowitego posługujemy się wzorami:

- ryzyko nieznormalizowane kategorii

$$R_x = \frac{\sum R_v}{n},$$

- ryzyko znormalizowane kategorii

$$R_{zn\_x} = \frac{R_x w_x}{w_{sr}},$$

- ryzyko nieznormalizowane całkowite

$$R_{\text{calk}} = \frac{\sum R_x}{k},$$

- ryzyko znormalizowane całkowite

$$R_{zn\_calk} = \frac{\sum R_{zn\_x}}{k}$$

gdzie:

- $n$  – liczba zadań należących do danej kategorii,
- $R_v$  – wartość ryzyka dla zadania należącego do danej kategorii,
- $k$  – liczba kategorii,
- $w_x$  – waga ryzyka kategorii,

$$w_{sr} \text{ – średnia wartość wagi wyliczana ze wzoru } w_{sr} = \frac{\sum w_x}{k}.$$

Zadaniem normalizacji jest wyrównanie skrajnych i subiektywnych ocen zagrożeń ze strony bezpośrednich wykonawców, kierowników zespołów czy niekiedy samego

kierownika projektu. Powyższe wskaźniki liczymy przed konfliktem po wprowadzeniu akcji zapobiegawczych, aby ocenić czy ograniczyliśmy ryzyko projektu do poziomu akceptowalnego przez sponsora projektu (komitet sterujący).

Jeśli zadania wyspecyfikowane dla określonego projektu mają określony czas trwania, przypisane zasoby do ich realizacji, to kolejnym etapem jest identyfikacja tych zadań, których wykonanie zagrożone jest ryzykiem, następnie ocenić, do jakiej kategorii ryzyka to zadanie należy. Przeanalizowanie ryzyka pod kątem *objawów*, którym będzie się charakteryzowało dane ryzyko, tj. symptomy, jakie mogą przepowiadać, że zagrożenie wystąpiło lub zaczyna faktycznie uaktywniać się w projekcie. *Opis ryzyka* jest związany z tym, czego chcemy uniknąć lub ograniczyć w projekcie. Jeśli zadania np. wprowadzone są w MS Projekt, to możemy kolumnę z kolejnym nr zadania oraz nazwą zadania przenieść do arkusza *Excell*, jak w *przykładzie* tabeli 5.4. W tabeli 5.4 pozostawiamy tylko zadania, które obarczone są ryzykiem z zachowaniem numerów zadań ID zgodnych z projektem głównym, np. pierwszym zadaniem związanym z ryzykiem jest zadanie 80 – *analiza*, to zadanie należy do kategorii *T* – technicznej i ma wartość ryzyka 4 – średnie. Kolejnymi zadaniami będącymi przedmiotem ryzyka to *zadania* 81, 83, 84, 86 aż do *zadania* 155 – szkolenie użytkowników. W tabeli 5.5 wprowadzamy akcje zapobiegawcze minimalizujące (ograniczających) ryzyko zadań. W tej tabeli szacujemy koszty wprowadzonych akcji zapobiegawczych, następnie skutki na zadanie, jak również stopień ograniczenia wartości ryzyka oraz prawdopodobieństwo wystąpienia.

### Przykład

#### Szacowanie ryzyka metodą punktową projektu **WIGGOR**

Tabela 5.4. Zidentyfikowane rodzaje oraz wartości ryzyka dla wybranych zadań

ID	Nazwa zadania	Kat.	Wart.	Objawy	Opis ryzyka
80	Analiza	T	4	Brak informacji lub informacje niekompletne	Niedokładnie lub nieprawidłowo przeprowadzona analiza
81	Modelowanie	T	4	Członkowie stowarzyszenia zgłaszają braki w procedurach	Model nie odzwierciedla rzeczywistości lub jest niekompletny
83	Analiza	T	4	Brak informacji lub informacje niekompletne	Niedokładnie lub nieprawidłowo przeprowadzona analiza
84	Modelowanie	T	4	Członkowie stowarzyszenia zgłaszają braki w procedurach	Model nie odzwierciedla rzeczywistości lub jest niekompletny
86	Analiza	T	4	Brak informacji lub informacje niekompletne	Niedokładnie lub nieprawidłowo przeprowadzona analiza

87	Modelowanie	T	4	Członkowie stowarzyszenia zgłaszają braki w procedurach	Model nie odzwierciedla rzeczywistości lub jest niekompletny
88	Integracja opracowanych procedur	T	3	Brak dokumentu z opracowaniem procedur w terminie	Procedury nie są spójne
92	Przeprowadzenie ankiet wśród studentów	O	5	Brak ankiet w terminie	Mała frekwencja, trudność z zebraniem odpowiedniej próby
94	Wywiady z członkami stowarzyszenia	O	5	Brak wyników rozmów w terminie	Nie można dotrzeć do odpowiednich osób
97	Projekt graficzny	O	4	Brak projektu w terminie	Grafik przydzielony do innych prac zleconych przez stowarzyszenie
98	Projekt funkcjonalny	T	3	Programiści skarżą się, że nie mają wszystkich informacji	Projekcie nie specyfikuje wszystkich zgłoszonych wymagań
99	Projekt bazy danych	T	3	Programiści często proszą o wprowadzanie zmian w schemacie bazy danych	W projekcie bazy danych nie są odzwierciedlone wszystkie związki występujące w rzeczywistości lub brakuje pewnych pól w tabelach
102	Wywiady z członkami stowarzyszenia	O	5	Brak wyników rozmów w terminie	Nie można dotrzeć do odpowiednich osób
104	Projekt graficzny	O	4	Brak projektu w terminie	Grafik przydzielony do innych prac zleconych przez stowarzyszenie
105	Projekt funkcjonalny	T	3	Programiści skarżą się, że nie mają wszystkich informacji	Projekcie nie specyfikuje wszystkich zgłoszonych wymagań
106	Projekt bazy danych	T	3	Programiści często proszą o wprowadzanie zmian w schemacie bazy danych	W projekcie bazy danych nie są odzwierciedlone wszystkie związki występujące w rzeczywistości lub brakuje pewnych pól w tabelach
110	Budowa sieci	F	4	Brak sprzętu	Sponsorzy nie dostarczają sprzętu lub są opóźnienia w przelewach pieniędzy
111	Instalacja serwera	F	5	Brak serwera	Sponsorzy się wycofali lub sprzęt nie dotrze na czas
112	Instalacja stacji roboczych w sieci	F	5	Brak stacji roboczych	Sponsorzy się wycofali lub sprzęt nie dotrze na czas
117	Moduł „Dokumenty”	T	4	Brak modułu na czas	Problem z dołączaniem i pozycjonowaniem zdjęć w obrębie dokumentów
118	Moduł „Aktualności”	T	4	Brak modułu na czas	Nie gotowy moduł dokumentów

121	Moduł „Subskrypcja”	T	4	Brak modułu na czas	Brak dostępu do serwera SMTP
125	Moduł „Rekrutacja	T	4	Brak modułu na czas	Brak dostępu do serwera SMTP
129	Wprowadzanie danych	Z	3	System nie gotowy do testów akceptacyjnych	Brak wszystkich materiałów, które mają być zamieszczone na stronie
133	Szkolenie z CMS	O	4	Opóźnienia w szkoleniu	Trudności z zebraniem ludzi na czas szkolenia
142	Moduł „I-Mail”	T	4	Brak modułu na czas	Brak dostępu do serwera SMTP
144	Moduł „Tablica ogłoszeń	T	3	Brak modułu na czas	Brak dostępu do serwera SMTP
149	Wprowadzenie danych	Z	3	System nie gotowy do testów akceptacyjnych	Brak wszystkich materiałów, które mają być zamieszczone na stronie.
154	Szkolenie administratorów	O	4	Opóźnienia w szkoleniu	Trudności z zebraniem ludzi na czas szkolenia
155	Szkolenie użytkowników	O	5	Opóźnienia w szkoleniu	Trudności z zebraniem ludzi na czas szkolenia

Tabela 5.5. Wprowadzanie akcji zapobiegawczych minimalizujących (ograniczających) ryzyko zadań

ID	Akcja zapobiegawcza	Koszt	Kat.	Wart.	Prawd.	Skutki
80 83 86	Szkolenie wewnętrzne nt. metodologii prowadzenia analizy procesów	1500	T	2	0,7	dokładniejsze wykonanie fazy modelowania, mniej błędów w opracowaniach
81 84 87	Szkolenie wewnętrzne nt. metodologii modelowania procesów	1500	T	3	0,7	dokładniejsze wykonanie fazy modelowania, mniej błędów w opracowaniach
88	2 spotkania w trakcie procesu modelowania	–	T	2	0,25	wykrycie i wyeliminowanie niespójności w trakcie modelowania
92	Przeprowadzenie akcji ankietowej podczas rekrutacji oraz Targowiska	–	Z	3	0,7	Ułatwienie organizacyjne dotarcia do studentów, większa frekwencja
94	Przeprowadzenie ankiet lub burzy mózgów podczas spotkania ogólnego oraz wyjazdu integracyjnego	–	O	3	0,7	Wykonanie wywiadów na czas, bo będzie dostęp do większości członków stowarzyszenia
97	zamówienie projektu na zewnątrz stowarzyszenia	500	T	3	0,7	projekt graficzny na czas
98	weryfikacja projektu przez innego członka zespołu w trakcie jego tworzenia (wewnętrzny audyt)	–	T	3	0,25	wyeliminowanie większości niedopatrzeń

99	weryfikacja projektu przez innego członka zespołu w trakcie jego tworzenia (wewnętrzny audyt)	–	T	2	0,25	wyeliminowanie większości niedopatrzeń
102	Przeprowadzenie ankiet lub burzy mózgów podczas spotkania ogólnego oraz wyjazdu integracyjnego	–	O	3	0,7	Wykonanie wywiadów na czas, bo będzie dostęp do większości członków stowarzyszenia
104	zamówienie projektu na zewnątrz stowarzyszenia	500	T	3	0,4	projekt graficzny na czas
105	weryfikacja projektu przez innego członka zespołu w trakcie jego tworzenia	–	T	3	0,25	wyeliminowanie większości niedopatrzeń
106	weryfikacja projektu przez innego członka zespołu w trakcie jego tworzenia	–	T	2	0,25	wyeliminowanie większości niedopatrzeń
110	Spotkania z dodatkowymi sponsorami (rezerwowe źródło)	100	F	2	0,7	Większa pewność otrzymania sprzętu w terminie
111	Spotkania z dodatkowymi sponsorami (rezerwowe źródło)	100	F	4	0,7	Większa pewność otrzymania sprzętu w terminie
112	Spotkania z dodatkowymi sponsorami (rezerwowe źródło)	100	F	3	0,7	Większa pewność otrzymania sprzętu w terminie
117 118	Dodatkowe testy technologii przed przystąpieniem do prac implementacyjnych	–	T	2	0,25	Zapoznanie się z możliwościami i łatwiejsza implementacja modułów
121	Ustalenie dodatkowego serwera testowego w innym miejscu (np. na uczelni)	–	T	3	0,4	Moduły i testy na czas
125	Ustalenie dodatkowego serwera testowego w innym miejscu (np. na uczelni)	–	T	3	0,4	Moduły i testy na czas
129	Opracowanie części materiałów na wyjeździe strategicznym	–	O	2	0,25	Materiały na czas
133 154 155	2 dniowy wyjazd szkoleniowo rekreacyjny do Srebrnej Góry	1000	O	3 2 3	0,25	szkolenie przeprowadzone w terminie
142	Ustalenie dodatkowego serwera testowego w innym miejscu (np. na uczelni)	–	T	3	0,4	Moduły i testy na czas
144	Ustalenie dodatkowego serwera testowego w innym miejscu (np. na uczelni)	–	T	2	0,4	Moduły i testy na czas
149	Opracowanie części danych (materiałów) na wyjeździe strategicznym		O	2	0,25	Materiały na czas
	<b>SUMA</b>	<b>5300</b>				



Zaproponowane akcje zapobiegawcze zmniejszają ryzyko nieznormalizowane z **3,39** do **2,38**, a znormalizowane z **3,36** do **2,36**. Jest to zmiana o ponad jeden rząd (z ponad średniego ryzyka do ryzyka małego). Koszt dodatkowy, jaki będzie trzeba ponieść z uruchomieniem zadań związanych z akcjami zapobiegawczymi, to **5300 zł**. Należy zauważyć, że większość akcji zapobiegawczych jest związana z inną organizacją pracy lub wykorzystaniem innych działań organizacji mogących ułatwić wykonanie projektu. Te akcje nie pochłaniają dodatkowych środków finansowych a jedynie sposób wykorzystania dostępnych zasobów oraz poszerzenie jakości i częstotliwości kontroli.

Tabela 5.6. Zbiorcze zestawienie ryzyka w poszczególnych kategoriach nieznormalizowane i znormalizowane przed i po akcji zapobiegawczej

Kategoria ryzyka	Waga	Nieznormalizowane		Znormalizowane	
		przed akcją	po akcji	przed akcją	po akcji
Techniczne	3	3,13	2,17	2,02	1,27
Organizacyjne	4	3,86	2,20	3,86	2,48
Finansowe	5	4,00	2,57	5,00	3,75
Zewnętrzne	4	2,57	2,57	2,57	1,93
Całkowite	4	3,39	2,38	3,36	2,36

Zestawienie podstawowych wskaźników zmierzonego ryzyka przedstawia tabela 5.7, w której przedstawione są całkowite wartości ryzyka oraz przewidywane skutki procesu zarządzania ryzykiem projektu WIGGOR.

Tabela 5.7. Szacunek ryzyka projektu WIGGOR

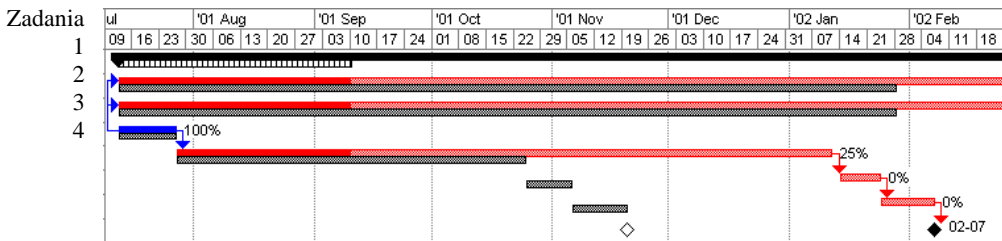
Lp.	Informatyzacja WIGGOR	Wartość (zł)
1	Ryzyko projektu	3,39
2	Ryzyko znormalizowane	3,36
3	Koszt akcji zapobiegawczych	5 300
4	Ryzyko projektu po akcjach zapobiegawczych	2,38
5	Ryzyko znormalizowane po akcjach zapobiegawczych	2,36

Ostateczną decyzję dotyczącą wprowadzenia akcji zapobiegawczych zmniejszających (ograniczających) ryzyko podejmuje PM, a w przypadku gdy projekt ma ograniczony budżet i poziom ryzyka projektu jest akceptowalny, dla sponsora mogą nie być wprowadzane niektóre działania. W powyższym przykładzie 5300 zł stanowi zwiększenie budżetu projektu o ok. 2% (całkowity budżet projektu wynosi 220 000 zł). Jeśli koszty wprowadzenia akcji zapobiegawczej, ograniczających ryzyko stanowiły 10% lub więcej budżetu projektu, to sprawa staje się bardziej złożona i wymaga niejednokrotnie dodatkowych negocjacji ze sponsorem oraz komitetem sterującym projektu.

### Zadanie 5.1

7 września 2003 w dniu kontroli projektu, kierownik programistów poinformował kierownika projektu, że zadanie nr 4 *Implementacja elementów interfejsu* przedłuży się. Jest to dzień, w którym zadanie to powinno być zrealizowane w około 50%. Po oszacowaniu dotychczas zrealizowanych prac kierownik programistów stwierdził, że prace są zrealizowane jedynie w około 25%, co przekazał menedżerowi projektu w postaci raportu. Informacja ta po wprowadzeniu do programu *Microsoft Projekt* pokazała, że projekt jest opóźniony i skłoniła kierownika projektu do przeprowadzenia analizy metodą *Earned Value*.

Wykresy (rys. 3.9) przedstawiają różnice między harmonogramem wcześniej zaplanowanym (szary) a pokazującym obecną sytuację (czerwony). Zadanie zaplanowane było na około 2500 h. Ponieważ w połowie jego realizacji kierownik programistów ocenił, że zadanie jest zrealizowane w około 25%, a ich obecny czas realizacji wynosił będzie około 5000 h. Pozostałe zadania 1, 2 trwają odpowiednio 140 dni, zadanie 3 – 3 dni.



Rys. 3.9. Różnice między harmonogramem planowanym a obecnym  
Zad. 1. Implementacja, Zad. 2. Konsultacje i doradztwo, Zad. 3. Nadzór zadań,  
Zad. 4. Implementacja elementów interfejsu

Oblicz ryzyko dla zadań projektu, rozpisując zadania główne 1–4 do zadań cząstkowych, np. każde zadanie podziel na 2 podzadania. Zaplanuj akcje zapobiegawcze i oblicz ryzyko (całkowite bez normalizacji i znormalizowane) metodą punktową dla tego projektu, zakładając, że zadania 1 i 4 są kategorii ryzyka wewnętrznego, a zadania 2 i 4 należą do kategorii ryzyka zewnętrznego. Waga dla ryzyka wewnętrznego równa się 4, a dla zewnętrznego 1.

Oblicz ryzyko:

1. Dla poszczególnych kategorii.
2. Całkowite podanych zadań.
3. Ryzyko znormalizowane wymienionych zadań.

## 5.5. Metoda PERT szacowania ryzyka

Technika PERT (ang. *Program Evaluation and Review Technique*) została stworzona w celu oszacowania przybliżonych czasów trwania realizacji aktywności/zadań oraz wyznaczenia prawdopodobieństwa zakończenia tych *aktywności/zadań* w żądanym czasie.

Przez *aktywność* należy rozumieć wydzieloną czynność – realizowanych najczęściej przez pojedynczego członka zespołu projektowego. Przyjmuje się, że dekompozycja projektu na aktywności zmierza do realizacji zadań, których realizacja zamykała się w przedziale kilku dni. Dłuższe aktywności mogą zamiennie przechodzić w zadania. Metoda PERT została stworzona na potrzeby kosztownych projektów, których stopień ryzyka był wysoki. Jest bardzo prosta w stosowaniu, a jednocześnie bardzo efektywna [10, 44, 48]. Główny trzon algorytmu szacowania stanowią trzy następujące kroki.

### Algorytm szacowania czasu realizacji projektu

Oszacowanie czasu realizacji pojedynczej aktywności  $t_a$  określa się wzorem:

$$t_a = \frac{a + 4m + b}{6}$$

gdzie:

$m$  – najbardziej prawdopodobny czas wykonania aktywności,

$a$  – optymistyczny, czyli najkrótszy spodziewany czas wykonania aktywności,

$b$  – pesymistyczny, czyli najdłuższy spodziewany czas wykonania aktywności.

Obliczony w ten sposób czas  $t_a$  poszczególnych aktywności wykorzystuje się do obliczania czasu trwania projektu i wyznaczania jego ścieżki krytycznej.

**Obliczenie odchylenia standardowego aktywności  $s$**  jest miarą stopnia niepewności oszacowania czasu  $t_e$  trwania aktywności i dane jest wzorem:

$$s = \frac{b - a}{6}.$$

Może być stosowane jako miara porównawcza stopnia niepewności lub ryzyka każdej aktywności.

**Wyznaczenie prawdopodobieństwa osiągnięcia celów** zakończenia (właściwie niezakończenia) danego zadania w ustalonym czasie  $T$ , należy:

a) Obliczyć czas trwania zadania. Jeżeli na dane zadanie składa się kilka aktywności, które wykonywane są jednocześnie, za czas realizacji zadania przyjmuje się czas najdłuższej aktywności (patrz przykład).

b) Obliczyć standardowe odchylenia zadania. Jeżeli na dane zadanie składa się kilka aktywności, które są wykonywane jednocześnie, standardowe odchylenie obliczane jest na podstawie najdłuższej aktywności (tej, która posłużyła do obliczenia czasu w

punkcie a). Jeżeli tą aktywność poprzedza inne zadanie, standardowe odchylenie zadania końcowego obliczane jest z wzoru:

$$s = \sqrt{s_{\text{zad\_poprz}}^2 + s_{\text{akt}}^2}.$$

c) Wyznaczyć dla zadania wartość współczynnika  $z$  ze wzoru:

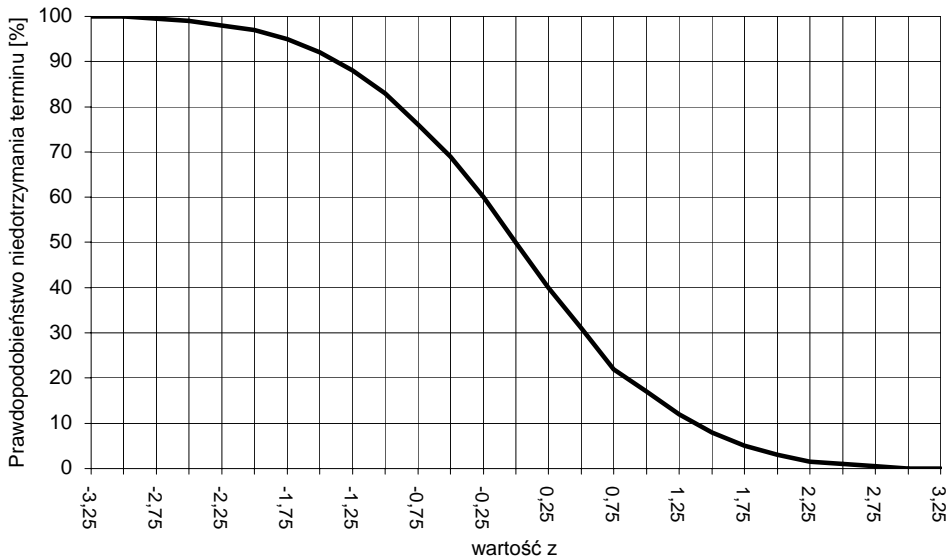
$$z = \frac{T - t}{s}$$

gdzie:

$T$  – żądana data docelowa zakończenia zadania,

$t$  – czas oszacowany w punkcie a).

d) Odwzorować wartość  $z$  na prawdopodobieństwo, korzystając z „odpowiednich krzywych” zamieszczonych w np. tablicach matematycznych. Krzywa taka może być przedstawiona na rysunku 5.3.



Rys. 5.3. Odwzorowanie prawdopodobieństwa niedotrzymania terminu w zależności od wartości  $z$

### Zalety metody PERT

1. Ustanawiając daty docelowe zadań na ścieżce krytycznej, zwraca się szczególną uwagę na te zadania, które wprowadzą do projektu pewne opóźnienia.

2. Obliczenie standardowego odchylenia zadania i porównanie go ze stopniem ryzyka każdego zadania pozwoli na wyłonienie tych zadań, które wymagają „szczególnej opieki”.

### Przykład

Projekt **SEZAM** składa się z ośmiu aktywności. Oznaczmy je literami A–H. Załóżmy, że eksperci na podstawie swojego doświadczenia i analizy projektu wyznaczili czas realizacji poszczególnych aktywności w następujący sposób (patrz tabela 5.8).

Tabela 5.8. Czas trwania poszczególnych aktywności

Aktywność	Czas trwania aktywności [tygodnie]		
	Optymistyczny (a)	Najbardziej prawdopodobny (m)	Pesymistyczny (b)
A	5	6	8
B	3	4	5
C	2	3	3
D	3,5	4	5
E	1	3	4
F	8	10	15
G	2	3	4
H	2	2	2,5

W przypadku określenia czasu pesymistycznego bierze się pod uwagę wszelkie niepomyślnie zdarzenia, które mogą wystąpić w czasie realizacji projektu. Kalkulacje tego czasu może uwzględniać konieczność uruchomienia akcji zapobiegawczej. Jeśli chodzi o czas optymistyczny, to zakłada się, że czynniki wpływające negatywnie na wykonanie zadania nie wystąpią lub nie spowodują poważniejszych zmian w projekcie, a przede wszystkim obciążenia czasowego.

Dla każdej aktywności obliczamy oczekiwany czas trwania  $t$  oraz odchylenie standardowe  $s$  przedstawione w tabeli 5.9. Po obliczeniu widzimy, że oczekiwany czas aktywności A wynosi 6,17 tygodni, a odchylenie standardowe 0,5, aktywność B – odpowiednio 4,00 i 0,33 itd.

Tabela 5.9. Oczekiwany czas trwania oraz odchylenia standardowe poszczególnych aktywności

Aktywność	Oczekiwany czas trwania [tygodnie]	Standardowe odchylenie (s)
A	6,17	0,50
B	4,00	0,33
C	2,83	0,17
D	4,08	0,25
E	2,83	0,50
F	10,5	1,17
G	3,00	0,33
H	2,08	0,08

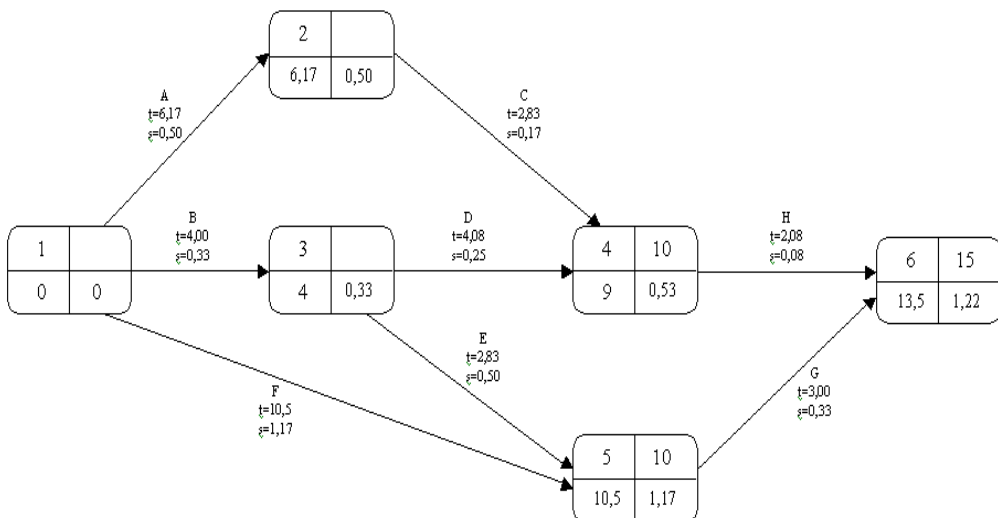
Nasz projekt, zgodnie z grafem zamieszczonym poniżej, składa się z 6 zadań. Dodatkowo wiemy, że zadanie 4 musi zakończyć się po 10 tygodniach, gdyż pracownicy zaangażowani w realizację aktywności C (poprzedzającej zadanie) odchodzą po tym czasie do innego projektu. Zadanie 5 także musi zakończyć się po 10 tygodniach, gdyż po tym czasie zobowiązaliśmy się pokazać klientowi prototyp systemu. Zakończenie projektu – zadanie 6 – planowane jest na 15 tygodni.

Zauważmy, że czas trwania aktywności  $F$  ( $t = 10,5$ ) jest dłuży niż suma czasu aktywności  $B$  i  $E$  ( $t = 4,00 + 2,83 = 6,83$ ), dlatego czas realizacji zadania 5 wynosi  $t = 10,5$ .

Przewidywany czas realizacji zadania 6 to  $t = 10,5 + 3,00 = 13,5$ , a standardowe odchylenie  $s$  wynosi:

$$s = \sqrt{1,17^2 + 0,33^2} = 1,22$$

Przedstawiony graf sieci PERT ukazuje powiązanie między aktywnościami oraz zawiera obliczone poszczególne wartości. Graf składa się z węzłów, które opisują cztery parametry zadania (czas realizacji zadania, na które składają się aktywność, odchylenie standardowe, numer zadania oraz wymagany termin zakończenia zadania) oraz łuków, którymi są aktywności opisane trzema parametrami (nazwa aktywności, oczekiwany czas trwania oraz odchylenie standardowe).



Rys. 5.4. Graf sieci PERT z uwzględnieniem czasu i obliczonym standardowym odchyleniem

Ostatnim krokiem, zgodnie z algorytmem prezentowanym w punkcie 6.3.1 jest obliczenie wartości  $z$  i odwzorowanie jej na prawdopodobieństwo niedotrzymania terminów poszczególnych zadań. Obliczenia zebraliśmy w tabeli 6.10, która umożliwi na

podstawie odczytania z wykresu (rys. 5.3) wartości prawdopodobieństwa niedotrzymania terminów realizacji poszczególnych zadań w zależności od wartości współczynnika  $z$ .

Tabela 5.10. Wyznaczenie prawdopodobieństwa niedotrzymania terminów realizacji poszczególnych zadań

Zadanie	Docelowa data ukończenia [tygodnie]	Wartość $z$	Prawdopodobieństwo porażki [%]
1	–	–	–
2	–	–	–
3	–	–	–
4	10	1,887	4
5	10	-0,427	68
6	15	1,231	11

Widać, że zadanie 5 ma największe prawdopodobieństwo przekroczenia czasu realizacji i wynosi ono aż 68%. Jest to zgodne z naszymi oczekiwaniami, gdyż oszacowany czas realizacji ( $t = 10,5$  tygodnia) był większy od ustalonego z klientem ( $T = 10$  tygodni – data pokazania prototypu sytemu).

## 6. Projekty wdrożeniowe – *outsourcing*

Wdrożenie w ramach projektu informatycznego można zdefiniować jako: przedsięwzięcie mające na celu stworzenie unikatowej usługi lub produktu, wykorzystujące rozwiązania informatyczne, oparte na technologii komputerowej oraz wprowadzające korzyści biznesowe, poprawę funkcjonalności lub osiągnięcia nowej jakości w obszarze jego zastosowania. Przez rozwiązania informatyczne rozumiemy metody komputerowego wspomagania przebiegu różnorodnych procesów zarządczych, ekonomicznych, informacyjnych w firmie, a pod pojęciem technologii komputerowej rozumie się wszystkie aspekty techniczne takich rozwiązań (np. sprzęt, oprogramowanie).

### 6.1. Rodzaje projektów informatycznych oraz organizacja pracy i zespołów

**Podział projektów informatycznych ze względu na to, jaki obszar informatyki obejmuje projekt i czy projekt realizuje zupełnie nowe rozwiązania, czy też wprowadza nowe elementy do już istniejących rozwiązań, jest następujący:**

**nowe** – podejmowane przedsięwzięcie ma zupełnie nowy charakter dla organizacji, tj. nie funkcjonują w niej systemy informatyczne,

**uzupełniające** – realizowane przedsięwzięcie wnosi nowe elementy do już istniejących rozwiązań (np. rozbudowa sieci komputerowej),

**programowe** – projekt dotyczy wdrożenia nowego typu oprogramowania przy istniejących rozwiązaniach sprzętowych (np. budowa bazy danych klientów),

**sprzętowe** – w wyniku projektu następuje modyfikacja stosowanych rozwiązań sprzętowych (np. wymiana stacji roboczych na nowsze),

**kompleksowe** – łączące w sobie projekty sprzętowe i programowe (np. projekt komputeryzacji firmy od podstaw).

Takie przyporządkowanie umożliwi zorientowanie się, jak bardzo skomplikowany może być projekt oraz jaką przyjąć strategię jego realizacji. Zasadniczo najbardziej skomplikowane będą projekty nowe i kompleksowe, gdyż dotyczyć będą czegoś, co do tej pory nie było realizowane. Prostsze natomiast będą projekty uzupełniające, sprzętowe i programowe, gdyż zawsze będą opierały się na już istniejących rozwiąza-



niach. Rozpoznanie typu projektu może być istotne z tego względu, że z niektórymi rodzajami projektów firma chcąc je zrealizować, może sobie nie poradzić i wskazane wtedy będzie posłużenie się fachową pomocą firm zewnętrznych.

Szczególną właściwość mają tzw. **projekty wdrożeniowe**, gdzie o sukcesie takiego projektu w bardzo dużym stopniu decydują czynniki i kwalifikacje socjotechniczne kierownika projektu.

Aby wdrożenie zakończyło się sukcesem, należy:

- pozyskać zaangażowanie kierownika firmy, w której wdrażamy system (współdzielenie się odpowiedzialnością),
- zapewnić niezbędne zasoby ludzkie firmy,
- przyjąć zasady stopniowego rozwoju,
- zapewnić elastyczność w doborze parametrów projektu,
- uwzględnić aspekty socjotechniczne związane z mentalnością i nawykami użytkowników,
- zaplanować (wybrać) lidera procesu wdrożeniowego.

Są to najważniejsze czynniki, o których należy pamiętać, ale nie jedyne, ponieważ jakość oraz niezawodność produktu, jak również przygotowanie organizacyjne użytkownika do procesu restrukturyzacji, w którym system informatyczny wspomaga określone procesy, stanowi o zintegrowaniu procesu wdrożeniowego.

Zmiany organizacyjne w jednostce są czasami konieczne, aby nastąpiło wdrożenie ponieważ:

- systemy zarządzania wewnątrz organizacji nie są przystosowane do realizacji projektów,
- powodzenie projektu zależy w takim samym stopniu od firmy, w której projekt ma być wdrożony, jak i od innych organizacji realizujących projekt,
- lider wdrożenia potrzebuje pomocy ze strony kierownictwa działów firmy–klienta.

Struktura zespołu wdrożeniowego powinna bazować na:

- komitecie sterującym – odpowiedzialnym za strategiczne zarządzanie całym przedsięwzięciem,
- komitecie wykonawczym – odpowiedzialnym za taktyczne zarządzanie całym przedsięwzięciem.

Komitec sterujący najczęściej stanowią:

- sponsor przedsięwzięcia,
- główny kierownik przedsięwzięcia ze strony producenta systemu,
- lider wdrożenia ze strony informatyzowanego przedsiębiorstwa,
- konsultanci zewnętrzni.

Zadania komitetu sterującego obejmują realizację takich prac, jak:

- określenie celów i zdefiniowanie pojęcia „wdrożenie systemu”,
- przeprowadzenie analizy przedsiębiorstwa oraz określenie budżetu wdrożenia,

- wybór dostawcy oprogramowania oraz firmy doradczej,
- zapewnienie aktywnego udziału naczelnego kierownictwa w pracy zespołu wdrożeniowego,
- przygotowanie harmonogramu wdrożenia,
- powołanie komitetu wykonawczego,
- weryfikacja wykonywanych działań,
- przekazanie systemu do eksploatacji.

Komitet wykonawczy to zespół operacyjny projektu, w skład którego wchodzi:

- główny kierownik przedsięwzięcia ze strony dostawcy systemu,
- lider wdrożenia ze strony klienta,
- kierownicy dziedzinowi,
- użytkownicy kluczowi.

Do zadań komitetu wykonawczego należą:

- podział prac i odpowiedzialności w zespole,
- bieżące zarządzanie realizacją prac wdrożeniowych,
- prowadzenie dokumentacji wdrożeniowej,
- nadzorowanie prowadzonych prac, monitorowanie ich wydajności, sporządzanie okresowych raportów,
- inicjowanie działań korygujących w przypadku wystąpienia zagrożeń realizacyjnych,
- powoływanie oraz bieżące koordynowanie pracy zespołów wdrożeniowych.

Czynności etapu wdrożenia obejmują między innymi:

- zakup sprzętu,
- instalację bądź rozwój sieci komputerowej,
- zainicjowanie bazy danych,
- wprowadzenie do niej danych,
- opracowanie i testowanie programów,
- zakup pakietów oprogramowania,
- przygotowanie dokumentacji systemu,
- przeszkolenie jego użytkowników.

Prace programistyczne można przyspieszyć przez:

- wykorzystanie zasad prototypowania systemu,
- wykorzystanie języków czwartej generacji,
- generatory kodu,
- zakupienie i wdrożenie zintegrowanego pakietu oprogramowania.

Kategorie testów oprogramowania:

- indywidualne, dotyczące sprawdzenia poprawności poszczególnych modułów

- oprogramowania,
- zintegrowane, zwane testami akceptacyjnymi, związane z kontrolą i korektą całości oprogramowania systemu, będące podstawą harmonijnego współdziałania wszystkich jego modułów.

**Jakość według ISO 9000** – ogół cech i właściwości wyrobu lub usługi, decydujących o zdolności wyrobu lub usługi do zaspokojenia stwierdzonych lub przewidywanych potrzeb.

#### **Kryteria jakości – model McCalla**

Działanie programu – przyjazność, wydajność, poprawność, niezawodność.

Przystosowanie do modyfikacji – pielęgnowalność, elastyczność, testowalność.

Mobilność oprogramowania – przejrzystość, uniwersalność, otwartość.

#### **Strategie wdrażania systemu**

- krokowe (ang. *step-by-step*),
- wdrażanie wszystkich modułów jednocześnie (ang. *big-bang*),
- wdrażanie głównych modułów jednocześnie (ang. *middle-size big bang*).

#### **Wdrożenie istniejącego systemu**

- wymiana interfejsów na przyjazne,
- wymiana platformy sprzętowej,
- wymiana oprogramowania użytkowego,
- wprowadzenie architektury rozproszonej,
- restrukturyzacja.

#### **Wymiana oprogramowania aplikacji**

- konwersja bezpośrednia – natychmiastowe zastąpienie systemu,
- konwersja równoległa – nowy i stary funkcjonują jednocześnie, aż nowy będzie w pełni niezawodny i stabilny,
- konwersja pilotowa – tylko część użytkowników wykorzystuje nowy system,
- konwersja fazowa – etapowe wprowadzenie nowego systemu poprzez sukcesywne instalowanie poszczególnych modułów zastępujących dotychczas użytkowane.

## **6.2. Wdrożenie przez *outsourcing***

*Outsourcing to sprzedaż – kupno niematerialnych usług informatycznych; sposób świadczenia usług dotyczących zarządzania, eksploatacji i utrzymania części albo całości systemu informatycznego, polegający na powierzeniu tych czynności specjalistycznej firmie usługowej na ściśle określony czas [17, 18, 19].*

Alternatywą dla tradycyjnych metod wdrażania systemów, która zdobywa sobie ostatnio coraz większą popularność jest *outsourcing*. Firma wdrażająca system na

zasadzie *outsourcingu* zarządza nim na poziomie administracji sprzętowej i aplikacyjnej oraz nadzoruje pracę sieci rozległej. Firma odpowiedzialna jest również za infrastrukturę oraz komunikację zewnętrzną. Serwery są najczęściej przenoszone do centrum obliczeniowego firmy wdrażającej, co uwalnia komputeryzowaną firmę od troski o bezpieczeństwo systemu informatycznego. Informatyzowany klient nie musi także zatrudniać dodatkowych pracowników, którzy byliby odpowiedzialni za administrowanie systemem. W firmie zainstalowane są tylko komputery użytkowników końcowych, przetwarzanie danych odbywa się w centrum obliczeniowym firmy *outsourcingowej*, zdalne jest także administrowanie całym systemem. Szkoleniom poddawani są tylko użytkownicy końcowi.

Podsumowując – jest to względnie szybki i bezpieczny sposób wdrożenia zintegrowanego systemu informatycznego.

Porównanie tradycyjnego podejścia do zakupu i wdrożenia systemu informatycznego i *outsourcingu* zawarto w tabeli 6.1.

Tabela 6.1. Porównanie tradycyjnej metody zakupu oprogramowania z *outsourcingiem*

	Metoda tradycyjna	Outsourcing
Sieć komputerowa	Tak	Częściowa
Sprzęt komputerowy	Zakup	Dzierżawa; opłaty zależne od klasy sprzętu
Wdrożenie	Tak	Tak
Opłata za łącza TP SA	Brak	Wymagana
Oprogramowanie aplikacyjne	Zakup	Dzierżawa; opłaty zależne od rodzaju oprogramowania i efektywnego czasu jego wykorzystywania (biling)
Oprogramowanie narzędziowe	Zakup	Opłaty wliczone w opłaty za oprogramowanie
Instalacja i konfiguracja oprogramowanie	Wymagane	Opłaty wliczone w opłaty za oprogramowanie i sprzęt
Koszt utrzymania personelu informatycznego	Wymagane	Nie

### Podział *outsourcingu*

*Outsourcing* jest pojęciem dość rozległym, dotyczącym różnych dziedzin działalności gospodarczej i biznesowej. W informatyce rozwinął się i dotyczy nie tylko systemów informatycznych i ma różne odmiany. Klasyfikacja tego zagadnienia pozwala uniknąć nieporozumień oraz ułatwia zdefiniowanie potrzeb i możliwości klienta.

Podział według stawianego celu

- **taktyczny** – podyktowany jest bieżącymi potrzebami firmy koncentruje się zwykle na jednym aspekcie działalności (np. termin realizacji), przeważnie ograni-

czony do części systemu (np. zarządzanie siecią LAN, WAN), świadczony przez stosunkowo krótki okres; może wynikać z bieżących kłopotów przedsiębiorstwa (np. brak możliwości zatrudnienia wykwalifikowanych pracowników),

- **strategiczny** – element strategii biznesowej przedsiębiorstwa pomyślany jest jako pełny transfer usług, zarządzania i odpowiedzialności, ma długotrwały charakter, cechuje się partnerskimi relacjami między firmami; wynika z przemyślanej strategii przedsiębiorstwa (koncentrowanie się na podstawowej działalności firmy – ang. *core business*).

Podział według zakresu działania

- **totalny** – polega na przekazaniu jakiejś całej dziedziny działalności biznesowej w ręce firmy usługowej (np. przekazanie systemów informatycznych do centrum przetwarzania danych),
- **selektywny** – przekazuje się fragment działalności w obrębie danej dziedziny (np. zarządzanie siecią rozległą).

Podział według rodzaju (dot. systemów informatycznych)

- **outsourcing przetwarzania danych** – udostępnianie mocy obliczeniowej zewnętrznej bez względu na rodzaj aplikacji,
- **outsourcing systemów informatycznych** – udostępnianie określonego systemu informatycznego wraz zapewnieniem jego informatycznej obsługi,
- **outsourcing procesów biznesowych** – przekazanie działalności całego działu do firmy usługowej (np. naliczanie płac, rozliczeń z kasą chorych).

### Integrator wdrożenia

Podstawowym zadaniem integratora wdrożenia jest zagwarantowanie powodzenia w realizacji wdrożenia systemu przez:

- prawidłowe planowanie i harmonogramowanie prac,
- sprawne koordynowanie działań wszystkich uczestników przedsięwzięcia,
- rygorystyczne przestrzeganie terminów i budżetu przy spełnianiu wymogów jakościowych.

Firma taka bierze odpowiedzialność za końcowy rezultat prac projektowo-wdrożeniowych, jakim jest efektywne funkcjonowanie systemu informatycznego.

### Przykład modelu integracji

Faza 1 – analiza przedsiębiorstwa, określenie celów strategicznych przedsięwzięcia, wymagania, kryteria wyboru Zintegrowanego Systemu Informatycznego (ZSI):

- udokumentowanie istniejących procedur działania,
- opracowanie opisów procesów, przygotowanie koniecznej restrukturyzacji przedsiębiorstwa,

- ocena skali przedsięwzięcia, ryzyka, kosztów i terminów.
- Faza 2 – opracowanie koncepcji informatyzacji przedsięwzięcia:
  - selekcja i wybór gotowego systemu informacji,
  - konfigurowanie oprogramowania aplikacyjnego według modelu prototypowania,
  - modelowanie konfiguracji oprogramowania.
- Faza 3 – realizacja systemu obejmująca:
  - przeprowadzenie koniecznej restrukturyzacji przedsiębiorstwa,
  - organizację dostaw sprzętu i oprogramowania,
  - instalację i uruchomienie oprogramowania,
  - działalność szkoleniowo-edukacyjną,
  - szkolenia użytkowników,
  - wdrożenie i testowanie.
- Faza 4 – konserwacja i bieżący rozwój systemu:
  - umowy na długoterminową współpracę w ramach nadzoru autorskiego (wykonawczego),
  - konieczne modyfikacje systemu, wynikające ze zmian obowiązujących przepisów,
  - rozbudowa oprogramowania i sprzętu, wynikająca z rosnących wymagań użytkownika,
  - stały rozwój i dostosowywanie.

### **Formy szkoleń**

Każde wdrożenie systemu informatycznego musi zawierać efektywną formę szkolenia użytkowników systemu. Są różne metody i formy szkolenia np.:

- szkolenia prowadzone w zorganizowanych ośrodkach szkoleniowych wyposażonych w komputery połączone siecią,
- szkolenia prowadzone na miejscu u klienta z wykorzystaniem miejscowego sprzętu i oprogramowania,
- szkolenia przez sieć (Internet).
- inne.

Nawet najlepiej zbudowany i sprawdzony system informatyczny może przynieść złą sławę wykonawcy, jeśli w niewłaściwy sposób przygotowują odbiorcę systemu do jego użytkowania. W tej klasie projektów szczególną uwagę należy zwrócić na zarządzanie ryzykiem głównie do zadań w kategoriach zewnętrznych i organizacyjnych.

## 7. Czynniki sukcesów i niepowodzeń projektów

Omawiane wcześniej główne czynniki, które są przyczyną niepowodzeń projektów informatycznych, wskazano między innymi na takie jego parametry, jak rozmiar-wielkość, która powoduje, że przekracza on umiejętności niezbędne do jego zarządzania. Wybór lub brak wyboru adekwatnej strategii realizacji, niewystarczające wsparcie projektu przez sponsora i kierownictwo firmy wykonawczej, niewystarczająca współpraca z klientem, zbyt ogólna analiza systemowa. Niewłaściwy harmonogram i alokacja zasobów, powodująca konieczność wykonania prac kosztem ich jakości, utrata kontroli nad zarządzaniem zmianami i inne. W tym rozdziale przedstawione zostaną wyniki badań *The Standish Group*, dotyczące najczęstszych przyczyn niepowodzeń projektów.

### 7.1. Niektóre badania statystyczne niepowodzeń projektu

O tym jak ważne jest planowanie w osiągnięciu sukcesu całego projektu mówi dokument „Chaos” stworzony w 1995 roku przez *Standish Group* [53], choć od 1995 r. upłynęło kilka lat, to „aktualność” tych danych co do specyfikacji niektórych zależności w dalszym ciągu jest pouczająca. Dostęp do najnowszych danych jest płatny i zainteresowanych odsyłam do zacytowanej strony WWW tej organizacji. Badania przeprowadzone przez *The Standish Group* w Stanach Zjednoczonych były przeprowadzone w firmach IT. Wyniki badań opierają się na wywiadach z ludźmi uczestniczącymi w tworzeniu projektów.

W badaniach były brane pod uwagę małe, średnie oraz duże przedsiębiorstwa (działające w Bostonie i San Francisco), począwszy od dużych przemysłowych organizacji, np. bankowość, bezpieczeństwo, sprzedaż detaliczna, do lokalnych, federalnych organizacji. W sumie przebadano 365 firm, wzięto pod uwagę 8380 aplikacji. Dla uzyskania poprawnych, rzetelnych wyników *Standish Group* przeprowadziło wiele wywiadów, zadało wiele pytań.

Według badań wynika, że tylko 16,2% projektów kończy się sukcesem, podczas gdy 52,7% kończy się niepełnym sukcesem, a 31,1% zostaje anulowane. Projekty

zakończone pełnym sukcesem to takie, które są kompletne, nie przekroczyły czasu ani budżetu. Projekty zakończone niepełnym sukcesem są kompletne, ale został przekroczony czas i budżet, oraz kilka funkcji i założeń pierwotnych zostało zmienionych w trakcie realizacji projektu. Projekty anulowane nie zostały ukończone.

Wyniki badań były pesymistyczne niezależnie od wielkości firmy. Tylko 9% projektów w wielkich przedsiębiorstwach zakończyło się sukcesem, w średnich 16,2%, w małych 28%. Większość z projektów wielkich firm zakończyła się sukcesem niepełnym, natomiast średnich i małych przedsiębiorstw odpowiednio: 46,7% i 50,4%. Projekty, które zostały anulowane stanowią 37,1% projektów w średnich firmach, 29,5% w wielkich, a 21,6% w małych.

Głównymi przyczynami niepowodzenia projektu jest przekroczenie budżetu i czasu realizacji projektu. Najczęściej jest to związane z nieprawidłowym planowaniem projektu.

Średnie przekroczenie budżetu dla wszystkich przedsiębiorstw wynosi 189% szacowanego budżetu; dla wielkich przedsiębiorstw: 178%, dla średnich 182% i 214% dla małych.

Tabela 7.1

Przekroczenie budżetu (w procentach)	Liczba projektów (w procentach)
Poniżej 20%	15,5%
21–50%	31,5%
51–100%	29,6%
101–200%	10,2%
201–400%	8,8%
Ponad 400%	4,4%

Średnie przekroczenie czasu stanowi 222% szacowanego czasu realizacji projektu. Dla wielkich przedsiębiorstw wynosi 230%, średnich 202% i małych 239% (tabela 7.2).

Tabela 7.2

Przekroczenie czasu (w procentach)	Liczba projektów (w procentach)
Poniżej 20%	13,9%
21–50%	18,3%
51–100%	20,0%
101–200%	35,5%
201–400%	11,2%
Ponad 400%	1,1%

O pełnym sukcesie projektu decyduje niewątpliwie zgodność produktu końcowego z założeniami, powstałymi przed realizacją projektu. Dla dużych przedsiębiorstw tylko



42% produktu końcowego zgadzała się z założeniami, dla średnich przedsiębiorstw sytuacja przedstawia się trochę lepiej – 65%, a dla małych 74% (tabela 7.3).

Tabela 7.3

Zgodność produktu końcowego z założeniami (w procentach)	Liczba projektów (w procentach)
Mniej niż 25%	4,6%
25–49%	27,2%
50–74%	21,8%
75–99%	39,1%
100%	7,3%

Z badań przeprowadzonych w 365 firmach na 3682 projektach wynika, że tylko 431, czyli 12% z tych projektów zostało zakończonych na czas i nie przekroczyło budżetu. Celem *Standish Group* było znalezienie przyczyny niepowodzenia projektów. W tym celu zapytała ankietowanych, jakie według nich czynniki wpływają na sukces projektu. Na czwartym miejscu znalazło się prawidłowe planowanie. Oznacza to, że projektanci docenili rolę planowania (tabela 7.4).

Tabela 7.4

Lp.	Czynniki sukcesu projektu	Liczba projektów (w procentach)
1	Zaangażowanie klienta	15,9%
2	Wsparcie kierownictwa	13,9%
3	Jasne określenie wymagań	13,0%
4	Właściwe planowanie	9,6%
5	Realistyczne oczekiwania	8,2%
6	Mniejsze odstępstwa między punktami kontrolnymi	7,7%
7	Kompetencje pracowników	7,2%
8	Odpowiedzialność	5,3%
9	Jasno sprecyzowane cele	2,9%
10	Ciężko pracujący pracownicy	2,4%
11	Inne	13,9%

Dla lepszego zrozumienia niepowodzeń projektów *Standish Group* szczegółowo przeanalizowało cztery słynne projekty: dwa z nich zakończyły się pełnym sukcesem (Hyatt Hotels, Banco Itamarati), a dwa pozostałe poniosły klęskę (California DMV, American Airlines). Rozpatrując powyższe przykłady pod względem czynników sukcesu projektu wynika, że właściwe planowanie odgrywa decydującą rolę w sukcesie projektu. Projekty, które odniosły sukces charakteryzowały się skrupulatnym planem, natomiast w projektach, które zakończyły się klęską proces planowania został zaniedbany.

Projekty zakończone pełnym sukcesem: Hyatt Hotels, Banco Itamarati.

Projekty zakończone klęską: California DMV, American Airlines.

Tabela 7.5

Lp.	Czynniki sukcesu projektu	California DMV	American Airlines	Hyatt Hotels	Banco Itamarati
1	Zaangażowanie klienta	nie	nie	tak	tak
2	Wsparcie kierownictwa	nie	tak	tak	tak
3	Jasne określenie wymagań	nie	nie	tak	nie
4	Właściwe planowanie	nie	nie	tak	tak
5	Realistyczne oczekiwania	tak	tak	tak	tak
6	Mniejsze odstępstwa między punktami kontrolnymi	nie	nie	tak	tak
7	Kompetencje pracowników	nie	nie	tak	tak
8	Odpowiedzialność	nie	nie	tak	tak
9	Jasno sprecyzowane cele	nie	nie	tak	tak
10	Ciężko pracujący pracownicy	nie	nie	tak	tak
11	Inne	nie	tak	tak	tak

Z przeprowadzonych badań przez *Standish Group* w roku 2003 wynika, że zwiększa się liczba projektów zakończonych sukcesem z 16% w 1994 r. do 34% obecnie. Innym pozytywnym zjawiskiem jest zmniejszanie się kosztów przekroczenia budżetu projektu z ok. 180% w połowie lat 90. do ok. 43% obecnie. Największy postęp w zakresie prowadzenia projektów co do ich wskaźnika projektów zakończonych sukcesem oraz kosztów wytwarzania odnotowano w dużych firmach. W małych firmach natomiast zauważono dość znaczący (50%) wzrost kosztów przy niewielkiej poprawie wskaźnika projektów zakończonych sukcesem.

*Standish Group* stwierdza, że istnieją trzy główne przyczyny poprawy procentowego projektów ukończonych sukcesem i do nich należą:

1. Obserwowany trend dekomponowania projektów na mniejsze aplikacje.
2. Ogólny wzrost umiejętności i kompetencji kierowników projektów oraz postęp nauki w dziedzinie zarządzania projektami.
3. Upowszechnianie standardów i narzędzi wspomagających zarządzania projektami.

W kolejnym rozdziale rozwinięto niektóre aspekty dotyczący wielkości projektu i wpływu tego parametru na sukces projektu.

## 7.2. Wpływ wielkości projektu na jego sukces

W jakim stopniu planowanie projektu wpływa na sukces projektu niewątpliwie zależy od wielkości projektu. Wiadomo, że jeżeli mamy do czynienia z dużymi projek-

tami, dokładne, skrupulatne, szczegółowe planowanie decyduje o sukcesie projektu. Nie jest możliwe przy takich rozmiarach projektu zaniedbać proces planowania. Przy małych projektach natomiast jest większa szansa na powodzenie, nawet przy niezbyt dokładnym i poprawnym planowaniu. Nasuwa się pytanie jak rozróżnić małe i duże projekty. Czy jest jakieś kryterium oceny wielkości projektu?

W 1979 roku na zlecenie IBM Allan Albrecht zaprezentował metodę punktów funkcyjnych. Punkty funkcyjne są miarą wielkości aplikacji komputerowych i projektu, który je stworzył. Wielkość ta jest mierzona ze względu na funkcjonalność z punktu widzenia użytkownika.

### Według standardu ISO/IEC/JTC1/SC7 14143

Rozmiar funkcjonalny to rozmiar oprogramowania otrzymany przez obliczenie funkcjonalnych wymagań użytkownika. Z pojęciem punktu funkcyjnego związany jest czynnik modyfikujący wartość punktów funkcyjnych VAF (ang. *Value Adjustment Factor*). Uzyskuje się go przez odpowiedzenie na 14 pytań związanych z projektem. Odpowiedzią jest ustalenie ważności podanego czynnika dla naszego systemu (skala od 0 do 5). Końcową wartość oblicza się ze wzoru:

$$VAF = 0,65 + [(\sum Ci)/100], \quad \text{gdzie } 0 < i \leq 14$$

Według IFPUG średniej wielkości projekt ma 899 punktów funkcyjnych.

ISBSG (ang. *International Software Benchmarking Standards Group*) w dokumencie *Release 6* w kwietniu 2000 roku podaje przykładowe obliczanie kosztu średniego projektu (889 punktów funkcyjnych) pisanego w języku C++. Średnia cena za punkt funkcyjny wynosi \$1,234. Praca przypadająca na jeden punkt funkcyjny trwa 14 godzin. Liczenie kosztu od strony klienta: \$90 za godzinę pracy [50, 55].

Ponieważ zauważono, że projekty mniejsze częściej kończą się sukcesem, więc zachodzi pytanie, co to jest mały projekt. Wielkość projektu jest sprawą względną, związaną z wielkością firmy i jej poziomem kompetencji, stosowanymi standardami oraz doświadczeniem. Te zagadnienia i ich ocena jest procesem dynamicznym, ponieważ wynika z tego, że krótsze ramy czasowe wytwarzania komponentów zwiększają współczynnik sukcesu, dzięki iteracyjnemu procesowi projektowania, prototypowania, testowania i przekazywania małych elementów. Większe systemy powstają z małych komponentów i każdy z nich ma posiada precyzyjnie określony zestaw celów i realistycznych oczekiwań klienta.

## 8. Rola i zadania kierownika projektu

Jak już zaznaczono we wprowadzeniu tej książki wiedza na temat zarządzania projektami nie powinna być zarezerwowana jedynie dla najwyższego kierownictwa, odpowiedzialnego za całokształt organizacji oraz realizację projektów. To samo dotyczy PM, jest ona bezsprzecznie potrzebna do realizacji celów projektu, podejmowania trudnych decyzji i alokowania zasobów w sposób sprzyjający realizacji projektów. Jednak bez odpowiedniego przygotowania zespołów projektowych nie ma gwarancji, że zadania przydzielane w poszczególnych etapach projektów zostaną prawidłowo zrealizowane. Wszyscy członkowie organizacji potrzebują wiedzy i umiejętności w zakresie zarządzania projektami – jedni bardziej pogłębionej i rozbudowanej, inni mniej szczegółowej. Mimo że coraz częściej możemy spotkać sytuacje, w której ktoś lub kogoś nazywamy *Project Managerami*, coraz częściej liczą się formalne, poświadczane certyfikatem uprawnienia do tego tytułu. Dyplomy wydawane przez MT&DC przy współpracy z *Educational Services Institute International (ESI®)* potwierdzają uzyskanie przez uczestnika kursu, gruntownego wykształcenia i nabycia kompetencji w tym zakresie. Otrzymanie certyfikatu ESI® łączy się z fundamentalnym poznanie niezbędnej tematyki związanej z zarządzaniem projektem i otwiera drogę do uzyskania dyplomu *Project Management Professional (PMP®)*.

Proces realizacji projektu jest niejednokrotnie długotrwały i skomplikowany, podczas realizacji projektu PM doświadcza zagadnień nie tylko dotyczących projektu, ale również zmian organizacji firmy, która realizuje projekt (zmiany zarządu, reorganizacja, zwolnienia itd.) W projektach informatycznych, szczególnie dużych, kiedy mamy do czynienia z wieloma zespołami zadaniowymi (np. zespół analityków, zespół wytwarzania aplikacji – produkcyjny, zespół komunikacji i przekazu elektronicznego dokumentów, baz danych, integracji, bezpieczeństwa itp.), mamy do czynienia z nowymi technologiami, nowatorskimi rozwiązaniami oraz ze specjalizacją prac w zespołach. Dynamika zmian w zakresie rozwoju narzędzi wspomagających prace projektowe i produkcyjne oprogramowania stworzyła sytuację, że już od kilku lat jeden człowiek nie jest w stanie być specjalistą ze wszystkich działach informatyki, zatem współczesny PM to przede wszystkim:

**Biznesmen** – do jego zadań należy:

- współpraca z klientami,
- adaptowanie zmian, wymagań i relacji wewnątrz firmy,

- szacowanie kosztów związanych z alternatywnymi wyborami.

**Technolog** odpowiedzialny za trafny wybór:

- zasobów,
- produktywność i efektywność działań ,
- innowacje techniczne,
- adaptowanie metod działania.

**Zarządzający**

- ludźmi,
- zasobami,
- zmianami,
- komunikacją,
- powiązaniem organizacyjnym z kooperantami,
- pracą grupową,
- rozwiązujący konflikty,
- twórca i lider zespołu.

PM to architekt projektu – integrujący zagadnienia ekonomiczne i techniczne, niekoniecznie „superspecjalista” w dziedzinie informatyki.

## 8.1. Usytuowanie kierownika projektu a jego skuteczność

Miejsce i rola kierownika projektu zależy między innymi od wielkości firmy i czym dla niej jest projekt. Może to być jedyny projekt dla firmy, cała firma pracuje na jego rzecz, a szef firmy kieruje jednocześnie projektem. Jednak istnieje inna możliwość: wiele projektów realizowanych jest równocześnie w firmie, a szef firmy jest informowany lub interesuje się tylko wybranymi projektami. W poszczególnych branżach czy zastosowaniach bezpośredni nadzór sprawują członkowie zarządu firmy. To czy firma realizująca projekt liczy 5–10 osób czy 2000 zasadniczo wpływa na usytuowanie PM. Typowe grupy i osoby, z jakimi ma do czynienia kierownik projektu:

- kierownicy innych zających się projektów,
- zarząd firmy i jego przedstawiciele,
- szefowie działów firmy, w której pracuje,
- członkowie zespołu projektowego,
- inni pracownicy firmy, którzy nie biorą udział w pracach nad projektem,
- zleceniodawcy zewnętrzni,
- dostawcy sprzętu i oprogramowania – podwykonawcy,
- kontrolerzy,
- radcy prawni,

- personel techniczny zapewniający pracę środowiska systemu oraz serwis,
- administracja firmy (dział finansowy, marketingu).

## 8.2. Dobór członków zespołu

Największe szanse na pomyślne zrealizowanie projektu (zadania) mają zespoły, które mają w swym składzie pracowników, którym można przypisać 8 poniższych ról (wg Belblina – na podstawie doświadczeń):

- *chairman*, koordynator,
- nadający kształt aplikacji,
- ogrodnik – zaszczipiający idee,
- kontroler – monitorujący prace i nadzorujący czy nie ma odstępstwa od założonego kierunku planu prac,
- brygadzysta, rozkładający pracę,
- osoba nastawiona na analizę zewnętrzną zasobów, zmian w środowisku, nowe rozwiązania techniczne, programowe itp.,
- wykonawca, członek zespołu,
- czuwający nad terminowym i właściwym zakończeniem.

Dobranie pracowników, o powyższych naturalnych predyspozycjach, sprzyja atmosferze pracy, a każdy pracownik wykonuje taką pracę, z której jest zadowolony i najefektywniejszy. Zadaniem kierownika projektu jest nie tylko uświadomienie sobie kogo potrzebuję, zdefiniowanie zadań, zakresu prac, ale również właściwe poznanie predyspozycji członków zespołu i podział zadań.

Przy specyfikacji zadań i ich rozdziale należy zwrócić szczególną uwagę na:

- jakie są podstawowe obowiązki, czy istnieje specjalna odpowiedzialność za sprzęt, bezpieczeństwo innych,
- główne trudności i uciążliwości,
- z jakimi osobami będzie mieć kontakt osoba wykonująca zadanie,
- zakładany stopień nadzoru, swobody w podejmowaniu decyzji,
- czy istnieją inne niedogodności tej pracy.

W specyfikacji zadania mogą pomóc:

- doświadczenia pracy nad analogicznymi projektami,
- fachowcy, zatrudnieni w firmie.

## 8.3. Rekrutacja członków zespołu

Coraz częściej rekrutacja do danej firmy – projektu odbywa za pośrednictwem wyspecjalizowanych firm lub komórki organizacyjnej, które zajmują się również zarzą-

dzaniem wiedzą i modelowaniem ścieżek rozwoju pracowników. Na ogół głównym czynnikiem, który wpływa na sposób i metodę rekrutacji:

- specyfikacja zadania jest podstawą dla charakterystyki poszukiwanego pracownika,
- pod uwagę należy brać budowanie zespołu i wpływ jednostki na zespół, na wspólną pracę,
- podstawowe elementy charakterystyczne poszukiwanego pracownika:
  - oczekiwane zdolności i możliwości kandydata,
  - cechy jako współpracownika,
  - osobowość.

Metody rekrutacji z firmy:

- spośród kolegów,
- od zleceniodawcy,
- dawni pracownicy,
- osoby, które wcześniej ubiegały się o pracę,
- inni, którzy przyprowadzą kolejne osoby,
- ogłoszenia (koszt),
- agencje pośrednictwa pracy,
- czasem: rządowa rekrutacja, przez centra zawodowe lub uczelnie czy kursy,
- Internet (formularze elektroniczne zgłoszenia zarówno wolnego miejsca pracy- poszukiwanie pracownika, jak również zgłoszenie swojego chęci zatrudnienia).

Metody wyboru:

- tworzenie krótkiej listy,
- na podstawie *interview*,
- selekcja zespołowa,
- test wyboru.

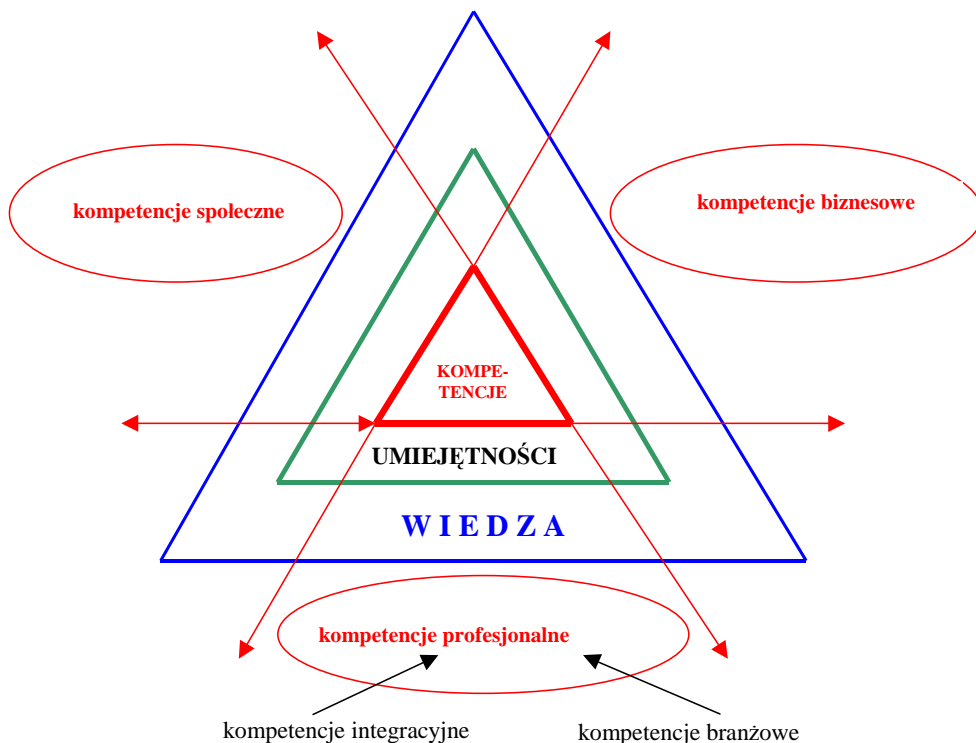
## 8.4. Budowa kompetencji w projekcie

Prowadzenie projektów informatycznych w nowoczesnych firmach związanych z rynkiem IT opiera się na zespołach ludzi o zróżnicowanych kwalifikacjach. Spektrum problemów, z jakimi zespół często musi się zmierzyć, wymaga, by kompetencje pracowników nawzajem się uzupełniały. Powodzenie realizacji projektu, między innymi, wynika z poziomu kompetencji całego zespołu. Struktura przydziału odpowiedzialności i zakresu obowiązków poszczególnych pracowników powinna być możliwie optymalna. Jasna ścieżka rozwoju technologicznego bądź aplikacyjno-wdrożeniowa ma udział w motywowaniu członków zespołu do systematycznego podnoszenia własnych kwalifikacji, co przekłada się na budowę kompetencji całej firmy.

**Wiedza (praktyczna i teoretyczna)** – wiedza mówi o tym, że dana osoba miała już do czynienia z wybranym zagadnieniem, mogła o tym zagadnieniu czytać lub ob-

serwować tok realizacji zadań z nim związanych, jednak nie miała okazji zdobyć praktyki w jego realizacji. Jeśli osoba posiada tylko wiedzę z danego zagadnienia, kierownik nie decyduje się raczej przydzielać pełnego zadania do wykonania przez tę osobę, ponieważ z wiedzy zgodnie z niniejszą definicją nie wynika jeszcze zdolność realizacji. W niektórych przypadkach znajomość dotyczy zagadnień, które są opisywane w literaturze lub wykładane na uczelniach, lecz nie ma zbyt silnego uzasadnienia, aby takie działania powtarzać, ponieważ są one już gdzie indziej praktykowane i uznane jako wzorcowe. Przykładem może być budowa kompilatorów, w czym specjalizują się duże koncerny, a zatem rzadko powtarzane są takie prace, więc również nie istnieje zbyt wiele sposobności w realizacji zadań z tym związanych.

**Zdolności** – to cechy osoby, która jest w stanie wykonać zadanie, którego rezultatem jest określona kategoria produktu (np.: raport z zakończonej sukcesem instalacji oprogramowania) bądź (poprawny) stan konfiguracji systemu (np.: zainstalowany system klasy...). Zdolność do wykonania określonej kategorii zadania oznacza, że dana osoba może zostać przydzielona do wykonania tego zadania, a jej praca nad zadaniem zakończy się sukcesem (należy zatem ze **zdolnością** szczególnie silnie kojarzyć pojęcie **samodzielności** w działaniu).



Rys. 8.1. Schemat ogólny klasyfikacji kompetencji



**Umiejętność** – mówi o tym, że dana osoba jest w stanie wykonać czynność praktyczną w sposób automatyczny (bez szczególnych przemyśleń, np.: może opierać się na naśladownictwie lub cechach wrodzonych) lub ze wsparciem. Umiejętność w szczególności dotyczy niezbyt złożonych podzadań cząstkowych, których wykonanie nie wymaga silnego zaangażowania w koordynację zasobów i zarządzanie. Umiejętność przejawia się także możliwością wykonania działań pod kierunkiem osoby bardziej doświadczonej, gdzie nie istnieje dla wykonującego konieczność szczegółowego planowania (nie jest wymagany duży stopień samodzielności).

**Kompetencje** – zdolności praktycznego wykorzystania umiejętności i wiedzy w pełni wystarczające do samodzielnego wykonywania określonego zadania w projekcie.

Przedstawiony na rysunku 8.1 graficznie podział i przenikanie się granic między wiedzą, umiejętnościami a kompetencjami, szczególnie istotne dla PM stają się kompetencje profesjonalne w obszarze produktów branżowych oraz integracyjnych. W przedsięwzięciach integracyjnych nie bez znaczenia są postawy prezentowane partnerów, osobowość PM oraz szersza wiedza wykraczająca poza wiedzę dotyczącą produktu z uwagi na konieczność prowadzenia wielu rozmów i uzgodnień nie tylko biznesowych, ale w kontekście otoczenia projektu. Uwarunkowania oraz umiejętności społecznej komunikacji i współdzielenia zainteresowań partnerów przedsięwzięcia jest kluczem zawiązania relacji, zaufania jak również współdziałania.

Kompetencje w projektach mogą być dzielone na 3 klasy:

- Profesjonalne.
- Biznesowe.
- Społeczne.

1. Wykaz kompetencji pożądaných dla realizacji danego projektu powinien znajdować się w opisie zasobów projektu.

2. Kompetencje podlegają ciągłej ocenie oraz są wyjściem do podejmowanych przez pracownika i w stosunku do pracownika działań i decyzji.

3. Działania i decyzje mogą być powiązane z pracami w projektach u klienta lub pracami wewnętrznymi (w szczególności projektami wewnętrznymi).

4. Wykorzystanie kompetencji w powyższych działaniach przekłada się na stopień realizacji celów przydzielonego stanowiska w projekcie.

5. Stopień realizacji celów stanowiska przekłada się na stopień realizacji celów zespołu.

6. Stopień realizacji celów zespołu przekłada się na stopień realizacji celów projektu.

W planowaniu należy uwzględnić każdy z powyższych zasobów i zdefiniować ograniczenia, jakie wnoszą do projektu. Niektóre zasoby są czasami rozmyte lub trudno definiowalne (biznesowe, społeczne, techniczne, środowiskowe, etyczne, polityczne), ale w niektórych projektach mogą mieć podstawowe znaczenie w doprowadzeniu projektu do sukcesu [5, 13, 30, 42, 59].

## 8.5. Konflikt

W ostatnich kilkunastu latach uległy zmianie zapatrywania na konflikt, czyli problem nieporozumień oraz brak zgodności co do sposobu podejścia do realizacji projektu czy metody realizacji. *Konflikt to sztuka przekonywania do swoich racji na tle zaburzeń komunikacji w zespole*. Obecnie uważa się, że konflikty, które powstają w zespołach pracujących nad projektami są naturalne i należy je wykorzystywać w celu osiągnięcia lepszych efektów pracy.

Tabela 8.1

Tradycyjnie	Współcześnie
Zbędny i szkodliwy	Nieunikniony
Można uniknąć	Należy nim kierować
Powodem jest błąd kierownictwa lub osoby konfliktowej	Zła struktura zarządzania, sposób postrzegania członków przez kierownictwo

### Przyczyny konfliktów:

- konieczność dokonania wyboru,
- zaspokojenie postulatów – oczekiwań kosztem innych,
- pełnienie funkcji w zespole i na zewnątrz.

Kompromis jest nietrwały i nie rozwiązuje problemu (*zgniły kompromis*), najczęściej łagodzi sytuacje konfliktowa na okres przejściowy, strony konfliktu czują się niezadowolone i oczekują na następną sytuację, kiedy i ich racje zwyciężą. Każdy element zbiorowości ludzkiej dąży do zaspokojenia swoich celów, co jest przyczyną konfliktów, ale prowadzi do zmian w tych zbiorowościach [11, 22, 56].

### Typy konfliktów:

- wewnątrzno-osobnicze,
- międzyludzkie,
- wewnątrz grupowe,
- międzygrupowe.

Zmiany są główną przyczyną powstawania konfliktów, ponieważ obejmują zagadnienia, które dotyczą ambicji oraz emocjonalnego zaangażowania najbardziej kreatywnych i o wyraźnej osobowości członków zespołu.

## 8.6. Motywowanie

Według Griffina [22] *Motywowanie, to zestaw sił, które powodują, że ludzie zachowują się w określony sposób*. Zasadniczym celem stosowania technik motywacyjnych

nych jest zwiększenie efektywności i wydajności pracy – jako że pracownik, który posiada odpowiednią motywację do wykonywania swojej pracy – wykonuje ją lepiej i szybciej.

Jak wykazały badania, najskuteczniejszym czynnikiem motywacyjnym nie jest wcale wysokie uposażenia pracowników, jak tradycyjnie przyjmowano za pracą Fredericka W. Taylora. Wprawdzie motywują one do zwiększania swojej wydajności i podwyższania kompetencji, ale nie stanowią czynnika budującego lojalność wobec firmy – pracownik motywowany głównie poprzez płacę nie ma oporów przed odejściem do konkurencyjnej firmy, jeżeli ta zaproponuje mu odpowiednio wysoką pensję. W ten sposób, przez odływ kapitału intelektualnego, można utracić znaczącą część potencjału firmy, co wywrze negatywny wpływ na jej wartość rynkową. Na przestrzeni ostatnich lat powstało wiele podejść i teorii motywowania, jak: motywowanie od strony treści oraz hierarchii potrzeb, np. przynależności, szacunku, teoria ERG. Hierarchia potrzeb według Masłowa sugeruje, że ludzie muszą zaspokoić pięć kolejnych potrzeb – fizjologicznych, bezpieczeństwa, przynależności, szacunku i samorealizacji.

Najskuteczniejszym motywatorem jest, jak wykazały badania, sama praca, o ile jest ona zgodna z kompetencjami i zainteresowaniami pracownika, pozwala mu ona na samodoskonalenie się i rozwój zawodowy i intelektualny. Dodatkowo, jest to silny element budujący lojalność pracownika wobec firmy i dodatkowy argument przeciwko odejściu z niej w razie próby przejęcia pracownika przez konkurencyjną firmę.

Wybrane czynniki motywujące, co do których istnieje konsensus to:

- rozdział pracy (interesująca, odpowiedzialna) – wspomniana wyżej praca zgodna z kompetencjami,
- łączenie celów indywidualnych – integracja celów indywidualnych z celami projektu (w ramach możliwości),
- perspektywiczność i rozwojowość.

Wybranymi czynnikami demotywującymi są:

- niski poziom płac,
- środowisko pracy,
- styl pracy zespołu, klimat (lider),
- źle dobrana praca,
- niewłaściwy poziom nadzoru lub jego brak,
- źle ustawione granice odpowiedzialności.

Jako potwierdzenie powyższych stwierdzeń może posłużyć cytata z artykułu Joanny Chylewskiej [13]:

*Najbardziej efektywnymi sposobami motywowania pracowników jest zapewnienie im możliwości zdobywania osiągnięć w zakresach, które są spójne z ich najgłębiej zakorzenionymi zainteresowaniami i pasjami*

Mechanizmami łączącymi motywowanie z zarządzaniem kompetencjami może być na przykład kompetencyjny system wynagrodzeń, uzależniający wysokość uposażenia pracownika, od zakresu jego kompetencji. Należy jednakże zaznaczyć, że rzadko można

spotkać rozwiązania, w których kompetencje stanowią jedyne kryterium wyznaczania wysokości pensji. Trudności z oceną kompetencji i wyceną ich wartości, a także bariery psychologiczne, powodują, że stosowane są systemy hybrydowe, w których ocena kompetencji stanowi tylko jeden z czynników wpływający na wysokość zarobków pracownika, będąc po części równoważone przez wycenę wartości wytwarzanych przez pracownika

(tj. przez ocenę wartości efektów jego pracy). Niemniej jednak system kompetencyjny potrafi skutecznie zachęcić pracownika do samodoskonalenia się i rozwoju.

Innym mechanizmem motywującym, dbanie o własne kompetencje jest system ocen okresowych – w którym to systemie co pewien czas urządza się badanie kompetencji pracowników, co stanowi później podstawę do oceny efektywności i jakości pracy danej osoby. Świadomość silnej konkurencji na rynku pracy i istotności wyników badań okresowych stanowi również bardzo silny czynnik motywujący pracowników do rozwoju.

Warto wreszcie przytoczyć bardzo celne stwierdzenie Ulricha:

*Kapitał intelektualny = Kompetencje · Motywacje, D.Ulrich, 1998*

Stwierdzenie to jest wyrażeniem bardzo prostej prawdy: na wartość pracownika jako kapitału intelektualnego składają się w równym stopniu jego kompetencje, jak i motywacja. Pracownik kompetentny, ale o słabej motywacji nie jest w stanie osiągnąć maksimum swojej potencjalnej wydajności i jakości pracy, podobnie w przypadku pracownika dobrze zmotywowanego, ale nie posiadającego odpowiedniego zakresu kompetencji. Dopiero pracownik kompetentny i dobrze zmotywowany stanowi naprawdę wartościowy nabytek dla firmy i jest zdolny do osiągania znakomitych wyników zarówno pod względem wydajnościowym, jak i jakościowym.

Ludzie dążą do spełnienia swoich celów. W innym przypadku pojawiają się „złe” odczucia, mające swoje konsekwencje w zachowaniu.

#### **Typowe cele pracy, które stymulują rozwój kompetencji:**

- wspólne cele zawodowe zmierzające do:
  - sukcesu,
  - finansowej satysfakcji,
  - rozwoju, nauki;
- indywidualne:
  - towarzystwo, atmosfera w pracy,
  - bezpieczeństwo,
  - odskocznia, zainteresowania.

Zadaniem prowadzącego projekt jest odpowiednie motywowanie i integracja celów zawodowych z celami indywidualnymi członków zespołu.

#### **Czynniki motywujące**

Coraz częściej, a wynika to z licznych ankiet, jak również preferowanych przez pracowników ścieżek rozwoju, że głównymi czynnikami motywującymi do pracy są:

- właściwy rozdział pracy dla każdego – interesującej i odpowiedzialnej,

- łączenie celów zespołowych z indywidualnymi,
- perspektywiczność i rozwojowość wykonywanej pracy,
- możliwość awansu.

#### **Czynniki demotywuujące**

- niski poziom płac,
- środowisko pracy,
- styl pracy zespołu, klimat (lider),
- źle dobrana praca: za łatwa, za trudna, nużąca, nie odpowiadająca zainteresowaniom, nie satysfakcjonująca,
- niewłaściwy poziom nadzoru lub pomocy, brak zainteresowania przełożonych,
- źle ustawione granice odpowiedzialności.

## **8.7. Delegowanie uprawnień**

Bardzo istotnym czynnikiem motywowania pracownika jest umiejętne przekazanie i przejęcie przez członków zespołu niektórych kompetencji (między innymi dotyczących planowania, kontroli) od kierownika projektu. Otrzymane uprawnienia powinny być powiązane nie tylko z możliwością podejmowania decyzji, ale też z ponoszeniem konsekwencji. Jest to forma zwiększenia efektywności i zaangażowania pracy nad projektem członków zespołu.

#### **Czynniki przeciwne delegowaniu uprawnień**

Delegowanie uprawnień nie jest powszechnie akceptowanym działaniem i możliwości jego zastosowania może ograniczać:

- niechęć kierownictwa do pozbywania się części uprawnień,
- niechęć personelu do podejmowania odpowiedzialności.

Szczególnie w organizacjach, gdzie autorytet czy pozycja zależna jest od tzw. funkcji (kierownika, dyrektora itd.) zachodzi obawa, że przekazanie części swoich uprawnień może stworzyć konkurencję lub osłabić możliwości oddziaływania na podwładnych. Personel przyjmujący uprawnienia musi mieć gwarancję, że podejmując się wykonywania niektórych zadań, które przynależne są wyższemu personelowi, działając w sytuacji braku doświadczenia, może popełnić błędy. I jeśli te błędy nie będą wynikały z zaniechania lub niedbalstwa, to przyjmujący uprawnienia nie poniesie odpowiedzialności.

Pracownicy – ich pożądane postawy:

- nie powinni mówić, że nie potrafią się czegoś zrobić,
- nie powinni bagatelizować trudności zadania,
- domagać się prawa do popełniania błędów.

#### **Skuteczność zespołu według Mc Gregora**

Mc Gregor sformułował podstawowe czynniki, które powinny występować w zespo-

le, aby działał skutecznie, są nimi:

- pozbawiona oficjalności, rozluźniona atmosfera,
- dużo i wszyscy dyskutują o przedmiocie projekt,
- cele i zadania są dobrze rozumiane i akceptowane,
- członkowie zespołu uważnie słuchają opisu kolegów,
- dopuszcza się niejednomysłność, nieporozumień nie dusi się w zarodku, przyczyny są głęboko rozważane,
- większość decyzji na zasadach konsensusu, rzadko oficjalne głosowania, większość głosów nie jest wystarczającą przesłanką do podjęcia określonych kroków,
- opinie krytyczne pojawiają się często, ale nie są personalizowane, odnoszą się do zadań oraz działań zespołu,
- gdy podejmuje się czynności, funkcje są jasno określone i akceptowane,
- przywódca grupy nie dominuje nad szeregowymi pracownikami.

Aby zespół natomiast opanował wymienioną metodę, skład członków tego zespołu powinien posiadać określone predyspozycje i cechy osobowe, które zostały sformułowane przez Belblina (patrz rozdz. 8.2. Dobór członków zespołu).

## 9. Dodatek

Ten rozdział ma wskazać na przykładowe fakty w postaci wypracowanych i przyjętych przez niektóre organizacje metod zarządzania projektami oraz narzędzia, które powstały, aby je wspomagać.

### 9.1. Metoda zarządzania projektami PRINCE-2

Metoda PRINCE 2 powstała w 1989/96 dzięki CCTA (*Central Computer and Telecommunications Agency*)/SPOCE. Została ona oparta na wcześniejszej metodzie, znanej pod nazwą PROMPT. W 1999 firma CRM S.A. adaptuje metodę PRINCE 2-SPOCE-CRM do warunków polskich [6]. PRINCE 2 jest standardem brytyjskiej administracji publicznej i biznesu do zarządzania projektami, jest też powszechnie przyjęta jako standard zarządzania projektami wszystkich rodzajów w sektorze publicznym i prywatnym.

1. Projekt jest skończonym zbiorem aktywności, mającym swój początek i koniec.
2. Projekt musi być zarządzany, aby skończył się sukcesem.
3. Aby uzyskać właściwe zaangażowanie stron, wszyscy muszą mieć pełną jasność co do celu, sposobów realizacji, odpowiedzialności stron;

Korzyści z zastosowania pakietu PRINCE 2-SPOCE-CRM.

Dzięki elastyczności tej metody jest stosowana zarówno do wielkich, wysokobudżetowych projektów, jak i do małych przedsięwzięć wewnątrz organizacji. Metoda ta może być z powodzeniem użyta do zarządzania wielkimi projektami inicjowanymi przez duże organizacje i agencje rządowe (np. PRINCE 2 jest standardowo używana przez brytyjskie instytucje rządowe), do zarządzania różnej skali projektami używane przez brytyjskie agencje rządowe).

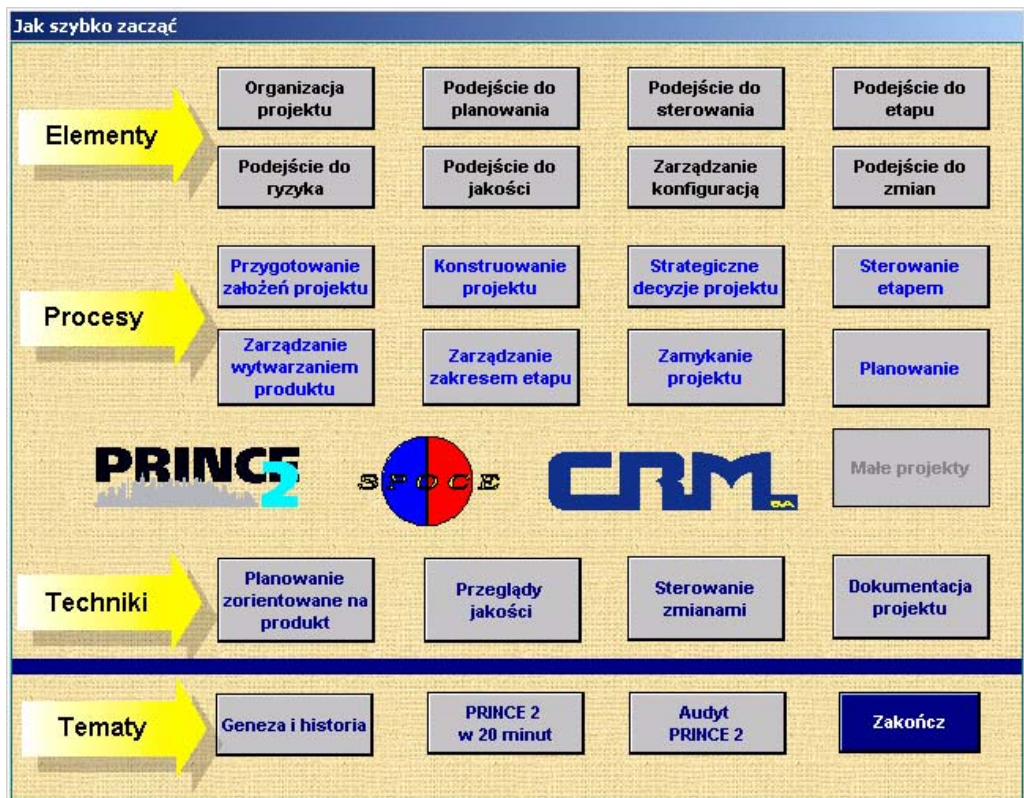
Podstawowe właściwości tej metody:

- skupienie na ocenach według kryteriów biznesowych,
- procesowe podejście do sterowania zarządzaniem, zespołem i jakością,
- zdefiniowana struktura organizacyjna zespołu zarządzającego projektem,

- planowanie działań zorientowane na produkty,
- dekomponowanie projektu na łatwo zarządzane etapy,
- zarządzanie ryzykiem podczas całego cyklu realizacji projektu,
- ustalone procedury postępowania,
- dokładny i efektywny system dokumentowania,
- kontrolowanie i zorganizowanie rozpoczęcia, rozwinięcia i zakończenia projektu,
- śledzenie postępów w stosunku do planów i założeń,
- automatyczna kontrola wszelkich odchyłeń od planu oraz elastyczność punktów decyzyjnych,
- zaangażowanie zarządu i akcjonariuszy we właściwym czasie i stopniu w projekt.

Dla kierownik projektu – możliwości pakietu

- opracowanie wstępnych założeń przed rozpoczęciem projektu oraz delegowanie uprawnień, dzielenie projektu, raportowanie.



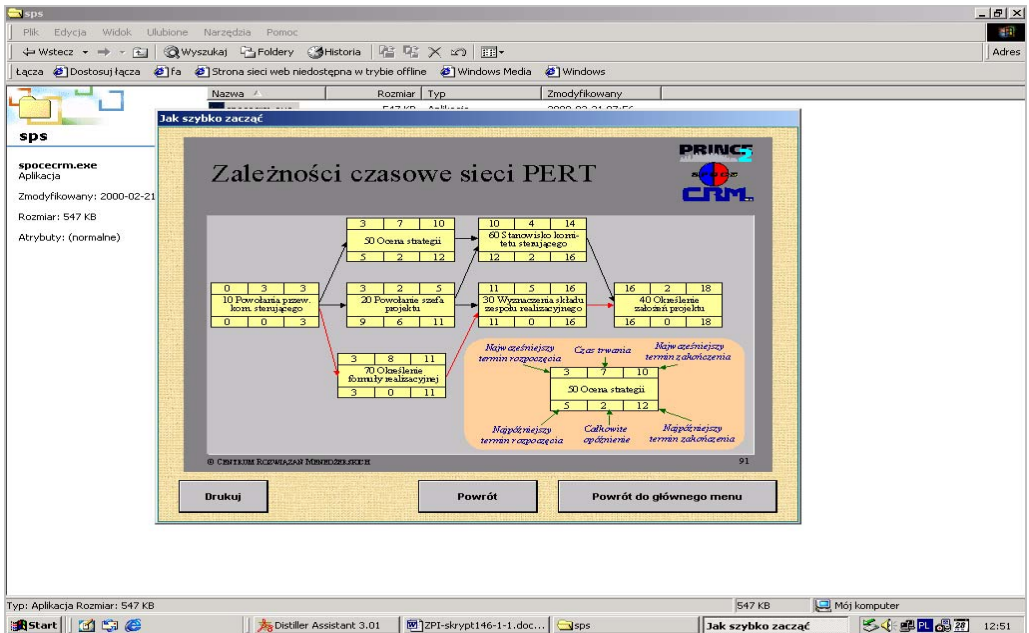
Rys. 9.1. Platforma startowa pakietu PRINCE 2-SPOCE – CRM



Z rysunku 9.1 widać, że metoda PRINCE 2 porządkuje czynności ich kolejność oraz produkty, które mają powstać w trakcie prowadzenia projektu. W metodzie tej wyróżnia się PROCESY, do których należy:

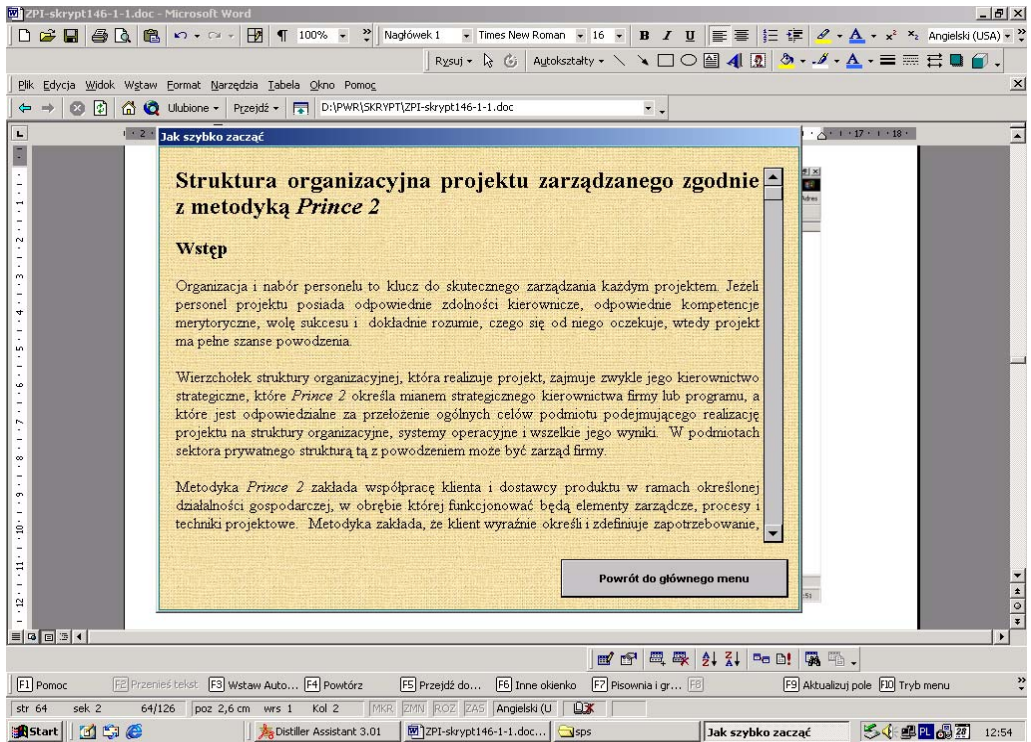
1. Przygotowanie założeń projektu.
2. Konstruowanie projektu.
3. Strategiczne decyzje projektu.
4. Inne procesy.

Procesy te są wspomagane TECHNIKAMI, które udostępniają narzędzia programowe lub przygotowane formularze, których wypełnianie jest elementem metody prowadzenia projektu. Załączone ekrany (rys. 9.2) wskazują na wykorzystywanie PERT w analizie czasowej realizacji projektu, w której szacuje się najwcześniejszy czas rozpoczęcia zadania, najpóźniejszy czas rozpoczęcia zadania, czas jego trwania, opóźnienie itd.



Rys. 9.2. Wyznaczanie ścieżki krytycznej i szacowanie zadań w projekcie za pomocą PERT w metodzie PRINCE 2

Przestrzeganie wszystkich zaleceń, które są zawarte w tzw. ELEMENTY, to między innymi struktura organizacyjna projektu zarządzana zgodnie z metodą PRINCE 2 (rys. 9.3).



Rys. 9.3. Opis struktury organizacyjnej projektu w metodzie PRINCE 2

## 9.2. Oprogramowanie wspomagające zarządzanie zmianami

Jedną z decyzji, którą należy podjąć we wczesnej fazie projektu, jest określenie sposobu śledzenia zachodzących w nim zmian. Powodzenie najmniejszych nawet przedsięwzięć informatycznych w istotny sposób zależy od tego, jakimi i czy odpowiednimi narzędziami będą dysponowali jego uczestnicy. Oprogramowanie przeznaczone do kontroli wersji powinno zapewniać analizę historii zmian i możliwość uzyskania kopii dowolnej wersji archiwizowanych artefaktów, umożliwiać wprowadzanie do projektu modyfikacji widocznych także dla innych jego uczestników i zapobiegać, lub chociaż informować, o powstałych w wyniku tych operacji konfliktach.

W najprostszym przypadku możliwe jest oczywiście wykorzystanie ręcznie tworzonych i odpowiednio nazywanych kopii projektu. Podejście to jest jednak bardzo nieefektywne, podatne na błędy, utrudnia przeglądanie zmian i jest praktycznie niemożliwe do zastosowania w zespołach większych niż jednoosobowe. Na szczęście

istnieje cała gama dedykowanego do tego celu oprogramowania, począwszy od prostych narzędzi, przeznaczonych dla pojedynczych deweloperów, na złożonych systemach skierowanych do zespołów liczących setki uczestników. Przegląd ten ma na celu przedstawienie głównych cech oprogramowania oraz wskazanie najbardziej znaczących narzędzi wspomagających zarządzanie zmianami.

Najczęściej jest to oprogramowanie darmowe, lecz niekoniecznie z pełnym wsparciem ze strony producenta. Jest dużo narzędzi dostępnych bez opłat – niektóre z nich są dostarczane z większością systemów *Unix*, niektórych trzeba poszukać w Internecie. W niektórych przypadkach wymagają samodzielnej kompilacji. Do większości darmowych narzędzi dostarczana jest wystarczająca dokumentacja. Ponieważ wiele z tych narzędzi jest dostarczana właściwie bez żadnego wsparcia, nie zaleca się używania ich w pewnych projektach. Dla kompletności zostały one ujęte tutaj pomimo potencjalnych wad.

### 9.3. Najpopularniejsze narzędzia *Open Source*

#### **CSSC – Compatibly Stupid Source Control (<http://cssc.sourceforge.net/>)**

Dokonana przez *Free Software Foundation* reimplementacja najstarszego (i przed pojawieniem się RCS jedyne) unixowego narzędzia kontroli wersji – SCCS (*Source Code Control System*). Jego zaletą jest pełna standaryzacja określona normą X/Open, powodująca, że dostępne jest we wszystkich markowych wersjach systemów z rodziny Unix. Oferuje funkcjonalność praktycznie identyczną z dostępną za pomocą RCS. Obecnie nie jest już praktycznie używane.

#### **RCS – Revision Control System (<http://www.gnu.org/software/rcs/rcs.html>)**

Jest to jedno z najstarszych (powstało w latach 80.) narzędzi. Program kontroluje modyfikacje pliku źródłowego i utrzymuje plik z listą zmian, zawierającą informacje potrzebne, by odtworzyć dowolną poprzednią jego wersję. Pozwala w łatwy sposób zapisywać, odzyskiwać i odczytywać dane oraz łączyć wersje. Jest obsługiwany przez wszystkie popularne w środowisku Unix narzędzia programistyczne (w tym *make*). Pracuje na poziomie pojedynczych plików i nie oferuje mechanizmów ułatwiających pracę grupową.

#### **CVS – Concurrent Versions System (<http://www.cvshome.org/>)**

Jest to obecnie najpopularniejsze narzędzie z rodziny *Open Source*. Powstało w 1986 r., by usprawnić i rozszerzyć możliwości RCS, na którym się opiera. Jest syste-

mem w pełni sieciowym, bazującym na centralnym repozytorium. Umożliwia jednoczesną pracę wielu programistów i oferuje mechanizmy automatycznego uzgadniania wprowadzonych przez nich zmian. Znacznie lepiej niż RCS zarządza zbiorami danych. Podstawową jednostką informacji jest w nim pojedynczy plik, lecz przez użycie mechanizmu znaczników pozwala również odnosić się do całości projektu. Umożliwia tworzenie odgałęzień. Zawiera mechanizm automatycznie rozwijanych słów kluczowych (np. \$Author\$ jest rozwijany do systemowej nazwy autora pliku). Oferuje oczywiście także dostęp do pełnej historii wydań, informacji o autorach wprowadzonych zmian, nadanych im komentarzach itp. Jest dostępny we wszystkich znaczących systemach operacyjnych i obsługiwany przez wszystkie popularne środowiska programistyczne.

### **Subversion (<http://subversion.tigris.org/>)**

Program pomyślany jako następca CVS. Wychodzi z podobnych założeń i oferuje podobne funkcje, lecz stara się unikać głównych wad CVS. Operacje przydziału znaczników i tworzenia nowych odgałęzień są w nim znacznie szybsze. Dysponuje lepszym wsparciem dla plików binarnych. Oferuje wersjonowanie katalogów i metadanych repozytorium. Znacznie lepiej wspiera zmianę nazw plików i zapewnia atomizację zmian repozytorium. Dane nie są przechowywane w formie RCS, lecz umieszczone w specjalnej bazie danych (aktualnie Berkeley DB). Mechanizmy sieciowe oferuje, wykorzystując serwer Apache. Jest dostępny dla wszystkich ważnych systemów operacyjnych. Projekt nie doczekał się jeszcze bazy użytkowników, choćby porównywalnej z CVS, lecz prawdopodobnie w przyszłości będzie stanowił dla niego godną konkurencję.

### **Arch (<http://arch.fifthvision.net/>)**

Arch jest obiecującym, choć znajdującym się jeszcze we wczesnej fazie rozwoju, narzędziem, oferującym całkowicie zdecentralizowane podejście do zarządzania kodem. W przeciwieństwie do programów kontynuujących filozofię CVS, umożliwia każdemu z deweloperów dysponowanie własną kopią centralnego repozytorium i oferuje narzędzia, umożliwiające łatwą integrację repozytoriów. Znacznie lepsze niż w CVS jest w nim wsparcie dla zmian nazw katalogów i plików (do każdego z nich przydzielany jest niezależny od jego nazwy znacznik). Zapewnia atomowość operacji, zaawansowane operacje na gałęziach projektu i automatyczne generowanie plików zmian (ang. *Changelog*). Operacje wykonywane są w nim nie dla indywidualnych plików (jak w CVS), lecz na poziomie całego drzewa projektu. Mechanizmy sieciowe oparte są o standardowe serwery, dostępne w systemach z rodziny Unix (ssh, ftp, http), co ułatwia konfigurację i zmniejsza wymagania sprzętowe. Obecnie istnieją dwie główne wersje Arch – utrzymywana przez oryginalnego autora, Toma Lorda

(<http://arch.fifthvision.net/bin/view/Arch/WebHome>) oraz, zmodyfikowana przez Waltera Landry, ArX (<http://arch.fifthvision.net/bin/view/Arx/WebHome>). Pierwsza z nich wydaje się stabilniejsza, lecz z powodu korzystania ze skryptu sh, znacznie wolniejsza, druga, napisana w C++, jest szybsza i oferuje kilka dodatkowych udogodnień.

### **Stellation (<http://www.eclipse.org/stellation/>)**

Narzędzie oparte na współpracy z zewnętrzną bazą danych. Umożliwia użycie praktycznie dowolnej relacyjnej bazy oferującej język SQL. Zmiany zapisywane są na poziomie całego projektu. Oferuje pełną historię zmian w projekcie, wersjonowanie i zmiany nazw wszystkich artefaktów i w pełni atomowe operacje. Umożliwia wprowadzenie modyfikacji na poziomie linii kodu, a nie, jak zazwyczaj, całych plików.

### **Emacs – rozszerzenia**

Sam w sobie Emacs nie jest narzędziem do zarządzania zmianami, jednakże Emacs 19 zawiera tryb określony jako VC, zwiększa wpływ available from RCS, SCCS, or CVS, oraz zmniejsza kłopoty wynikające z używania tych narzędzi. VC automatycznie, która wersja systemu jest aktualnie wykorzystywana, dokonując autokonfiguracji do używania systemu kontroli. (Systemy można łączyć.) Ukryte w ten sposób zostają szczegóły rejestracji, dostępu i blokowania plików, ukrywając je za jednym prostym poleceniem „wykonaj kolejny, logiczny krok”. VC zawiera również funkcje do przeglądania różnic w wersjach zmian w historii, tworzenia i otrzymywania kolejnych wersji. Wsparty został tryb Dired, który pozwala na wsadowe operacje kontroli wersji na grupach plików.

Dodatkowe informacje można uzyskać, wywołując Emacs 19 i pisząc ``M-x info RETURN m emacs RETURN m vc RETURN'`. 3

### **Aegis**

Aegis jest programem do nadzorowania zmian, opartym na licencji GNU. Jest to raczej narzędzie developera, a nie menedżera. Nie dostarcza mechanizmów śledzenia postępu czy też rozmieszczenie pracy.

Podczas gdy CVS (opisywany poniżej) dostarcza mechanizmy obsługi repozytorium, Aegis dostarcza mechanizmy obsługi repozytorium, linii bazowej oraz obowiązkowych przeglądów i testów. Aegis można tak skonfigurować, aby używał niemalże dowolnego narzędzia do historii (jak np RCS) i niemal dowolnego narzędzia do kontrolowania zależności (np. make).

Najnowsze informacje i wersja Aegis są dostępne pod adresem:

<http://www.canb.auug.org.au/~millerp/>

## **BCS**

BCS = Baseline Configuration System. Jest to system pracujący wyłącznie w systemie UNIX.

## **CVS**

CVS (Concurrent Versions System), który wymaga RCS (powyżej wersji 1.10), rozszerza RCS do kontroli konkurencyjnego edytowania źródeł przez kilku pracowników.

Autor programu podaje następującą analogię: „RCS jakby językiem asemblera, podczas gdy CVS jest podobny do Pascala”. Zaczynając od wersji 1.8, CVS odnotowuje modyfikację każdej linii pliku, z numerem korekty, nazwą użytkownika dokonującego modyfikacji i datą jej przeprowadzenia.

CVS można ściągnąć z

<ftp://ftp.cvshome.org/pub/>.

<http://www.loria.fr/~molli/cvs-index.html>

<http://stud.fh-heilbronn.de/~zeller/cgi/cvsweb.cgi/>

Wersje pod Windows (WinCVS) powinny być dostępne pod

<http://www.wincvs.org/>

## **ICE**

Według autorów ICE (*Incremental Configuration Engine*) jest narzędziem, które dostarcza logiczne wsparcie dla wszystkich dziedzin zarządzania konfiguracją, włączając w to zintegrowane i zunifikowane zarządzanie zmianami i korektami, repozytoria plików binarnych, wnioskowanie o spójności konfiguracji.

Niestety użytkownicy zgłaszają dość liczne problemy związane z zawieszaniem się GUI i z funkcjonowaniem linii poleceń.

ICE nie jest już wspierane, ale jest wciąż dostępne :

[www.cs.tu-bs.de/softech/ice/](http://www.cs.tu-bs.de/softech/ice/)

## **ODE**

<http://www.accurev.com/ode/index.html>

Project Revision Control System (PRCS)

<ftp://XCF.Berkeley.EDU/pub/prcs>

---

<http://www.xcf.berkeley.edu/~jmacd/prcs.html>

PRCS jest oparte na licencji GNU.

## **RCS**

RCS (Revision Control System), oparte na licencji GNU, utrzymuje ostatnią linię bazową i przyrosty poprzednich, co nieco przyspiesza pracę.

<ftp://prep.ai.mit.edu/pub/gnu/rcs/>

<http://www.winsite.com/>

<http://www.fsf.org/order/windows.html>

## **SCCS**

SCCS (Source Code Control System) jest rozpowszechniane z większością dystrybucji systemu Unix.

## **ShapeTools**

Program pod Unix'a.

<ftp://gatekeeper.dec.com/pub/plan/shape/>

<http://swt.cs.tu-berlin.de/~shape/index.html>

## **Ant**

Program oparty na Javie

<http://jakarta.apache.org/ant/>

## **Bake**

<http://bake.werken.com/>

## **Bras**

<http://wsd.iitb.fhg.de/~kir/brashome/>

BuildRef

<http://www.sander.cupertino.ca.us/source.html>

## **Cons**

<http://www.dsmit.com/cons/>

<http://www.baldmt.com/cons-faq/>



## 9.4. Oprogramowanie komercyjne do zarządzania zmianami

Wraz ze wzrostem kosztów wytwarzania oprogramowania coraz więcej firm oferuje autonomiczne narzędzia zarządzania konfiguracją. Poniżej wymieniono programy najczęściej wymieniane przez użytkowników

**+1CM**

+1CM dostarczany przez +1 Software Engineering jest jednym z 14 produktów wspierających +1Environment. Umożliwia pracę wielu użytkownikom nad projektem poprzez sieć. +1CM wspiera wszystkie podstawowe problemy zarządzania konfiguracją, linie bazowe. Zawiera również predefiniowane raporty. Języki wspierane : C, C++, FORTRAN, Pascal, Ada i inne.

<http://www.plus-one.com>

### **2AllChange**

AllChange jest narzędziem do zarządzania konfiguracją i kontroli zmian dystrybuowanym przez Intasoft. Jego cechy to:

- tworzenie wersji, ich śledzenie, odtwarzanie zmian,
- definiowane przez użytkownika cykle życia z automatycznym wyzwalaniem akcji i procedur,
- śledzenie żądań zmian,
- workspaces, shared pools,
- obsługa linii bazowej, wydań, ...
- udogodnienia w raportowaniu/zapytaniach,
- generowanie metryk i graficznych raportów,
- całkowita konfigurowalność (język skryptowy),
- GUI Motif/Windows lub linia poleceń,
- dostępne pod Unix, Windows 3.x, NT i 95,
- wsparcie modelu client/server.

Użytkownicy uważają ten program jako elastyczne narzędzie zarządzania konfiguracją, z dobrym wsparciem ze strony producenta.

<http://www.intasoft.net>

### **ChangeMan**

<http://www.serena.com> Pakiet umożliwiający kontrolowanie konfiguracji sprzętu, platform bazodanowych oraz środowiska developerskiego.



**CM Synergy**

<http://www.telelogic.com/>

**CMF**

<http://www.cmvision.com/>

**Code Co-op**

[http://www.relisoft.com/co\\_op/](http://www.relisoft.com/co_op/)

**CMS and MMS**

[http://www.openvms.compaq.com/commercial/decset/decset\\_index.html](http://www.openvms.compaq.com/commercial/decset/decset_index.html)

**PVCS**

<http://www.qef.com>

**Quma Version Control System (QVCS)**

ftp.clark.net in /pub/jimv/qvcs1625.zip

/pub/jimv/qvcs3225.zip

<http://www.qumasoft.com/>

**RAZOR**

<http://www.razor.visible.com>

**Software Configuration Library Manager (SCLM)**

<http://www.ibm.com/software/ad/ispf/>

**Software Manager**

<http://www.verticalsky.com/solutions/>

**Source Code Manager**

[http://www.unipress.com/free\\_evals/  
eridani.unipress.com/pub/free\\_evals](http://www.unipress.com/free_evals/eridani.unipress.com/pub/free_evals)

**StarTeam**

<http://www.starbase.com>

**TeamConnection**

<http://www.software.ibm.com/ad/teamcon/>

**TeamSite**

<http://www.interwoven.com/>

**TRUEchange**

<http://www.truesoft.com/>

**VisualEnabler**

<http://www.softlabna.com/>

**Visual SourceSafe**

SourceSafe dostarcza mechanizmy kontroli konfiguracji dla rzeczywistych projektów. W 1995 r. SourceSafe został przejęty przez Microsoft i ponownie nazwany. Według działu sprzedaży, Microsoft narzędzia konwersji z takich programów, jak Delta i PVCS. Wersja 4.0 wspiera długie nazwy plików i ścieżki UNC, a tab dialog for setting options, jest w 5 językach, zgodny z designem Windows 95. VSS jest ściśle zintegrowany z Visual Basic, Visual C, Visual Test oraz z Fortran PowerStation. Posiada bardzo miły model do ustawiania wielu wersji projektu. Kluczowe polecenia dotyczą współdzielenia, rozgałęzień, scalania, łączenia i komend ścieżki. Zamiast używać licznych rozgałęzień, tak jak wersja 2.3.6.1 SCCS, logiczne wydania lub nazwy użytkownika mogą być użyte do osiągnięcia tej samej konstrukcji. SourceSafe także pracuje na wielu platformach systemowych, może być użyte w pojęciu opartym na modelu klient/serwer, gdzie kodowanie odbywa się w Visual Basicu pod Windows na komputerze PC Windows PC using Visual Basic i na stacji Unixowej, gdzie się używa C. Microsoft System Journal (maj, 1993 r.) nazwany SourceSafe jest najlepszym narzędziem zarządzania konfiguracją dla systemu Windows. Jedną komendą w SourceSafe

można zrobić „zdjęcie” całego projektu, przypisując jednocześnie nazwę wersji. Ta operacja jest bardzo szybka, nawet jeśli project zawiera 2000 programów. SourceSafe jest zintegrowany z VisualStudio, które automatycznie uzyskuje dostęp do kodu podczas pracy programistów.

Mówi się, że użytkownik może mieć jednocześnie dostęp do kilku projektów w VSS, ale bezpieczeństwo w SourceSafe dobrze dopracowane; posiada tylko 4 bezpieczeństwa: read-only, checkout, add i destroy. To może być niewystarczające dla niektórych projektów. Zostało zgłoszonych wiele przypadków uszkodzeń repozytoriów danych, zwłaszcza dużych.

<http://msdn.microsoft.com/ssafe/>

### **MainSoft Visual SourceSafe for UNIX**

<http://www.opengate.co.uk/opengate/>

### **Metrowerks Visual SourceSafe for Macintosh**

Metrowerks wytwarza wersje Visual SourceSafe na Macintosha. Oprogramowanie to jest w pełni kompatybilne z Microsoft Visual Source Safe. Dodatkowe informacje :

<http://www.metrowerks.com>.

### **Voodoo**

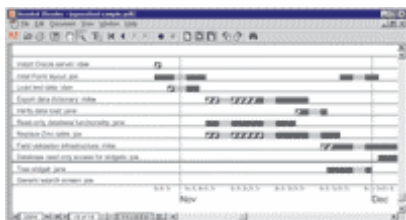
<ftp.swe.uni-linz.ac.at/pub/voodoo>

<http://www.unisoft.co.at/products/voodoooserver.html>

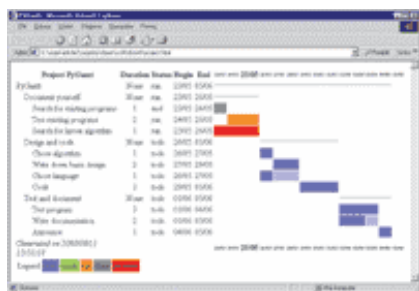
## **9.5. Narzędzia *open source* do zarządzania projektami**

### **Narzędzia open source do zarządzania projektami**

Funkcjonalność dostępnych obecnie produktów open source pozostaje daleko w tyle za produktami komercyjnymi. Darmowe programy OpenSched, PyGantt i QtGantt (patrz ramka Zajrzyj) znajdują się w wersjach bardzo wstępnych i są niedokumentowane. Dwa pierwsze to narzędzia generujące raporty, uruchamiane z odpowiednimi parametrami z wiersza poleceń. QtGantt jest narzędziem wyposażonym w interfejs graficzny. Wszystkie programy są przeznaczone dla platformy Linux, przy czym PyGantt może być też uruchamiany na innych systemach z zainstalowaną obsługą języka Python.



OpenSched 0.1.0 ma największe możliwości, jako jedyny uwzględnia zarządzanie zasobami ludzkimi. Tworzy spory zbiór raportów zawierający: listę zadań, powiązania zasobów ludzkich z zadaniami, zależności pomiędzy zadaniami, listę dni wolnych (poza weekendami), tygodniowe i miesięczne spisy zadań, wykres Gantta. Zestawienia tekstowe są bardzo eleganckie, wykres Gantta – dość słaby. Dane wejściowe program pobiera z pliku tekstowego o dość dobrze przemyślanej strukturze. Na wyjściu jest dokument LaTeX-a, który następnie może być przekonwertowany do formatu PostScript lub PDF. Skrótowy opis formatu danych wejściowych znajduje się w przykładowych plikach wejściowych.



PyGantt 0.6.0 jest narzędziem do generowania wykresów Gantta, napisanym jako skrypt w Pythonie. Przystosowano go wprawdzie do systemu Linux, jednak po niewielkiej zmianie może być również uruchamiany pod Windows – oczywiście po zainstalowaniu obsługi języka Python. Dane wejściowe zapisane są w formacie XML. Struktura pliku nie jest w ogóle udokumentowana; aby ją opanować, należy przeanalizować dołączony przykład. Skrypt generuje na wyjściu niedopracowany dokument HTML zawierający wykres Gantta. Na koniec niespodzianka: PyGantt podczas tworzenia wykresu nie uwzględnia w ogóle dni wolnych(!), dlatego przydatność narzędzia jest na razie żadna.



QtGantt 0.0.7, jako jedyne z wymienionych tu narzędzi oferuje graficzny interfejs wykorzystujący bibliotekę Qt. Program zawiera cztery widoki, z których na razie tylko dwa zostały zaimplementowane: oglądanie wykresu Gantta i podgląd wydruku wykresu. Wykres prezentuje listę zadań z punktami kontrolnymi, informuje o stopniu zaawansowania poszczególnych zadań i porównuje je z linią bazową. Funkcjonalność programu ogranicza się jednak do otwierania plików, prezentacji ich na ekranie i wydruku. Dane wejściowe zawarte są w pliku tekstowym o strukturze bardzo prostej, ale nie do końca przemyślanej – edycja zadań jest uciążliwa. Opis struktury danych wejściowych znajduje się w plikach przykładowych.

Opis wg PCKurier 2/2002 [52].

## Literatura

- [1] Abran A., Robillard P. N., *Identification of the structural weaknesses of Function Point metrics*, 3rd Annual Oregon Workshop on Software Metrics, Portland, Oregon, March 1991.
- [2] Anthony R., *Planning and Control System: A Framework for Analysis*, Harvard Business Review, 1965.
- [3] Badiru A.B., *Project Management in Manufacturing and High Technology Operations*, Willey, New York, 1988.
- [4] Boehm W.B. and all, *Software cost estimation with COCOMO II*, Prentice-Hall, July 2000.
- [5] Bates P., Huws U., *Modeling eWork in Europe Estimates Models and forecasts from the EMERGENCE project*, <http://www.employment-studies.co.uk/summary/summary.php?id=388>.
- [6] Bradley K., *Podstawy metody Prince 2*, Wydane przez: Centrum Rozwiązań Menedżerskich S.A., 00-272 Warszawa, Rynek Starego Miasta 21/21A/9.
- [7] Burton C., Michael N., *Zarządzanie projektem: Jak to się robi w Twojej Organizacji*, Astrum 1999.
- [8] Byzia T., *Szybkie diagnozowanie procesów jako przykład metodyki podnoszenia jakości zarządzania projektem informatycznym*, II Konferencja Project Management – perspektywy i doświadczenia, Stowarzyszenie Project Management Polska (SPMP), Gdańsk, 2000.
- [9] Cegiela R., Zalewski., *Racjonalne zarządzanie przedsięwzięciami informatycznymi i systemami komputerowymi*, Nakom, Poznań 2000.
- [10] Charfield C.S., Jonson T.D., *Microsoft Project 2000, RM 2000*.
- [11] Chrościcki Z., *Zarządzanie projektem – zespołami zadaniowymi*, Wyd. C.H. Beck, Warszawa 2001.
- [12] Cleland D.I., Kimball R.K., *The Strategic Context of Projects*, Project Management Journal, August 1987.
- [13] Chylewska J., *Jak walczyć o najlepszych w firmie, jak zatrzymać kluczowych pracowników*, Materiały firmy. Hewitt Associates:  
[http://was.hewitt.com/hewitt/worldwide/europe/poland/articles/publikacje/jak\\_zatrzyac\\_kluczowych.pdf](http://was.hewitt.com/hewitt/worldwide/europe/poland/articles/publikacje/jak_zatrzyac_kluczowych.pdf).
- [14] De Marco Toom., *The Deadline*, Dorest Hors Publishing, 1997.
- [15] Den J. W. Junior, Junior, Evans J.R., *Total Quality Management*, Organization and Strategy, St. Paul MN, West, 1994.
- [16] Duncan W., *A guide to the project management body of knowledge*, PMI Standards Committee, PA 19082 USA.
- [17] Frączkowski K., Lapkiewicz J., *Zarządzanie wirtualne zasobami projektu informatycznego realizowanego w firmie o strukturze macierzowej*, Materiały VII Konferencji i Warsztatów użytkowników ORACLE. Ploug 2001, Zakopane Kościelisko 23–27.10.2001.
- [18] Frączkowski K., Mechliński T., *Telepraca i zarządzanie wirtualne w projektach informatycznych*, Materiały VIII Konferencji i Warsztatów użytkowników ORACLE. Ploug 2001, Zakopane Kościelisko 22–26.10.2002. [http://www.ploug.org.pl/konf\\_02/materialy/spis.htm](http://www.ploug.org.pl/konf_02/materialy/spis.htm).
- [19] Frączkowski K., Woźniak M., *Wdrożenie systemu informatycznego w szpitalu – aspekty organizacyjne, psychologiczne oraz formalne*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2000.
- [20] Goldratt E.M., *Łańcuch krytyczny*, Wyd. Werbel 2000.

- [21] Grudzewski W.M., Hejduk I., *Sposoby i techniki zarządzania projektem innowacyjnym*, II Konferencja Project Management – perspektywy i doświadczenia, Stowarzyszenie Project Management Polska (SPMP), Gdańsk, 2000.
- [22] Griffin R.W., *Podstawy zarządzania organizacjami*, Wyd. PWN 1996.
- [23] Górski J., *Inżynieria Oprogramowania w Projekcie Informatycznym*, Wyd. Mikom, Warszawa 2000.
- [24] Haywood M., *Managing Virtual Teams: Practical Techniques for High – Technology Project Managers*, Artech House, Boston – London, 1998.
- [25] Hubbard D.G., *Work Structuring*, in P.C. Dismore ed., *The AMA Handbook of Project Management*, AMACOM, New York, 1993.
- [26] IFPUG, *International Function Point Users Group, Function Point Counting Practices Manual*, Release 3.0, IFPUG, Werterille, Ohio, 1990.
- [27] Jaszkievicz A., *Inżynieria Oprogramowania*, Wyd. Helion, Gliwice 1997.
- [28] Jonem C., *Assessment and Control of Software Risks*, Yourdon Press, New Jersey, 1994.
- [29] Kamerlas H., *A Look at Major Planning Methods: Deployment, Implementation, Strengths and Limitations*, Long Rage Planning, August 1978.
- [30] Kućmierowski S., *Przeciwdziałanie zagrożeniom w projektach uruchamiania instytucji typu call center*, II Konferencja Project Management – perspektywy i doświadczenia, Stowarzyszenie Project Management Polska (SPMP), Gdańsk, 2000.
- [31] Lapkiewicz J., Frączkowski K., *High-Availability Infrastructure Architecture*, Web Hosting Transition. Materiały VII Konferencji i Warsztatów użytkowników ORACLE. Ploug 2001. Zakopane Kościelisko 23–27.108.2001.
- [32] Liberatore M.J., *A Decision Support System Linking Research And Development Project Selection with Business Strategy*, Project Management Journal, November 1988.
- [33] Love S.F., *Achieving Problem-Free Project Management*, Wiley, New York, 1989.
- [34] Managelli R.J., Klein Mark N.M., *Reengineering*, Wyd. PWN 1998.
- [35] Mayer M., *The virtual Edge*, Published by: Project Management Institute Headquarters, 1998.
- [36] Meredith J.R., Mantel S.J. Jr., *Project Management, A Managerial Approach*, third edition, John Willey & Sons, Inc., New York, 1995.
- [37] McConnell S., *Rapid Development*, Microsoft Press, 1996.
- [38] Micklelhwait J., Woddrige A., *Szamani zarządzania*, Wyd. WNT, 2000.
- [39] *Murawa Projekt: Project Management: Rola i usytuowanie w przedsięwzięciu projektowym w Polsce i Szwecji*, II Konferencja Project Management – perspektywy i doświadczenia, Stowarzyszenie Project Management Polska (SPMP), Gdańsk, 2000.
- [40] O’Connel F., *How to run successful projects II-the silver bulle*, Wyd. T.J. International Ltd.
- [41] Pniewski K., *Koszty działań pod kontrolą*, PC Kurier nr 22/2000.
- [42] Sajkowicz A., *Zasoby ludzkie w firmie*, Poltex, Warszawa, 2000.
- [43] Stokolski B., *Perspektywy zarządzania projektami wobec nowych wyzwań IT*, II Konferencja Project Management – perspektywy i doświadczenia, Stowarzyszenie Project Management Polska (SPMP), Gdańsk, 2000.
- [44] *Słownik Języka Polskiego*, PWN, Warszawa, 1981.
- [45] Szych J., *Typowe zagrożenia w projektach informatycznych w administracji państwowej*, II Konferencja Project Management – perspektywy i doświadczenia, Stowarzyszenie Project Management Polska (SPMP), Gdańsk, 2000.
- [46] Szyjewski Z., *Analiza strategiczna w tworzeniu systemów informatycznych*, [w:] *Studia Informatica nr 9, Zeszyty naukowe projects II – the silver bulle*. Wyd. T.J. International Ltd.
- [47] Szyjewski Z., *Zarządzanie Projektem Informatycznym*, Agencja Wydawnicza Placet, Warszawa, 2001.
- [48] Tarnowski B.T., *Project Management – elastyczność czy kaftan bezpieczeństwa?*, II Konferencja Project Management – perspektywy i doświadczenia, Stowarzyszenie Project Management Polska

- 
- (SPMP), Gdańsk, 2000.
- [49] Thayer R.H., *Software Engineering Project Management*, Published by the IEEE Computer Society Press, 1987.
- [50] Tyrowicz A., *Od klęski do sukcesów – rozwój zarządzania realizacją projektów informatycznych w administracji celnej*, II Konferencja Project Management – perspektywy i doświadczenia, Stowarzyszenie Project Management Polska (SPMP), Gdańsk, 2000.
- [51] Webster J.L., Reif W.E., Bracker J.S., *The Manager's Guide To Strategic Planning Tools and Techniques*, Planning Review, Nov/Dec 1989.
- [52] Wojtan G., *Przyszłość Project Management w Polsce z perspektywy międzynarodowej*, II Konferencja Project Management – perspektywy i doświadczenia, Stowarzyszenie Project Management Polska (SPMP), Gdańsk, 2000.
- [53] [www.standishgroup.com](http://www.standishgroup.com).
- [54] [www.isbsg.org.au](http://www.isbsg.org.au).
- [55] [www.pckurier.pl/archiwum/art0.asp?ID=5284&haslo=projekt%20informatyczny](http://www.pckurier.pl/archiwum/art0.asp?ID=5284&haslo=projekt%20informatyczny).
- [56] [www.pmi.org](http://www.pmi.org).
- [57] [www.spmp.org.pl](http://www.spmp.org.pl).
- [58] [www.sunset.usc.edu/research/COCOMOII/index.html](http://www.sunset.usc.edu/research/COCOMOII/index.html).
- [59] Yourdon E., *Marsz ku klęsce*, WNT, 2000.
- [60] Zieliński B., *Microsoft Project 98*, Mikom 2000.