



Witold Chmielarz

# ZARZĄDZANIE PROJEKTAMI & ROZWÓJ SYSTEMÓW INFORMATYCZNYCH ZARZĄDZANIA



Wydawnictwo Naukowe  
Wydziału Zarządzania  
Uniwersytetu Warszawskiego



**Zarządzanie projektami  
@ rozwój systemów  
informatycznych  
zarządzania**



# Zarządzanie projektami @ rozwój systemów informatycznych zarządzania

Witold Chmielarz



Wydawnictwo Naukowe  
Wydziału Zarządzania  
Uniwersytetu Warszawskiego

WARSZAWA 2013



Recenzenci

Prof. dr hab. Jerzy Gołuchowski  
Prof. dr hab. Mirosława Lasek

Redakcja

Anita Sosnowska

Projekt okładki

Agnieszka Miłaszewicz

© Copyright by Wydawnictwo Naukowe Wydziału Zarządzania  
Uniwersytetu Warszawskiego  
Warszawa 2013

ISBN 978-83-63962-37-1

ISBN 978-83-63962-38-8 (online)

DOI: 10.7172/2013.wwz.14



Opracowanie komputerowe, druk i oprawa:  
Dom Wydawniczy ELIPSA,  
ul. Inflancka 15/198, 00-189 Warszawa  
tel./fax 22 635 03 01, 22 635 17 85  
e-mail: [elipsa@elipsa.pl](mailto:elipsa@elipsa.pl), [www.elipsa.pl](http://www.elipsa.pl)

# Spis treści

---

<b>Wprowadzenie</b> .....	7
<b>1. Charakterystyka ewolucji metod projektowania systemów informatycznych</b> .....	10
1.1. Ewolucja pojęć i koncepcji zarządzania projektami .....	10
1.2. Determinanty efektywnego zarządzania projektami .....	16
1.3. Przegląd metodologii projektowania systemów informatycznych ....	24
1.4. Kierunki rozwoju tradycyjnych metod projektowania systemów informatycznych .....	29
1.5. Realizacja rozwoju nowoczesnych (zwinnych) metod projektowania systemów informatycznych w praktyce .....	42
<b>2. Przegląd systemów usprawniania procesów biznesowych</b> .....	61
2.1. Metodyka Project Management Institute (PMI) .....	64
2.2. Metodyka Modelu Poziomów Dojrzałości (CMM – <i>Capability Maturity Model</i> ) oraz Zintegrowanego Modelu Poziomów Dojrzałości (CMMI – <i>Capability Maturity Model Integration</i> ) .....	69
2.3. Metodyka Prince2 ( <i>Projects In a Controlled Environment</i> ) .....	71
2.4. Metodyka ITIL ( <i>Information Technology Infrastructure Library</i> ) .....	78
2.5. Pozostałe metodyki wspierające usprawnianie procesów w organizacji .....	85
2.5.1. Metodyka <i>The Open Group Architecture Framework (TOGAF)</i> .....	85
2.5.2. Metodyka <i>Six Sigma</i> .....	88
2.5.3. System zarządzania jakością według standardów ISO .....	95
<b>3. Tendencje rozwojowe systemów informatycznych zarządzania</b> .....	99
3.1. Przegląd literaturowy podejścia do tendencji rozwojowych systemów informatycznych .....	100
3.2. Rozwój systemów informatycznych drogą komplikacji architektury logicznej systemów informatycznych .....	103
3.2.1. Transakcyjne systemy przetwarzania ( <i>Transactional Processing Data Systems – TSP – od 1951 r.</i> ) .....	103

3.2.2. Systemy informacyjne zarządzania ( <i>Management Information Systems</i> – MIS – od 1964 r.) .....	104
3.2.3. Systemy wspomagające podejmowanie decyzji ( <i>Decision Support Systems</i> – DSS – od 1978 r.) .....	107
3.2.4. Systemy informowania kierownictwa (systemy informacyjne kierownictwa – <i>Executive Information Systems</i> – EIS i systemy wspomagania kierownictwa – <i>Executive Support Systems</i> – ESS – od 1980 r.) .....	110
3.2.5. Systemy eksperckie ( <i>Expert Systems</i> – ES – I generacja – od 1978 r., II generacja – od 1985 r.) .....	112
3.2.6. Systemy <i>Business Intelligence</i> ( <i>Business Intelligence Systems</i> ) od 1990 r. ....	114
3.2.7. Integracja „wewnętrzna” systemów informacyjnych zarządzania .....	116
3.3. Rozwój w drodze integracji funkcjonalnej systemów informacyjnych zarządzania .....	117
3.3.1. Zintegrowane systemy planowania zasobów materiałowych ( <i>Material Requirements Planning</i> – MRP – od 1959 r.) .....	118
3.3.2. Zintegrowane systemy planowania zasobów produkcyjnych ( <i>Manufacturing Resource Planning</i> – MRP II – od 1989 r.) ....	119
3.3.3. Zintegrowane systemy planowania zasobów przedsiębiorstwa lub planowanie zasobów na potrzeby przedsiębiorzeń ( <i>Enterprise Resource Planning</i> – ERP – od 1995 r.; ERP II – od 1998 r.) .....	120
3.4. Ścieżka rozwoju drogą infrastrukturalnych rozszerzeń sieciowych – przestrzennych .....	125
3.4.1. Systemy oparte na sieciach prywatnych (1970–1990) .....	126
3.4.2. Systemy oparte na sieciach komercyjnych (1991–1999) .....	127
3.4.3. Systemy oparte na sieci Internet (od 1995 r.) .....	128
3.5. Całościowa integracja systemów na płaszczyźnie portali korporacyjnych .....	131
<b>Zakończenie</b> .....	135
<b>Literatura</b> .....	143
<b>Spis tabel</b> .....	150
<b>Spis rysunków</b> .....	151

# Wprowadzenie

---

Wydaje się czasem, że kwestie tworzenia i wykorzystania systemów informatycznych wspomagających zarządzanie są dla niektórych osób (nawet tych, mających stały kontakt z komputerem) zupełnie oderwane zarówno na płaszczyźnie prywatnej, jak i zawodowej. Spowodowane jest to częściowo sytuacją, która wykształciła się w latach dziewięćdziesiątych ubiegłego stulecia:

- gdy na rynku systemów tradycyjnych dominowały systemy powielarne, a wdrożenia ich najczęściej dokonywano bez głębokich analiz wymagań użytkownika;
- a rynek elektroniczny kształtował się w sposób raczej żywiołowy, bardzo często na zasadzie kopiowania wzorców oraz narzucania przez projektanta ich idei i postaci klientowi.

Powodów należy także upatrywać w zaniedbaniach propagacyjnych – powszechna dostępność Internetu sprawia, że jego użytkownicy, koncentrujący się zwłaszcza na sferze społecznej, zapominają o związku pomiędzy tym co widać na ekranie a koniecznością budowy oprogramowania, które to pokazuje. Nieświadomość sięga jednak niestety głębiej – nie jest rejestrowany również związek pomiędzy oprogramowaniem a koniecznością stworzenia jego projektu. Co więcej, zwrócenie się do przyszłego użytkownika z pytaniem o potrzeby w zakresie oprogramowania skutkuje niekiedy agresją, ponieważ w odczuciu społecznym oprogramowanie przecież jest, wystarczy więc z niego korzystać.

W przypadku dużych, skomplikowanych i innowacyjnych projektów gdzieś na pograniczu stereotypów związanych z informatyką tkwi wiedza o konieczności prowadzenia prac, przez niektórych uważanych za uciążliwe i przeszkadzające w wykonywaniu bieżących obowiązków w organizacji. Mowa tym razem o zarządzaniu projektami informatycznymi – następnym mało rozpoznawalnym przez użytkownika aspekcie informatyzacji firmy. Jednakże najbardziej niezrozumiałe dla większości użytkowników są wzajemne powiązania wymienionych przednio sfer i faktu, że nadzwyczaj dynamicznie nadal się rozwijają. W ostatnich latach ponownie zwrócono uwagę na potrzeby użytkownika końcowego oraz lepsze dopasowanie do funkcjonalności organizacji, a nawet na ekonomiczną efektywność zastosowań systemów informatycznych, co wobec ślepego pędu do informatyzacji za wszelką cenę z początków XXI wieku wydaje się poważnym krokiem naprzód. Ciekawe, że wymienione zagrożenia dotyczą też części pro-



fesjonalistów (środowiska informatyków), którzy zastygli w najlepiej znanych sobie, rozpoznanych i nie zawsze już pasujących do rzeczywistości (nawet nie tylko wirtualnej) rozwiązaniach (bo przecież... *w informatyce wszystko co nowe już było...*).

Mam nadzieję, że prezentowana książka, przedstawiająca syntezę rozważań teoretycznych przeprowadzonych w zakresie dotyczącym aspektów metodycznych i praktycznych tworzenia i rozwoju systemów informatycznych wspomagających zarządzanie, chociaż trochę przełamie wymienione stany nieświadomości, przypomni to, co już było oraz pokaże to, co dopiero będzie.

Biorąc pod uwagę takie założenia, w strukturze pracy uwzględniono trzy zasadnicze części:

- pierwsza – dotyczy kierunków rozwoju zarządzania projektami informatycznymi w zmieniających się dynamicznie warunkach teoretycznych i praktycznych, przełamuje niektóre schematy przyjęte niemal za aksjomaty tej dziedziny;
- druga – zawiera koncepcję powiązania rozwoju metodyk analizy i projektowania systemów informatycznych z rozwojem; z jednej strony technologicznych aspektów rozwoju systemów, z drugiej zaś – systemów informatycznych zarządzania w komercyjnych metodykach zarządzania procesami i projektami;
- trzecia – przedstawia rezultaty działań projektowych; systemy informatyczne wspomagające zarządzanie poprzez autorską koncepcję analizy ich rozwoju i krótką charakterystykę ich wybranych własności.

W pracy więc ukazano i przeanalizowano związek pomiędzy ideologią (teoria „czysta”) a teoretycznymi (wynikającymi z praktyki) i komercyjnymi metodykami zarządzania projektami informatycznymi. Starano się także odpowiedzieć na pytanie: czy, jak, dlaczego i jakimi drogami następował rozwój informatycznych systemów wspomagających zarządzanie i ich wielopłaszczyznowa integracja. Celem poznawczym stały się:

- analiza podstaw rozumienia projektu i procesu oraz różnic i podobieństw w zarządzaniu nimi,
- analiza teoretycznych metod analizy i projektowania oraz komercyjnych metod zarządzania projektami,
- charakterystyka genezy i rozwoju systemów informatycznych zarządzania.

Celami metodologicznymi były natomiast:

- analiza krytyczna metodyk używanych w tym zakresie,
- porównanie metodyk tradycyjnych oraz nowoczesnych,
- wyszczególnienie ich zalet i słabych punktów,
- prezentacja nowego podejścia do koncepcji rozwoju systemów informatycznych wspomagających zarządzanie.

Cele użytkowe prowadziły się zaś do:

- uświadomienia płynności i elastyczności granic pomiędzy zarządzaniem projektami a zarządzaniem procesami,
- analizy potrzeby i wykorzystania metodyk zarządzania projektami,
- poprawnej klasyfikacji i identyfikacji systemów informatycznych w celu właściwego zastosowania odpowiedniej klasy systemów w organizacji.

Osiągnięcie powyższych celów rzutowało na układ logiczny i strukturę pracy, która składa się z wprowadzenia, zakończenia i trzech rozdziałów merytorycznych.

W rozdziale pierwszym skoncentrowano się na zdefiniowaniu podstawowych pojęć związanych z zarządzaniem projektami oraz ich rozwoju w ujęciu klasycznym i nowoczesnym.

W rozdziale drugim dokonano przeglądu najczęściej używanych komercyjnych metodyk usprawniania procesów biznesowych, wykorzystywanych w zarządzaniu projektami: Project Management Institute, Modelu Poziomów Dojrzałości (CMM) oraz Zintegrowanego Modelu Poziomu Dojrzałości (CMMI), Prince II, ITIL, TOGAF, *Six Sigma* oraz zarządzania jakością według standardów ISO.

Rozdział trzeci zawiera autorskie podejście do kierunków rozwoju i integracji systemów informatycznych wspomagających zarządzanie w trzech płaszczyznach: logicznej komplikacji struktury systemów; integracji funkcjonalnej oraz rozwoju rozwiązań sieciowych.

Książka jest monografią przeznaczoną zarówno dla nauczycieli akademickich, jak i studentów informatyki ekonomicznej. Powinna ponadto zainteresować profesjonalnych i nieprofesjonalnych odbiorców ze środowisk nieakademickich. Dla specjalistów z praktyki gospodarczej, zaangażowanych w zarządzanie projektami może ona stanowić cenne źródło wiedzy i ideologiczny przewodnik w ich codziennej pracy. Natomiast menedżerom, mającym sporadyczny kontakt z problematyką kształtowania systemów informatycznych wspomagających zarządzanie w organizacji, może dostarczyć wskazówek, jak należy formułować własne wymagania w stosunku do ich wykonawców.

Praca ta powstała w wyniku pytań i dociekliwości moich studentów przede wszystkim specjalności e-Biznesu oraz studiów podyplomowych Wydziału Zarządzania Uniwersytetu Warszawskiego oraz grona znajomych kierujących na co dzień projektowaniem i wdrażaniem systemów informatycznych głównie analityków, projektantów i szefów projektów. Składam im serdeczne podziękowanie i jednocześnie mam nadzieję, że przynajmniej część ich wątpliwości udało mi się w zadowalający sposób wyjaśnić.

Ostateczną ocenę pozostawiam Czytelnikom.

# 1

## Charakterystyka ewolucji metod projektowania systemów informatycznych

---

### 1.1. Ewolucja pojęć i koncepcji zarządzania projektami

W klasycznej postaci termin zarządzanie projektami występuje w dwóch zakresach znaczeniowych – wąskim oraz szerokim. W wąskim, praktycznym zakresie „...można je określić jako zespół działań kierowniczych związanych z realizacją projektów oraz zbiór wykorzystywanych w tych działaniach zasad, metod i środków...” (Kruger, 1993, s. 3359; za: Trocki, Grucza i Ogonek 2003, s. 27). „...Tradycyjnie zarządzanie projektami postrzega się jako planowanie, tworzenie harmonogramu oraz kontrolę projektu w celu osiągnięcia jego zamierzeń...” (Bukłaha, 2010, s. 21). Oznacza więc ono de facto „...zintegrowany i niepowtarzalny zbiór czynności informacyjno-decyzyjnych wykonywanych dla osiągnięcia wymaganych zakresem określonego projektu celów, ograniczonych środkami według określonych metodyk używanych w ramach wybranych metodologii z przypisanymi im technikami realizacji zadań szczegółowych...” (Chmielarz i Klincewicz, 2010, s. 239). Zintegrowany – ponieważ musi doprowadzić do realizacji celów określonych w projekcie. Niepowtarzalny, ponieważ wynika to z niepowtarzalności i odrębności zadań projektowych. Metodologia (jako nauka o metodach badań naukowych, ich skuteczności i wartości poznawczej) występuje tu w charakterze swoistego poradnika, zbioru wytycznych i reguł przedstawiających co należy zrobić w danym momencie z projektem, ale już nie pokazuje w jaki konkretny i dostosowany do danego przypadku sposób (zob. Pieter, 1967). Prezentuje to natomiast metoda – rozumiana jako ogół czynności i sposobów niezbędnych do rozwiązywania problemów badawczych, pozwalający później (w ramach określonej metodyki) na pełną formalizację projektu, a odpowiadający na pytanie, w jaki sposób należy osiągnąć cele projektu. Metodyka ukazuje dokładną, formalną procedurę osiągnięcia celu za pomocą zbioru odpowiednich dla rozwiązania problemu badawczego narzędzi i technik szczegółowych. Technika szczegółowa sprowadza się do przedstawienia proceduralnego użycia narzędzi, za pomocą których metoda jest realizowana.

Z szerokiego, teoretycznego punktu widzenia zarządzanie projektami jest określoną dziedziną naukową, opartą na rozwiązaniach teoretycznych problemów praktycznych wynikających „...z konieczności zaspokojenia zanalizowanych wymogów zleceniodawcy za pomocą dostępnych umiejętności, wiedzy, metod, technik i narzędzi realizacyjnych...” (Mingus, 2009, s. 21; za: Project Management Institute, 2004). Jest to więc nauka o skutecznym osiągnięciu zakładanych celów (działaniu) za pomocą racjonalnego użycia zasobów (ludzkich, finansowych, materialnych itp. i relacji między nimi) w zakładanym czasie. Charakterystyczna dla tej dziedziny naukowej jest jej zmienność, związana z dynamiką otoczenia gospodarczego. Jeśli ma ona być odpowiedzią na bieżące problemy gospodarcze, to powinna cały czas dostosowywać się do możliwości ich rozwiązywania. A to już niekoniecznie jest spójne z klasycznym rozumieniem pojęcia zarządzania projektami i raczej ewoluuje w kierunku zarządzania procesami.

Podmiotem zarządzania projektami, a jednocześnie jego podstawową składową występującą we wszystkich powyższych definicjach jest bowiem niewątpliwie projekt. Istnieje wiele bardzo podobnych, a niesprzecznych ze sobą definicji projektów, wyróżniających zasadnicze charakterystyki tego pojęcia. Autorzy definicji zgadzają się co do zawartości treściowej, a dopiero później w cyklu realizacji projektów zgłaszają postulaty, które sprawiają, że ich podejścia do projektów wydają się być nieco różne. Niemniej jednak – niezależnie od stosowanych definicji – istotą projektu jest skuteczna i metodyczna realizacja określonego przedsięwzięcia rozumianego jako złożony ciąg, niekoniecznie sekwencyjnych, a skomplikowanych działań, prowadzonych zgodnie z założonym planem. Jednocześnie z analizowanych definicji wynika, że projekt to nowatorskie, unikalne, niepowtarzalne, złożone, grupowe, koordynowane przedsięwzięcie ograniczane ramami czasowymi i finansowymi, ukierunkowane na realizację sprecyzowanego celu, o określonym z góry zakresie, często strategicznym i wysokim na ogół wymogu jakości. Realizacja projektu następuje za pomocą określonych metod i zawartych w nich technik stosowanych z pełną premedytacją lub w sposób intuicyjny, przeważnie według przypisanego jego etapom kosztorysu, zgodnie z wyspecyfikowanym harmonogramem i wymaganą przepisami dokumentacją. Zasięg tego pojęcia jest dziś bardzo szeroki – od tworzenia nowych obiektów (rzeczy, dóbr, zbiorów informacji), modernizacji istniejących, dodania nowych wartości lub rozszerzenia zakresu uprzednio zdefiniowanych obiektów, po niszczenie (wymianę na nowe) starych obiektów lub działania składające się ze złożenia tych operacji.

Generalnie ujmując, cechy współczesnych projektów są następujące:

- celowość,
- złożoność,
- unikalność,
- determinizm,

- ryzyko,
- autonomiczność.

**Celowość** – działanie ukierunkowane na uzyskanie określonych dla wykonawcy lub zamawiającego wyników. Należy zauważyć, że niekiedy zamawiający uczy się w trakcie projektu i domaga się zmodyfikowania celu projektu, nawet poza ramy wyznaczone podpisanymi umowami, w związku z czym istotne może się tu stać nadażanie za zmieniającym się celem.

**Złożoność** – oczywiście można zarządzać nawet projektami prostymi w celu ich prawidłowej realizacji, nie jest to jednak na ogół działanie tak skomplikowane, by wymagało specjalnych metod czy technik wykonawczych. Dlatego pojęciem projektu opatruje się zwykle przedsięwzięcia na tyle złożone, by jedna osoba nie była w stanie objąć swoim działaniem całości projektu, dotyczące konieczności koordynacji funkcjonowania wielu osób, zespołów, jednostek organizacyjnych, a nawet organizacji. Złożoność potęgowana jest postępowaniem techniczno-organizacyjnym, szybkie zmiany w tym zakresie wymagają często dostosowania do nich (poprzez zmianę) założeń projektowych.

**Unikalność** – to cecha szczególna – nowatorskość, niepowtarzalność conceptualna oraz wykonawcza, jest traktowana w rzeczywistości w sposób dość elastyczny. Czasami jako obiekt unikalny traktuje się obiekt oryginalny po pewnych modyfikacjach lub zastosowany w innej dziedzinie (branży). W związku z tym mówi się czasami o projektach standardowych, powielalnych, gdzie innowacyjność i niepowtarzalność zawiera się w generowaniu specyficznych parametrów projektu i unikalnych, tworzonych dla jednego i tylko jednego zamawiającego. Realizacja tych ostatnich jest wielokrotnie droższa niż standardowych, powielalnych projektów. Inne interpretacje mówią o jednorazowości samego projektu, później zaś w nowych projektach wykorzystywane są wiedza i doświadczenie nabyte dzięki uczestnictwu w poprzednich.

**Determinizm** – określoność w czasie (przestrzeni), zakresie i budżecie. Budżet, zakres i czas realizacji pozostają między sobą w określonych relacjach, ograniczonych zasobami. Nieuzasadnione naruszenie tej szczególnej równowagi (trójkąta projektu) wpływa na zmianę pozostałych parametrów. W skład zarządzania projektami wchodzi też zarządzanie ryzykiem, zarządzanie w warunkach niepewności itd. Istnieją też specjalne metody rozwiązywania tych zagadnień. Trzeba również zdawać sobie sprawę, że w rzeczywistości gospodarczej jasno określonych sytuacji jest stosunkowo mało, a zarządzając projektami, odnosi się jedynie do zdeterminowanych modeli zdarzeń na ogół losowych czy obarczonych ryzykiem. Projekty, zwłaszcza duże i innowacyjne, na ogół przekraczają czas trwania i zakładany poziom budżetu i nie dotrzymują zakresu, ryzyko ich realizacji jest więc stosunkowo duże. W ostatnich latach zaczęto zwracać uwagę na czwartą składową – wymagania użytkownika końcowego (łatwość posługiwania

się rozwiązaniami projektowymi, duża szybkość uzyskania wyniku końcowego, dokładność odwzorowań wymagań użytkownika – nawet kosztem częściowego pogorszenia jakości i uzyskania wyników dopuszczalnych zamiast optymalnych), co znacznie zwiększa obszar manipulowania wariantami realizacji projektu.

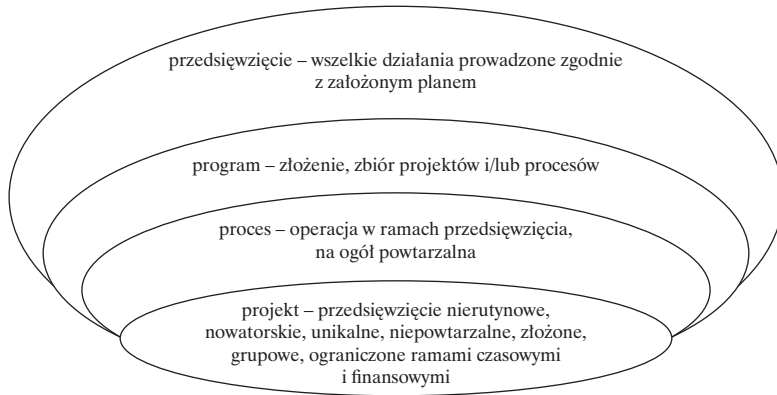
**Ryzyko** – związane z trudnością realizacji – skoro jest to na ogół przedsięwzięcie złożone, skomplikowane, nowatorskie i w związku z tym obarczone dużymi kosztami, istnieje wiele czynników, które mogą zakłócić prawidłowy cykl realizacji projektu. Ponadto metody i techniki wspomagające procesy decyzyjne występujące w zarządzaniu projektami często oparte są na modelach, a więc pewnym uproszczonym odzwierciedleniu rzeczywistości gospodarczej. Jeżeli to odzwierciedlenie jest nieprawidłowe, ryzyko realizacji wzrasta pomimo matematycznej doskonałości i sprawdzalności metod i technik. Wystarczy bowiem, że tylko jedno z założeń początkowych nie zostanie realizowane, a przydatność wyniku końcowego jest już mocno wątpliwa. Wydaje się jednak, że dla decydentów nawet taka przybliżona informacja ma wartość większą niż jej brak.

**Autonomiczność** – całkowita (np. outsourcing w różnych postaciach) lub częściowa niezależność od struktur organizacyjnych wykonawcy lub zamawiającego. Oznacza tu na ogół wydzielenie w pewien sposób operacji projektowych z rutynowych zadań tych organizacji. Pojęcie nierutynowości dopuszcza więc postać projektu realizowanego wewnątrz danej organizacji, chociaż jego złożoność wskazywałaby na konieczność realizacji również za pomocą sił i środków zewnętrznych.

Podsumowując, w ujęciu klasycznym zarządzanie projektem jest to wyjątkowe, jednostkowe przedsięwzięcie podejmowane w celu wytworzenia ilościowo i jakościowo określonego unikatowego produktu lub usługi, za pomocą przydzielonych zasobów ludzkich, materialnych i finansowych, ograniczone w czasie jednoznacznie zdefiniowanym punktem startowym i końcowym, które są połączone etapami realizacji wyodrębnionymi w ramach określonej metody.

Obecnie, jak wynika z powyższych rozważań, determinizm, jednoznaczność i statyka w określeniu cech i wyników projektów przesuwają się w kierunku probabilistyki, nieokreśloności i dynamizmu. Teoretycznie powiększać się więc powinna rozpiętość pomiędzy dwoma podstawowymi rodzajami działań różniami we współczesnej organizacji: projektami i procesami. Projekty przecież zdefiniowano jako przedsięwzięcia unikalne, jednorazowe, wymagające odpowiedniego przygotowania, podczas gdy procesy są powtarzalne i mogą podlegać automatyzacji lub stawać się działaniami rutynowymi. Główną różnicą jest to, że procesy są wykonywane stale i są z natury powtarzalne, natomiast projekty są wykonywane w momencie powstania nowych potrzeb, a każdy z nich jest zupełnie inny. Ale w zasadzie, w pewnym sensie projekty są podzbiorem procesów – są to wszystkie procesy, które możemy określić jako nierutynowe (ukierunkowane na zmiany), innowacyjne, pragmatyczne, obciążone dużym

ryzykiem i niepowtarzalnością (zob. rys. 1). Wynika to ze swoistych podobieństw – oba rodzaje działalności są wykonywane przez wytypowane do tego zespoły, determinowane określonymi i ograniczonymi w czasie zasobami, na zasadzie planowania, sterowania, nadzorowania i kontrolowania poszczególnych czynności.



**Rysunek 1.** Relacje pomiędzy projektami, procesami a przedsięwzięciem i programem

Źródło: opracowanie własne.

To z kolei sprawia, że zmiany w zarządzaniu procesami wpływają bezpośrednio na zarządzanie projektami. Projekty są wykonywane w celu doskonalenia istniejących procesów, tworzenia zupełnie nowych procesów oraz rozwiązania określonych problemów związanych z koniecznością zmian procesów. W każdej organizacji występują zarówno działania procesowe, jak i projektowe. Wbrew swojej klasycznej definicji projekty w zasadzie nigdy się nie kończą. Każdy koniec jednego projektu oznacza początek następnego (np. w informatyce: prace przedwdrożeńowe, a potem wdrożenie), w sumie stanowią one czasami niekończący się cykl projektów, których nawet nie sposób nazwać podprojektami, ponieważ nigdy, chociażby ze względu na niepewność i wysokie ryzyko, nie wiemy w jakim kierunku rozwiną się wymogi ostatecznych użytkowników. Ale najbardziej symptomatyczne dla rozwoju projektów jest to, że w gruncie rzeczy metodyki zarządzania projektami były w klasycznej wersji tworzone, uogólniane, „usztyniane”, ujednoczane itp., aby jak najlepiej znormalizować procesy występujące w projekcie i zminimalizować prawdopodobieństwo zajścia zdarzeń nieprzewidzianych, mogących doprowadzić do niepomyślnego zakończenia projektu (ryzyko). Paradoks, jaki z tego wynika, polega więc na tym,

że zarządzanie projektami coraz bardziej zbliża się do metodyk zarządzania procesami, dąży bowiem do operowania standardowymi regułami rozwiązywania niestandardowych problemów, które stara się za pomocą specyficznych metod i technik zestandaryzować (czyli zamienić w procesy) przez daleko idącą formalizację. Odnosi się więc wrażenie, że występujące w literaturze rozgraniczenie (zob. tab. 1, kolumna pierwsza) trąci, zwłaszcza obecnie, sztucznością i nie oddaje w pełni rzeczywistości.

**Tabela 1.** Podobieństwa i różnice pomiędzy projektami i procesami

Projekt tradycyjnie	Proces	Projekt współcześnie
Dynamizm, zarządzanie zmianami	stabilność w sensie powtarzalności	dynamizm sterowany ekonomicznym pragmatyzmem
Unikatowość	rutynowość	dobrze praktyki zarządzania
Pragmatyzm	automatyzm	zarządzanie wiedzą
Zarządzanie zmianami (rewolucja)	modyfikacje (ewolucja)	nadążanie za zmianami
Zagrożenie realizacji ryzykiem	niewielkie ryzyko	minimalizacja ryzyka
Innowacyjność, nowatorstwo	tradycjonalizm postępowania	niekonwencjonalność kontrolowana wzorcami
Zaangażowane kierownictwo	brak wpływu kierownictwa na przebieg procesów	współpraca zespołów z kierownictwem
Konflikty w organizacji	współpraca w organizacji	kooperacja w organizacji w celu minimalizacji konfliktów

Źródło: opracowanie własne.

Czasami zamiennie z pojęciem projektu stosuje się też pojęcie programu albo jako projektu w dziedzinach niekomercyjnych, albo jako wiązki projektów (projekt bardzo złożony, drogi, ryzykowny, skomplikowany itd.). Od pewnego czasu definicja programu ewoluuje w kierunku „...uporządkowanego zbioru współzależnych projektów, które są zarówno pożądane i konieczne, jak i wystarczające do osiągnięcia efektu biznesowego i dostarczenia wartości oczekiwanych przez sponsorów programu...” (Pańkowska, 2010, s. 244). Programy składające się z wielu projektów w odróżnieniu od nich nie są ograniczone w czasie. Może więc jednak jednym z kierunków rozwoju projektów i zarządzania nimi są programy? A może problem tkwi w niedostosowaniu wąskich ram narzucanych na klasyczne pojęcie projektu do teorii i praktyki zarządzania projektami?



## 1.2. Determinanty efektywnego zarządzania projektami

W klasycznym zarządzaniu projektami uważa się, że podstawowymi determinantami projektu są czas, koszty i zakres. Czas realizacji wynika ze swoistego „przywiązania” do typowych metod rozwiązywania problemów zarządzania projektami, gdzie podanie początku i końca projektu było warunkiem niezbędnym dla uzyskania wyników w stosowanych metodach (zwłaszcza sieciowych). Koszty projektu związane są z dysponowanymi zasobami. Zakres zaś dotyczy celu i ograniczeń dziedziny, której projekt dotyczy. Wymienione trzy elementy stanowią tzw. złoty trójkąt zależności w zarządzaniu projektami (rozpowszechnione przez: Kerzner, 2005a), w którym poszczególne parametry są powiązane tzw. równaniem równowagi. Obszar pomiędzy wierzchołkami trójkąta (określonymi czasem, kosztami i zakresem) zawiera rozwiązania dopuszczalne dla wykonawcy projektu, ale nie optymalne. Optymalne dla każdego projektu są wielkości graniczne wyznaczone kombinacją wartości wierzchołków tego trójkąta. W ten sposób ustalenie dwóch wielkości sprawia, że trzecia jest wynikowa, zmiana jednej z nich powoduje konieczność zmiany pozostałych. Tak więc skrócenie czasu realizacji oznacza zwiększenie kosztów lub zmniejszenie zakresu projektu. Zwiększenie zakresu może spowodować podniesienie kosztów i/lub dłuższy czas realizacji. Zmniejszenie kosztów powoduje na ogół zmniejszenie zakresu i wydłużenie czasu realizacji. Niekiedy dodaje się do tej trójki jakość i koszty (zob. Kisielnicki, 2011; Schmidt, 2010). Dodanie jakości jako wielkości wypadkowej w stosunku do realizacji pozostałych parametrów wydaje się uzasadnione, ponieważ kombinacja zakresu projektu, jego kosztu i czasu stanowi o jakości projektu. Koszty realizacji traktowane przez większość autorów zamiennie z planowanymi kosztami w budżecie, można tu pominąć.

Jednakże wydłużenie czasu realizacji nie musi oznaczać zmniejszenia kosztów lub zwiększenia zakresu projektu. Przyczyny i skutki braku zwrotności tej relacji leżą po stronie ostatecznego użytkownika-odbiorcy projektu. Dlatego pojawia się postulat dołączenia do tej trójki podstawowych parametrów, czwartego – wymagań odbiorcy<sup>1</sup>. To podejście zwiększa zakres dostępnych rozwiązań i pomnaża ilość potencjalnych kombinacji zdarzeń, które mogą zajść w trakcie realizacji projektu. Wyjaśnia też szereg zjawisk zachodzących w trakcie realizacji projektów w rzeczywistości gospodarczej.

Przeprowadzone w latach 1994–2012 przez The Standish Group International badania dotyczące założonej realizacji projektów pokazują, że współczynnik sukcesu w tych przedsięwzięciach utrzymywał się w granicach 16–37% (zob. tab. 2). Oczywiście wysoki stopień współczynnika niepowodzeń spowodowany

<sup>1</sup> Spotyka się też, próby dołączenia do tego podstawowego zestawu parametrów jakości, niekiedy kojarzonej z zakresem prac oraz ryzyka towarzyszącemu projektom, ale nie wydaje się to w tym przekroju uprawnione.

jest przyjęciem dość restrykcyjnego założenia, że niepowodzenie jest każdym odchyleniem od niezrealizowania podstawowych parametrów projektu (np. przekroczenia budżetu, niedotrzymania terminów, niezrealizowania zakresu prac), ale i tak procent niepowodzeń jest zastanawiająco wysoki. I niestety z czasem specjalnie nie maleje – pomimo postępu technologicznego utrzymuje się wciąż na poziomie około 35%.

**Tabela 2.** Średnia skala realizacji projektów analizowanych przez The Standish Group w wybranych latach z okresu 1994–2010

Rok	Współczynnik sukcesu	Współczynnik niepowodzenia
1994	16%	84%
1996	27%	73%
1998	26%	74%
2000	28%	72%
2002	34%	66%
2003	33%	67%
2004	29%	71%
2006	35%	65%
2008	32%	68%
2010	37%	63%

Źródło: opracowanie własne na podstawie: Johnson, 2005, s. 11; The Standish Group International, 2007; 2009, s. 1; 2009a.

Jeżeli porównujemy z kolei przyczyny tak wysokich niepowodzeń, to wśród czynników wpływających na utrudnienia realizacji projektu wyróżnia się zespół parametrów miękkich, związanych z czynnikiem ludzkim (stanowią one aż 40% wszystkich determinant niepowodzeń klasyfikowanych w pierwszej dziesiątce zarówno w 2000 r., jak i 2008 r.). O połowę mniej i na gorszych pozycjach w klasyfikacji występuje czynników ściśle technologicznych (zob. tab. 3).

**Tabela 3.** Czynniki utrudniające realizację projektu

Lp.	Czynnik utrudniający realizację projektu	Procent odpowiedzi	
		miejsce 2000	miejsce 2008
1.	<b>Brak zaangażowania użytkowników biznesowych i informacji od nich</b>	2	1
2.	Niekompletne wymagania biznesowe i funkcjonalne lub ich zmiana	6	2
3.	<b>Niedoświadczony kierownik projektu</b>	4	3

cd. tab. 3

Lp.	Czynnik utrudniający realizację projektu	Procent odpowiedzi	
		miejsce 2000	miejsce 2008
4.	Brak wsparcia kierownictwa firmy	1	4
5.	Brak kompetencji technologicznych	5	5
6.	<b>Brak zasobów do realizacji projektu (ludzkich)</b>	8	6
7.	Nierealne oczekiwania od zespołu realizatorów	10	7
8.	Niejasno określone wymagania (cele)	3	8
9.	Nierealny harmonogram projektu	9	9
10.	Częsta i radykalna zmiana technologii	7	10

Źródło: opracowanie własne na podstawie: Johnson, 2005, s. 11; The Standish Group International, 2009a.

Inne podejście do czynników determinujących realizację projektu wyraża raport zatytułowany *The Silence Fails*, będący podsumowaniem badań przeprowadzonych przez firmę szkoleniową VirtualSmarts i firmę The Concours Group (VirtualSmarts, 2006; 2006a).

Menedżerowie projektów zidentyfikowali i scharakteryzowali pięć krytycznych obszarów stanowiących największe ograniczenia w osiągnięciu sukcesu (Galant-Pater, 2009; Waszczuk, 2008):

- nieprawidłowo przeprowadzona analiza wymagań projektu – spowodowana brakiem wywiadów i dyskusji z wykonawcą na temat doświadczeń płynących z wcześniej wykonanych projektów, które mogą być wykorzystane w bieżącym projekcie, na wczesnym etapie realizacji projektu; na ogół pierwsze spotkanie zespołu projektowego rozpoczyna dyskusję nad gotowym harmonogramem, z określonymi zasobami i planowanymi terminami ich wykorzystania;
- brak wsparcia sponsora projektu – zleceniodawca stara się minimalnie angażować w realizację projektu; brak tej współpracy powoduje nieprecyzyjne rozpoznanie informacyjno-organizacyjne problemu, na podstawie którego trudno stworzyć prawidłową specyfikację rozwiązania problemu i modelowania procesów, które do tego doprowadzą; projekty opóźniają się więc coraz bardziej i/lub nie spełniają wymagań sponsorów; problemów tych można uniknąć, dzięki regularnie organizowanym spotkaniom bezpośrednich wykonawców ze zleceniodawcą;
- brak wiedzy o stanie projektu – ukrywanie problemów zaistniałych w zespole przed przełożonymi w obawie przed ich reakcją i próby rozwiązywania ich we własnym zakresie lub ominięcia, przerzucanie na innych informacji o krytycznych zdarzeniach w projekcie;

- unikanie odpowiedzialności za priorytetowe zadania projektu – rozpoczynanie realizacji projektu (wbrew harmonogramowi) od najłatwiejszych i niekoniecznie najważniejszych zadań w projekcie oraz ignorowanie na początku zadań najtrudniejszych, a priorytetowych, które niosą ze sobą ryzyko niepowodzenia; to może prowadzić do braku koordynacji w skali całego projektu;
- przerzucanie odpowiedzialności – za zadania, których członkowie zespołu nie chcą lub nie są w stanie wykonać z powodu braku wiedzy lub praktyki; tworzenie sztucznych barier (...*to niemożliwe, to nieprofesjonalne, to będzie dużo kosztowało...*), które mają pozwolić na przerzucenie zadań na innych bądź wręcz zaniechanie ich.

Przedstawione badania dowiodły, że w przypadku braku analizy i odpowiedniej reakcji na powyższe sytuacje w stosunku do prowadzonego projektu prawdopodobieństwo niepowodzenia tego projektu (określone jako przekroczenie zaplanowanego budżetu i czasu oraz niespełnienie wszystkich wymagań klienta co do zakresu, jakości i funkcjonalności wytworzonego produktu) wzrasta do 85%. Jeśli zaś działania zaradcze dotyczące „krytycznych obszarów” się powiodą, to prawdopodobieństwo porażki spada o 50–70%.

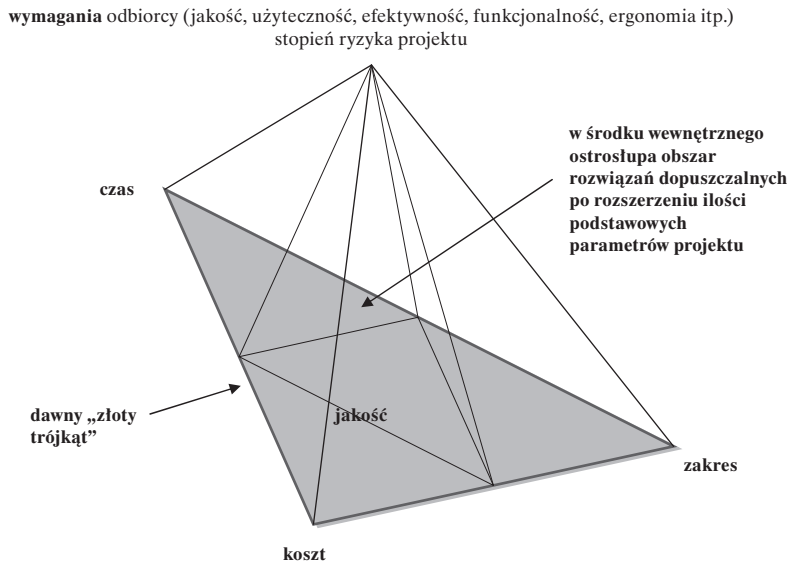
The Standish Group International (2007), opierając się na badaniach własnych, opublikowała listę dziesięciu najważniejszych czynników przyczyniających się do sukcesu projektu:

- zaangażowanie klienta w realizację projektu,
- wsparcie kierownictwa (sponsora) projektu,
- jasny cel biznesowy projektu (sprecyzowane wymagania w świetle istniejących ograniczeń),
- zoptymalizowany zakres projektu (dostosowany do możliwości wykonania),
- metodyka zwinnego projektowania (agile) w miejsce tradycyjnego,
- doświadczony i kompetentny kierownik projektu,
- prawidłowe zarządzanie budżetem projektu,
- wykształcone zasoby ludzkie,
- formalna metodyka prowadzenia projektu,
- standardowe narzędzia programistyczne i infrastruktura.

Wśród czynników przyczyniających się do sukcesu podobny procent, jak przy niepowodzeniach, zależy od czynnika ludzkiego (tylko 20% dotyczy czynników związanych z technologiami). Jak można zauważyć, postęp technologiczny zdaje się nie mieć aż tak znaczącego wpływu na realizację zarządzania projektami, jak by wynikało to ze środków zaangażowanych w jego rozwój.

Z powyższych powodów pojęcie sukcesu projektu ewoluje obecnie w kierunku oceny wybiegającej poza klasyczny, a wąski trójkąt równowagi pomiędzy kosztami, czasem i zakresem. Przyjmuje się punkt widzenia użytkownika-odbior-

cy projektu i jego sposób widzenia zarówno w projektach wewnętrznych (w których realizujący projekt i jego odbiorcy są pracownikami tej samej organizacji, w której pozostaje produkt ostateczny projektu), jak i zewnętrznych (produkty wykonywane dla klientów spoza organizacji i stanowiące źródło przychodów organizacji je wytwarzających). Już samo rozszerzenie „złotego trójkąta” o czwarty parametr powoduje również poszerzenie możliwości podejmowania decyzji w zakresie jego realizacji (każda decyzja jest opisywana uporządkowaną już czwórką parametrów (czas, zakres, koszty, wymagania), a nie jak poprzednio – trzema. Przy czym relacje pomiędzy parametrami stają się, jak się wydaje, nierównoważne – wymagania odbiorców są nadrzędne w stosunku do pozostałych parametrów. Jeżeli natomiast dodatkowo wprowadzimy piąty parametr, którym jest jakość – ilość dopuszczalnych rozwiązań znów się zawęzi, wpłynie na to wielowymiarowość realizowanego projektu i ścisły związek wymagań użytkownika z określonym poziomem jakości. Nie wszystkie rozwiązania dopuszczalne projektem i zgodne z wymaganiami użytkownika mogą spełniać zakładane standardy jakości (zob. rys. 2).



**Rysunek 2.** Obszar dopuszczalnych kombinacji podstawowych parametrów projektu i jego rozszerzenie

Źródło: opracowanie własne.

Na zmiany relacji pomiędzy zarządzaniem projektami a zarządzaniem procesami ma więc wpływ ich otoczenie (środowisko). Środowisko, w którym pro-

Projekty są realizowane dzieli się na (zob. Stępień, 2012): ekonomiczne (ceny, cła, podatki, kursy walutowe, stopy procentowe, polityka gospodarcza, rynki, stopień rozwoju gospodarczego), prawne (system prawny, jego dostosowanie do warunków realizacji, licencje), technologiczne (rozwój technologiczny, stan technologii w organizacji, normy jakości), organizacyjne (struktury organizacyjne, styl kierowania, umiejętności i wiedza kadry kierowniczej i pracowników, funkcjonalność organizacji, metoda zarządzania projektem), psychologiczne (kultura, opór przeciwko zmianom, stopień innowacyjności, bezpieczeństwo realizacyjne i wykonawcze) oraz polityczne (uwarunkowania geopolityczne, trendy rozwojowe, sojusze, moda).

Pojawia się tutaj druga istotna sprawa. Sukces projektu w ujęciu klasycznym i sukces projektu w ujęciu współczesnym (oraz zarządzania nim) wynikający z praktyki zdecydowanie się różnią. W ujęciu klasycznym (w ten sposób traktuje go wiele badań) sukcesem jest nieprzekroczenie kosztów (a najlepiej ich wykonanie), pełna zgodność harmonogramu z terminami wykonania oraz zgodność wykonanego zakresu prac z wyspecyfikowanym w projekcie. Dołączenie optyki ostatecznego użytkownika (odbiorcy, klienta) oznacza już dodanie do kryteriów oceny sukcesu: kwestii zadowolenia klienta z uzyskanego produktu lub usługi. Dołączenie dynamicznego otoczenia – zmniejszenie ryzyka niepowodzeń, sprawności, efektywności, elastyczności, adaptacyjności, funkcjonalności itp. A to już są oceny bardzo bliskie ocenie sukcesu prawidłowego zarządzania procesami w organizacji. Analiza wyników badania uzyskanych przez The Standish Group International wskazuje na bardzo praktyczne determinanty sukcesu. Rezultat wykonania projektu w potocznym odczuciu może być krańcowo różny od tego, który wynika z ujęcia klasycznego. Przeciętnie prawie trzy czwarte projektów wykonywanych w ostatnich latach przekraczało koszty, nie dotrzymało terminów ani zakresu prac, a mimo to zostały one uznane za sukces zarówno przez wykonawców, jak i zamawiających (klientów, odbiorców, użytkowników). W przeprowadzanych z nimi rozmowach twierdzili oni, że najważniejszą dla nich sprawą był fakt wykonania projektu, który stanowił dla nich podstawę do dalszego działania (nierzadko rozpoczęcia nowego projektu lub jego modyfikacji). Może się też zdarzyć sytuacja, kiedy za sukces zostanie uznany projekt, który przekroczył koszty i harmonogram, jednak zrealizował zwiększony zakres, za porażkę zaś taki, w którym te dwa parametry zostały dotrzymane, a nie dotrzymano zakresu. Oczywiście nierzadko zdarzają się też sytuacje, kiedy ocena wykonawcy projektu różni się krańcowo od oceny klienta, ale coraz częściej wybiega poza kanon niedotrzymania trzech podstawowych parametrów. Jeżeli zarządzanie projektem odnosi sukces, nie oznacza to, że był nim projekt, może oznaczać natomiast, że było nim dokonanie w istotnym dla projektu momencie odchylenia od założeń projektowych (zarządzanie zmianą).

Na każdym etapie planowania i realizacji projektu, przedsięwzięcie takie może być postrzegane przez niektórych interesariuszy jako szansa, a przez innych jako zagrożenie. Dlatego też może dochodzić do rozgrywek organizacyjnych, które swój początek będą miały zapewne jeszcze przed rozpoczęciem projektu (dotyczące np. tego czy projekt jest nam w ogóle potrzebny – wstępne definiowanie) i nie ustaną nawet po zakończeniu projektu (gra o to czy projekt zostanie uznany za sukces, czy za porażkę).

Przykładowa lista kryteriów sukcesu może – biorąc powyższe pod uwagę – mieć następującą postać (zob. też: <http://zarzadzanie-projektami-it.pl/sukces-projektu/>):

- osiągnięcie celów projektu;
- zadowolenie klienta, odbiorców (użytkowników), wykonawców;
- zgodność z harmonogramem (np. na poziomie całego przebiegu projektu, etapów, zakończenia projektu, wobec przewidzianych zmian i modyfikacji);
- zgodność z kosztami (np.: w poszczególnych pozycjach kosztów, ich grupach, nieprzekroczenia (oszczędności) w budżecie lub jego przekroczenie w określonych sytuacjach);
- zgodność z zakresem (np.: dotrzymanie ustalonego zakresu projektu wykonanie najważniejszych elementów lub zaplanowanych zadań, uzgodnione zmniejszenie lub zwiększenie zakresu projektu);
- zgodność z wymaganiami (np. osiągnięcie wyższej efektywności, akceptowalnej funkcjonalności lub użyteczności, utrzymanie jakości na określonym poziomie – uzgodnionym, normatywnym, porównywalnym);
- produkt lub usługa zakończona sukcesem rynkowym;
- właściwe zarządzanie ryzykiem (np.: minimalizacja zagrożeń, odpowiednia reakcja na sytuację nieprzewidzianą).

Dodatkowym problemem występującym zawsze w ocenach projektów jest skala preferencji w odniesieniu do poszczególnych kryteriów. W momencie przyjęcia założenia, że mamy więcej niż trzy równoważne (i zrównoważone kryteria, jak to było w teorii klasycznej), w przypadku równoważności kryteriów możemy mieć kłopoty z określeniem czy dany projekt odniósł rzeczywisty sukces. Możemy wtedy albo ustalić z interesariuszami projektu, które z kryteriów są dla nich (i w jakim stopniu) najważniejsze, albo zastosować do oceny jedną z metod relacyjnych (np. kompromisowych – typu AHP/ANP) (Saaty, 1999) lub funkcyjnych (np. wielokryterialnych).

Zarządzanie projektami w postaci klasycznej zostało określone jako „...zastosowanie wiedzy, umiejętności, narzędzi i technik w odniesieniu do działań realizowanych w projekcie, umożliwiające spełnienie wymagań danego projektu. To zastosowanie wiedzy wymaga skutecznego wykorzystania odpowiednich procesów...” (Project Management Institute, 2004, s. 39). Proces w tym przy-

padku oznacza zbiór powiązanych ze sobą działań podjętych w celu uzyskania określonych wyrobów lub usług. Charakteryzowany jest przez przypisane mu nakłady, zasoby, czas, narzędzia realizacji oraz spodziewane wyniki. Unikalność analizowanych i modyfikowanych procesów polegać ma na tzw. aktywach organizacyjnych zawierających zasady i kryteria dostosowywania procesów do konkretnych potrzeb projektów (zbioru procesów innowacyjnych i standardowych). Ogólnie dzieli się je na dwie grupy:

- procesy organizacyjne zapewniające skuteczny przebieg przedsięwzięcia (jak, w jakiej kolejności – następstwo działań, warianty działania, czy można prowadzić działania równolegle);
- procesy obiektowe – pozwalające na uściślenie warunków realizacji produktu lub usługi (co, w jaki sposób, za pomocą jakich technik).

Procesy te przeplatają się ze sobą na kolejnych etapach realizacji projektu, potrafią zachodzić równolegle i wchodzić w liczne interakcje. Standard PMI charakteryzuje procesy zarządzania projektami przez pryzmat wzajemnych oddziaływań między procesami i w ramach procesów, a także celów, którym służą (Project Management Institute, 2004, s. 41). Procesy te połączono w pięć grup, które są następnie realizowane w kolejnych fazach projektu. Projekty bowiem realizowane są według pewnej określonej kolejności wykonywanych przez poszczególnych realizatorów projektu czynności. Ten model sekwencyjnej oraz równoległej realizacji zadań projektu w wyznaczonym czasie i lokalizacji/ach, przy ograniczonych zasobach nazywany jest cyklem życia projektu, w nawiązaniu do procesów życia obiektu biologicznego od narodzin (inicjacji projektu), do śmierci (zakończenie projektu). Jest to de facto odwzorowanie sposobu realnego postępowania podczas budowy obiektu uwzględniające poszczególne fazy cyklu życia w postaci kanonicznej. Jeśli wystąpią istotne przyczyny zaburzające postać kanoniczną, jest on zmieniany. Zmiany mogą być spowodowane koniecznością dostosowania do potrzeb konkretnej organizacji, dostosowania się do wypracowanego przez wykonawcę i zleceniodawcę konsensusu bądź narzucenia zleceniodawcy rozwiązań wynikających z własnego rozeznania wymogów organizacji zleceniodawcy. Najbardziej skomplikowane, największe i przynoszące najwięcej kłopotów są projekty związane z realizacją systemów informatycznych.

Gwałtowny i wielokierunkowy rozwój technologii informacyjnych spowodował powrót do zainteresowania rozwojem metod projektowania systemów informatycznych wspomagających zarządzanie. Po latach zastoju spowodowanych praktykami wdrażania głównie powielarnych systemów zintegrowanych oraz konstruowaniem systemów biznesu elektronicznego poprzez kompilację gotowych fragmentów oprogramowania okazało się, że systematyzacja postępowania, uporządkowanie tej dziedziny i wypracowanie nowych sposobów analizy



i projektowania staje się niezbędne ze względu na jakość, użyteczność, funkcjonalność i efektywność tworzonego systemu informatycznego. Liczne i dotkliwe w skutkach aberracje w tym zakresie, skutkujące 60–80% współczynnikiem niepowodzenia realizacji projektów informatycznych skłoniły środowiska praktyków i środowiska akademickie do poszukiwania rozwiązań problemów projektowania systemów informatycznych. Zaczęto zwracać uwagę na czynniki, które wydawały się dotąd nieistotne wobec technologii, procedur, metodyk i innych czynników „twardych”. Stało to w sprzeczności z dotychczasową praktyką i należało „sproceduralizować” intuicyjne dotąd zachowania zespołów projektowych. Pojawiło się kilka szczególnych metod „miękkich” (zwinnych, nowoczesnych) o zróżnicowanym charakterze i przeznaczeniu.

Głównym celem niniejszego rozdziału jest analiza powyższych kierunków rozwoju metod projektowania systemów informatycznych, która jest prowadzona na tle dotychczasowego podejścia tradycyjnego.

### 1.3. Przegląd metodologii projektowania systemów informatycznych

W rozwoju projektowania systemów informatycznych wspomagających zarządzanie wyróżnia się generalnie cztery okresy, przebiegające na tle głębokich przemian sprzętu informatycznego i oprogramowania, które je umożliwiały:

- intuicyjnego projektowania funkcjonalnego,
- formalnego projektowania strukturalnego,
- formalno-intuicyjnego projektowania obiektowego,
- intuicyjno-formalnego projektowania społeczno-psychologicznego.

Intuicyjności i formalizmu w tych okresach nie należy oczywiście traktować dosłownie. Dotyczyły one głównie początkowych faz projektu i w każdym z wymienionych okresów dominacji rozwoju poszczególnego typu projektowania systemów informatycznych miały innych zasięg, odniesienie i charakter. Intuicyjność projektowania funkcjonalnego brała się początkowo z niedostatków teorii informatyki związanej z analizą i projektowaniem systemów. Rozpoznanie potrzeb użytkownika i ich przekształcanie w zapis, pozwalający na utworzenie na tej podstawie systemu, dokonywane było często przez (wówczas) nieprofesjonalistów wg metod zaczerpniętych z analizy funkcjonalnej przedsiębiorstwa lub statystyki (ankiety, wywiady, listy pytań itd.). Tworzone dokumenty – dotyczące całego systemu – grzeszyły często nadmiarowością informacji spowodowaną globalizacją problemu (system traktowany był jako całość bez wydzielenia zadań cząstkowych), wieloznaczności interpretacyjnych, wysokim stopniem niepotrzebnej szczegółowości i powtarzalności, związanym z tym bra-

kiem selekcji problemów i procesów istotnych od nieistotnych, długim czasem realizacji, w którym potrzeby użytkownika niejednokrotnie się zmieniały oraz długim czasem konwersji na wersję przydatną dla tworzenia oprogramowania (Chmielarz, 2000). Wysoka nieefektywność takich działań z jednej strony, z drugiej zaś – zmieniające się technologie informacyjne, zwłaszcza związane z procesami pozyskiwania, gromadzenia i wykorzystania danych wymusiły tworzenie nowych metod analizy i projektowania systemów informatycznych. Dwa najpoważniejsze problemy, przed jakimi stanęli twórcy tych metod, to:

- uproszczenie szczegółowych analiz funkcjonalnych przeprowadzanych niejako przed stworzeniem systemu informatycznego (zwłaszcza złożonego), poprzez rozłożenie na prostsze elementy składowe oraz ograniczenie go do informacji naprawdę niezbędnych dla budowy systemu (w ten sposób istotna staje się struktura problemu, a nie stopień szczegółowości opisu);
- uproszczenie metod komunikacji zwłaszcza z użytkownikiem końcowym systemu poprzez stworzenie języków (odwzorowań) graficznych analizowanych problemów i programów (klasy CASE – Computer Aided System Engineering) wspomagających poszczególne fazy lub całość projektu.

Stopniowe rozwiązywanie tych problemów doprowadziło do powstania grupy metod strukturalnych (główni przedstawiciele: T. DeMarco, V. Weinberg, E. Yourdon), stale doskonalonych i różniących się pomiędzy sobą w ramach różnych „szkół” analizy, projektowania i programowania strukturalnego. Istota podejścia strukturalnego sprowadza się do dwóch cech charakterystycznych – zstępującej metody dekompozycyjnej (*top-down* – określenie zakresu i kryteriów szczegółowych podziału) i wynikającej z wcześniej przejętego dorobku analizy systemowej oraz zdefiniowania dla każdego elementu dekompozycji: odwzorowania danych (statyka); dynamiki, czyli przejścia z jednego stanu w drugi oraz zakresu funkcjonalnego (transformacje danych wejściowych w dane wyjściowe). Odwzorowania danych w postaci statycznej oraz ich dynamika przedstawiane są na ogół w postaci dokładnie sprecyzowanego języka, przy użyciu ściśle określonych notacji (zapewniających jednoznaczność tworzonych modeli). Dane te są połączone zależnościami hierarchicznymi i relacyjnymi. Funkcjonalność zaś wyznaczana jest za pomocą algorytmów precyzujących zachowanie się systemu zgodnie z wymogami użytkownika. Podejście strukturalne w pierwszych fazach tworzenia systemu informatycznego składa się z trzech etapów:

- analizy – w którym następuje definiowanie modelu logicznego systemu i formalizacja wymagań użytkownika; szczegółowa specyfikacja zadań i funkcji realizowanych przez system, przedstawiana przeważnie obok warstwy opisowej poprzez graficzne języki prezentacji, np.: diagram przepływu danych (*DataFlow Diagram*), specyfikacje procesów (*Process Specification*), diagramy relacyjne danych (*Entity Relationship Diagram*), słowniki danych (*Data Dic-*

tionary), diagramy przepływu sterowania (*SteeringFlow Diagrams*), diagramy HIPO (*Hierarchy Input-Process-Output*) oraz diagramy przejść stanowych (*State Transition Diagrams*);

- projektowania – powstanie fizycznego modelu systemu (modelu procesora, modelu zadania i modelu implementacyjnego programu);
- implementacji – transformacja modelu fizycznego na oprogramowanie, testowanie go i wdrożenie.

Wywodząca się z analizy systemowej grupa metod strukturalnych była od momentu swojego powstania zrozumiała i akceptowalna przez analityków, projektantów i wdrożeniowców. Wraz z postępującą formalizacją została ona również przyjęta do użytkowania przez programistów. Metody strukturalne stały się szczególnie popularne w latach siedemdziesiątych przy tworzeniu programów obsługujących bazy danych. Metodyki szczegółowe występujące w tej grupie są do dziś doskonalone i wykorzystywane w praktyce.

Rozwój języków oprogramowania (począwszy od języków służących symulacji (*Simula 67*), w koncepcji dopracowanej jako *Smalltalk*) w kierunku programowania obiektowego wymusił natomiast powstanie i rozwój metod projektowania dostosowanych do tej koncepcji. Idea programowania obiektowego polegała na definiowaniu programów za pomocą obiektów będących funkcjonalnymi ich cech charakterystycznych, relacji pomiędzy nimi oraz sposobów zachowań (funkcjonalności). I o ile w programowaniu strukturalnym podstawą komunikacji w systemie w celu realizacji zadań były relacje pomiędzy tablicami danych, służących do realizacji poszczególnych procedur, o tyle w programowaniu obiektowym były to już relacje pomiędzy grupami procedur realizowanych przez wyodrębnione obiekty. Do projektowania takich systemów niezbędne były inne metodyki projektowania zwane obiektowymi.

Ich prekursorem była tzw. metoda operacyjna Jacksona, której meritum polega na transformacji operacyjnej danych wejściowych na dane wyjściowe (modelowanie struktur systemu oprogramowania na podstawie struktur danych), rozpoczynającej się od wyodrębnienia obiektów i funkcji przez nich wykonywanych, łączących się następnie w procesy, dekomponowane w kolejnym kroku na poziom modelu fizycznego.

Jest to metoda pośrednia pomiędzy metodami strukturalnymi a obiektowymi, w klasyfikacjach metod raz przypisywana do jednego, raz do drugiego podejścia. Składa się ona z trzech zasadniczych części:

- analizy – polegającej na tworzeniu modelu struktur danych, przeznaczonych do transmisji danych wejściowych do końcowego stanu wyjściowego; podstawowym narzędziem modelowania są tu: diagramy struktur danych (*Data Structure Charts*) – pozwalające na modelowanie hierarchiczne drzew danych (wyodrębnienie obiektów);

- specyfikacji – budowa schematu przepływów struktur danych przez programy, zapewniające powiązania pomiędzy danymi (wyodrębnienie funkcji wykonywanych przez każdy obiekt); podstawowe narzędzie wykorzystywane na tym etapie to sieci specyfikacji systemu (*System Specification Network*);
- implementacji – definiowanie modelu struktury programów odzwierciedlających zaprojektowane wcześniej struktury danych; narzędziem są tutaj diagramy struktury programów (*Program Structure Charts*), wykorzystujące dokładnie te same notacje, jakie występowały podczas modelowania struktur danych.

Projektowanie obiektowe przejęło z projektowania operacyjnego wyodrębnienie obiektów i klas obiektów charakteryzowanych specyficznymi atrybutami, poszło natomiast dalej niż metoda Jacksona, ponieważ w definicji obiektu ujęło zarówno cechy i ich powiązania, jak i wykonywane przez obiekt procedury obsługujące procesy organizacyjne. Sprawia to, że obiekty nabierają dodatkowego wymiaru, pozwalającego na efektywniejsze modelowanie skomplikowanych procesów gospodarczych. Generalnie ujmując, są to metody pozwalające, w odróżnieniu od dwu poprzednich, na jednoczesne modelowanie danych i procesów dziedziny przedmiotowej.

Wyształciły się wśród nich różne nurty metod opartych na analizie struktur danych (np. Booch, 1994; Coleman, 1995), zdarzeń (np. Martin, Odell, 1994) bądź scenariuszy zdarzeń (Wirfs-Brock, Wilkerson, Wiener, 1990). Pomimo istotnych różnic występujących pomiędzy wymienionymi metodykami, w większości z nich da się wyraźnie wyróżnić takie etapy projektu, jak:

- specyfikacja systemu i analizy – tworzenie modelu obiektów oraz wariantów ich użycia (statyczna struktura systemu), np. dzięki użyciu diagramów obiektów (*Object Diagrams*) oraz modelu interfejsów (dynamiczna struktura systemu modelowana za pomocą scenariuszy (*Scenario Diagrams*), modelu operacji użytkownika (*Operation Model*), słownika danych (*Data Dictionary*), modelu powiązań systemu obiektów (*System Interface Graphs*);
- realizacja poszczególnych operacji wyróżnionych na etapie analizy; definiowanie procesu współdziałania obiektów, np. w postaci opisu: grafów interakcji i widoczności obiektów, grafów, dziedziczenia, opisu klas, procesów, modułów;
- implementacja – przekształcanie modeli fazy projektu na oprogramowanie; wykorzystuje się w niej stworzone wcześniej diagramy stanów i ich zmiany, a następnie generuje bądź tworzy kod.

W latach osiemdziesiątych, zwłaszcza w drugiej ich połowie, wydawało się, że projektowanie obiektowe stanie się dominującym ze względu na masowe użycie obiektowego języka oprogramowania C++ będącego rozszerzeniem

języka C oraz dodawanie cech obiektowych do najpopularniejszych języków programowania (m.in. języka BASIC, Pascala, Ady, Lisp itp.). Szczególnie, że nastąpił znaczący wzrost popularności graficznych interfejsów użytkownika, w którym programowanie obiektowe bardzo dobrze się sprawdzało. Dodatkowym argumentem przemawiającym za tym było powstanie dziesiątek szkół projektowania i programowania obiektowego. Jednakże dodanie funkcji obiektowych do języków programowania zrodziło szereg problemów z integralnością modyfikowanych systemów, których poprzednie wersje nie były kompatybilne z następnymi. Również wielu projektantów uważało, że projektowanie strukturalne posiada wiele cech, do których zdążyli się już przyzwyczaić, a których nie posiadały programy obiektowe. Szczególnie nie odpowiadała im intuicyjność wydzielenia klas obiektów prowadząca niekiedy do daleko idącej powtarzalności oprogramowania. Dlatego cechy obiektowości rozpowszechniały się raczej w połączeniu z językami niższego rzędu (*assemblery*) czy sieciowymi (Java) niż w postaci czystej. Trwająca przez lata dyskusja, które z podejść (strukturalne czy obiektowe) jest lepsze, doprowadzona została do swoistego konsensusu – w informatycznych systemach wspomagających projektowanie często serwuje się obie grupy metod do wyboru, a często określone grupy metod stosuje się zwyczajowo lub z premedytacją do określonego rodzaju systemu (inne do małych i średnich, inne do dużych zintegrowanych, inne do usługowych, inne do systemów e-biznesu).

Niejako na marginesie metodyk sformalizowanych rozwijały się metody miękkie (*soft*), zwane niekiedy społecznymi. Są to metody, w których analiza i projektowanie systemu oparte jest na teorii rozwiązywania konfliktów – socjologiczno-psychologicznych negocjacjach zmian z grupami pracowniczymi oraz grupami kreującymi system, w aspekcie organizacyjnym. Z wynegocjowanych zmian strukturalnych wynika docelowa, pożądana postać systemu, jego celów i funkcji, która następnie jest informatyzowana. Metodyka ta akcentuje w procesie projektowania systemu informatycznego uwzględnienie zagadnień satysfakcji użytkownika i twórcy systemu z wykonywanej pracy, partycypacji wszystkich osób i jednostek organizacyjnych w tworzeniu systemu, szerokiego współuczestnictwa wszystkich stron i interesów w procedurze projektowej (przedstawiciele: W. Mumford, P. Checkland, B. Wilson). W metodyce tej, w trakcie pierwszych faz tworzenia systemu, oprócz wykorzystania procedur zapożyczonych z nauk organizacji i zarządzania, stosuje się różne metody socjologiczne i psychologiczne. W podejściu tym wyróżnia się takie etapy, jak:

- analiza i założenia systemu – przechodząca od definicji sytuacji problemowych oraz odwzorowań modelowych struktury i dynamiki między stanowiskami i komórkami organizacyjnymi, powiązań formalnych i zależności nieformalnych, do charakterystyki struktury i dynamiki sytuacji problemowej z opisem uczestników i ich ról w procesie budowy systemu i stworzenia sys-

- temu modeli konceptualnych (sekwencji wzajemnie powiązanych procesów realizowanych w celowej działalności);
- weryfikacja – odniesienie systemu modeli konceptualnych do rzeczywistości, określenie i przeprowadzenie pożądanych zmian wykonalnych ze względu na kulturę organizacji,
  - formalizacja i implementacja – technicystyczne odwzorowanie na model fizyczny ze wskazaniem na interakcyjne techniki prototypowania, pakiety wspomagane analizą ryzyka, systemy probabilistyczne, analizy wielokryterialne, systemy sztucznej inteligencji.

## 1.4. Kierunki rozwoju tradycyjnych metod projektowania systemów informatycznych

Przemiany w tworzeniu i wdrażaniu systemów informatycznych wspomagających zarządzanie, które nastąpiły na przełomie wieków zmieniły nieco optykę postrzegania dotychczasowych metod. Większość metodyk przedstawionych w poprzedniej części opartych na dotychczasowych, a już konwencjonalnych modelach cyklu życia systemu informatycznego zaliczono do tzw. metod tradycyjnych.

Cykl życia systemu informatycznego jest w nich traktowany jako sekwencja etapów czynności wzorowanych na życiu organizmów, którego ostatecznym celem jest budowa, wdrożenie, modyfikacje i wycofanie (lub zastąpienie) systemu informatycznego, począwszy od pierwszego pomysłu na ten temat (narodziny), po jego likwidację (śmierć). Idea cyklu życia (projektu, systemu, procesów inżynierii) jest w tych wszystkich koncepcjach tożsama, różnice występują w ilości i nazewnictwie etapów (faz), ich kolejności logicznej lub zrównolegleniu i następstwie oraz relacjach z otoczeniem, szczególnie z użytkownikiem finalnym. Różnice te nasiliły się w ostatnim dziesięcioleciu wraz z momentem rozpoczęcia szerokiego stosowania tzw. metodyk lekkich (zwinnych – agile).

Według tego podejścia w tworzeniu modeli cyklu życia projektu wyróżnia się metodyki:

- ciężkie (klasyczne, tradycyjne), takie jak: kaskadowa (liniowa), przyrostowa, ewolucyjna, baz danych, prototypowa, spiralna;
- lekkie (nowoczesne, zwinne – agile), takie, jak: XP (eXtreme Programming), Scrum, Feature Driven Development (FDD), Dynamic System Development Method (DSDM) czy Adaptive Software Development (ASD) (Chmielarz, 2010, s. 359–402).

Kanon postaci cyklu życia projektowania systemów informatycznych wywodzi się z historycznego rozwoju metod zarządzania projektem informatycznym

(szkół strukturalnych, operacyjnych, obiektowych i społecznych) z jednej strony i rozwoju technologii systemów – z drugiej. Były to procesy długotrwałe, dynamiczne i radykalne, dlatego w teorii lepiej i bardziej szczegółowo zaprezentowane i opisane są metody tradycyjne projektowania. Polegają one na realizacji sekwencji liniowej albo równoległej etapów, które dadzą się sprowadzić do następujących:

- inicjacji – polegającej na identyfikacji celu, problemów, stanu obecnego i perspektyw rozwoju systemu informacyjnego lub jego zmian;
- analizy informacyjnej systemu – na którą składa się identyfikacja otoczenia, łączności z otoczeniem, podsystemów, składowych systemu oraz obecnych i przyszłych warunków jego działania;
- projektowania systemu – koncentrującego się na uszczegółowieniu założeń analitycznych, stworzeniu modelu logicznego i fizycznego przyszłego systemu lub ewentualnych zmian w funkcjonowaniu obecnego systemu;
- oprogramowania systemu (implementacji) – polegającego na oprogramowaniu modelu fizycznego, konstrukcji programu lub systemu informatycznego;
- testowania – sprawdzenia pod względem semantycznym, technicznym i merytorycznym prawidłowości funkcjonowania systemu;
- wdrożenia – uruchomienia i zainstalowania systemu u użytkownika końcowego, załadowania systemu parametrami niezbędnymi do jego funkcjonowania, podłączenia do aktualnych źródeł danych, ostatecznego szkolenia użytkownika końcowego;
- eksploatacji – sprowadzającej się do uruchomienia użytkowego, monitoringu, oceny i modyfikacji systemu;
- wycofania systemu – po badaniu opłacalności, opracowanie procedury przejścia na nowy system informatyczny, zachowujący maksymalnie możliwą ilość pozytywnych cech starego systemu.

Najbardziej znanym, najczęściej używanym i najbardziej charakterystycznym modelem postępowania w metodach tradycyjnych – realizujących powyższe etapy – był model liniowy (kaskadowy). Przyjmowano w nim następujące założenia:

- na początku każdego projektu istnieje stabilny zestaw potrzeb i wymagań informacyjnych użytkownika i celów, do których on dąży;
- potrzeby nie zmieniają się w trakcie życia systemu;
- proces budowy systemu odbywa się stopniowo, dopiero po skończeniu jednej fazy zaczyna się faza następną;
- każdy kolejny etap oznacza uszczegółowienie i przybliżenie do rzeczywistości;
- powoduje to powrót do poprzednich etapów w momencie, gdy zostanie wykryty błąd i powstanie konieczność jego korekty.

W klasycznym kaskadowym podejściu wyróżnia się pięć podstawowych faz (zob. rys. 3), wspólnych ideowo także dla innych metodyk tradycyjnych (Chmielarz i Klincewicz, 2010, s. 238–252):

- inicjowanie i definiowanie projektów,
- określenie struktury projektów,
- planowanie przebiegu i zasobów projektu,
- sterowanie przebiegiem projektu,
- zamknięcie projektu.

Faza inicjowania i definiowania projektów składa się z trzech zasadniczych etapów: inicjowania projektu, definiowania projektu i organizowania zespołu projektowego. Inicjowanie projektu zawiera analizę potrzeb i wymagań zleceniodawcy wraz z artikulacją inicjatyw podjęcia projektu i jego zgłoszenia gremiom kierowniczym (chyba że to one są autorami inicjatywy); następnie oceny inicjatywy i jej przyjęcia do dalszej realizacji, odrzucenia bądź modyfikacji. Definiowanie projektu jest etapem, gdzie następuje doprecyzowanie inicjatywy w sensie określenia szczegółowego celu/ów, jakie będą zrealizowane dzięki projektowi, określenia ryzyka projektu (poprzez analizę i ocenę), wstępny rachunek (czasem wariantowy) kosztów, nakładów i efektów związanych z przyjęciem i realizacją projektu oraz – iteracyjnie – przyjęcia, modyfikacji lub odrzucenia zdefiniowanego projektu. Ostatnim – ale równie istotnym etapem – jest powołanie zespołu projektowego. Po pierwsze określa się sposób instytucjonalno-organizacyjnej formy wykonania projektu. Po podjęciu decyzji powołuje się kierownika projektu oraz zespół projektowy. Kierownik projektu jest osobą o największym wpływie na powstanie, realizację oraz efekt końcowy projektu. Uczestniczy on bezpośrednio w procesie zarządzania projektem poprzez koordynację, motywację, eliminację problemów i minimalizację ryzyka. Ma też istotny wpływ na pozyskanie optymalnego składu oraz organizowanie pracy (wraz z harmonogramem) zespołu projektowego.

Kolejną fazą rozwojową projektu jest określenie jego struktury. Polega ono na dalszym doprecyzowaniu celów projektu, które w fazie inicjowania mogły zostać pominięte lub niedocenione. Wymaga to na ogół zebrania dodatkowych informacji dotyczących wymogów użytkownika (ze strony zleceniodawcy), a następnie ustrukturalizowania dotychczasowych założeń. Ze względu na prawidłowość realizacji projektu – zwłaszcza w przypadku projektów złożonych i nowatorskich – ustala się kryteria podziału projektu na części składowe i określa strukturę pionową i poziomą. W trakcie określania struktury projektu dokonywana jest instytucjonalizacja formy zarządzania projektem. Oznacza ona zakres i sposób powiązania organizacji projektowej ze strukturą przedsiębiorstwa, w którym przeprowadza ona realizację projektu. Polega zaś na dokonaniu wyboru spośród różnych struktur zarządzania (np. struktury liniowej,



liniowo-sztabowej, macierzowej, projektowej, wydzielonej-projektowej, realizacji zewnętrznej) takiej, która zapewni najbardziej skuteczną realizację projektu. Instytucjonalna forma zarządzania projektem jest wymuszana problemami, które się pojawiają w związku z jego realizacją, dlatego wstępnie nie można określić najlepszej formy instytucjonalnej – czasem jest ona modyfikowana w trakcie realizacji – dodawane są do niej nowe elementy, typu – powołanie dodatkowego zespołu ekspertów niezależnych w postaci komórki sztabowej ustalonej wcześniej struktury. Powstają dylematy dotyczące zakresu funkcjonalności i kompetencji powoływanej struktury – czy organizacja ma być skierowana głównie na realizację zadań bieżących czy na realizację projektu. Niewątpliwie przy różnych formach instytucjonalnych różny jest również problem koordynacji i związanej z nią niestabilności. Powoływana struktura zależy również od wielkości projektu, stopnia jego komplikacji oraz zakresu innowacyjności (i kosztów z tym związanych, o których się tu jeszcze nie decyduje, ale o których należy pamiętać, ponieważ zostały wstępnie określone w pierwszej fazie projektu).

Trzecią fazą jest planowanie przebiegu i zasobów projektu. Plan powinien zawierać sprecyzowany cel główny i wynikające z niego cele cząstkowe oraz związane z nimi zadania realizacyjne w podziale strukturalnym, ustalonym w fazie poprzedniej, przypisanie zadań wykonawcom i podwykonawcom, jak również określenie czasu realizacji poszczególnych zadań projektu oraz poszczególnych czynności wykonywanych w ramach zadań (jeśli jest to wymagane ze względu na całość projektu). Dla zapewnienia realizacji zadań i czynności w czasie sporządzany jest harmonogram prac, w którym zaznacza się również krytyczne momenty projektu (kamienie milowe) oraz punkty weryfikacyjne (kontroli prac wraz z kryteriami), odbioru prac (zasady odbioru), przekazywania użytkownikowi. Ogranicza to jednocześnie konflikty wewnątrz zespołu projektowego oraz zespołu projektowego z organizacją, dla której wykonywany jest projekt i zmniejsza ryzyko jego niewykonania. W szczególności istotne dla tej fazy jest planowanie zasobów projektu. Działanie to polega na oszacowaniu zasobów fizycznych ludzi, materiałów, surowców, narzędzi niezbędnych do realizacji projektu oraz proporcji (czasem w postaci normatywnej) ich wykorzystania do realizacji poszczególnych zadań (czasem w postaci alternatywnych zamienników). Następnie wykonywane jest przybliżenie kosztów – zasobów fizycznych, ich zestawienie w postaci wartościowej oraz dokumentacja tych oszacowań. W wyniku określenia kosztów powstaje plan zarządzania kosztami, który powinien również uwzględniać reguły postępowania w przypadku odchyień realizacyjnych od początkowych założeń projektowych w tym zakresie. Dla realizacji planu dokonywane jest tzw. budżetowanie – czyli przypisanie elementom planu przewidywanych kosztów ich realizacji. Rozkład przewidywanych kosztów w czasie umożliwia ich późniejsze porównanie z kosztami realizowanymi na bieżąco i ewentualne działania korygujące w przypadku ewentualnych odchyień. Po zatwierdzeniu planowanego

przebiegu i planu zasobów projektu następuje podjęcie decyzji, co do dalszej realizacji projektu.

Szacunki do tej fazy dokonywane są na podstawie szeregu technik dodatkowych typu: techniki analitycznej, analogii, ocen parametrycznych oraz ocen korelacyjnych, technik oceny i porównań punktowych oraz ocen funkcjonalności (np. systemy oprogramowania).

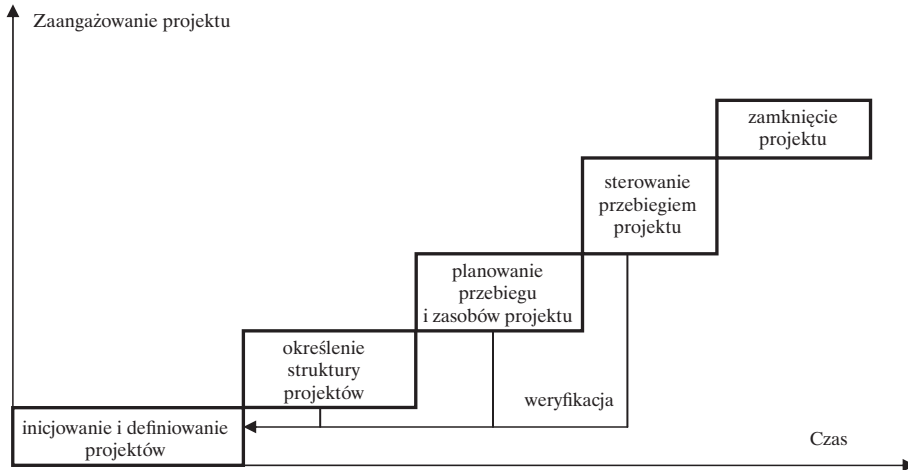
Kolejną fazą cyklu życia projektu jest sterowanie jego przebiegiem. W zasadzie składa się z trzech przenikających się etapów: zorganizowania wykonawstwa projektu, koordynacji przygotowania i realizacji oraz monitorowania i kontroli. Podstawą działania jest przygotowany uprzednio plan przebiegu projektu. Dla jego wypełnienia dokonywane jest pozyskanie środków finansowych na jego realizację, przygotowanie podziału zadań projektowych, „zabezpieczenie” dostawców, negocjacje i ustalenia z podwykonawcami, kontraktowanie potrzebnych dostaw i usług oraz opracowanie systemów motywacyjnych i osiągnięcia odpowiedniej jakości wykonawstwa projektu. W trakcie realizacji następują działania koordynacyjne – terminów wykonawstwa, zużycia zasobów, kosztów, ryzyka, jakości oraz współpracy pomiędzy członkami zespołu projektowego. Niezbędnym elementem tej fazy jest kontrola: terminów wykonania zadań, zużycia zasobów, wysokości ponoszonych kosztów, terminowości dostaw, minimalizacji ryzyka wykonania, zapewnienia odpowiedniej jakości oraz pracy zespołu i jego współpracy z otoczeniem organizacyjnym. Za normy odniesienia mogą tu służyć uzyskane w poszczególnych etapach wyniki ilościowe i jakościowe, czasowe (wywiązywanie się z ustalonych terminów), zużycia zasobów (w tym: finansowych) oraz ponoszenia ryzyka.

Kończącą fazą procesu projektowego jest zamknięcie projektu. Działania w tej fazie obejmują przygotowanie raportu końcowego z realizacji projektu, zebranie opinii i recenzji na temat jego przebiegu (jeśli są potrzebne), przekazanie (odbior) projektu zleceniodawcy, ostateczne rozliczenie wyników projektu (co się udało, co nie, co zostało zrealizowane inaczej niż planowano) i podjęcie ostatecznych ustaleń, co do działań poprojektowych (konserwacja, tryb modyfikacji itp.). Jako kolejny etap następuje podjęcie decyzji o zakończeniu projektu i rozwiązanie zespołu projektowego (albo przekształcenie go w zespół nadzorujący prawidłowe funkcjonowanie wyników prac wykonanych w trakcie projektu).

Procesy w poszczególnych fazach projektu nie muszą następować kolejno, ale mogą być realizowane równolegle. W każdej fazie poszczególne procesy są powiązane pomiędzy sobą przez tzw. wejścia i wyjścia. Znając metody, narzędzia i techniki transformacji wartości wejściowych w wyjściowe, możemy opisać każdy proces w poszczególnych fazach projektu.

W praktyce występują też projekty, które nie przechodzą przez wszystkie wymienione fazy cyklu życia. W niektórych z nich zapada na pewnym etapie

decyzja o wycofaniu się z projektu, w innych niektóre fazy czy etapy powtarzane są wielokrotnie, co może zależeć od przyjętej metodyki realizacji projektu.



**Rysunek 3.** Podstawowe fazy cyklu życia projektu w modelu kaskadowym

Źródło: opracowanie własne.

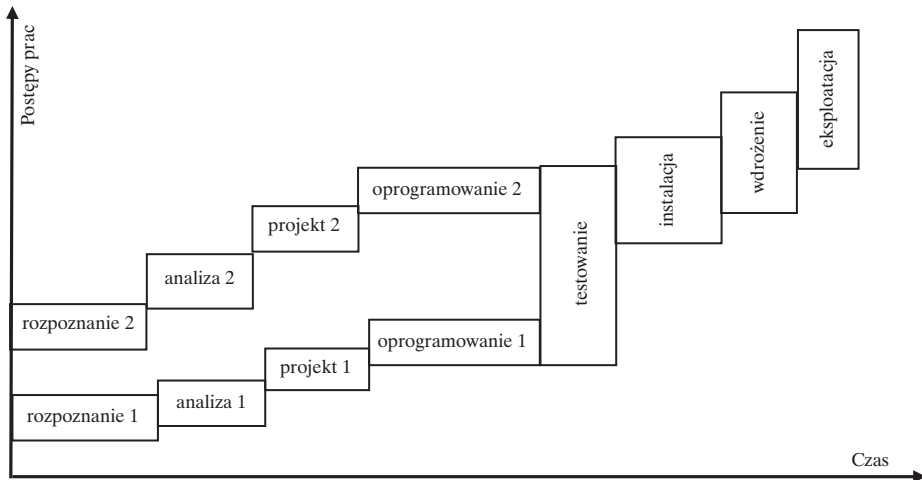
Niestety w rzeczywistości już założenie pierwsze na ogół się nie sprawdza (użytkownicy nie mają sprecyzowanych na wstępie potrzeb), dlatego może szybko nastąpić rozminięcie się z potrzebami użytkownika. Ponadto jest to bardzo droga kombinacja, ponieważ w przypadku realizacji przez firmę informatyczną tylko jednego projektu w danym czasie, oznacza konieczność utrzymywania ekipy realizacyjnej częściowo bezrobotnej w trakcie realizacji następnych etapów. Ponadto w rzeczywistości często występuje nakładanie się poszczególnych etapów na siebie. W tej sytuacji istnieją liczne modyfikacje tego podejścia.

Założenia modelu ewolucyjnego były następujące:

- cały system jest dzielony na moduły,
- każdy z nich odbywa przejście przez kolejne fazy cyklu budowy systemu,
- na końcu działań projektowych przystępuje się do specjalnego etapu polegającego na integracji całego systemu i przeprowadzeniu testów,
- w systemie podzielonym na części, których realizacja jest przesunięta w czasie łatwiej nadążać za zmieniającym się celem działania.

Ponieważ każdy moduł stanowi początkowo organicznie odrębną część, należy zwrócić uwagę na niebezpieczeństwo związane z koniecznością integracji

modułów w całość. Bardzo często staje się to główną przyczyną niepowodzeń realizacji projektu.



**Rysunek 4.** Schemat ideowy ewolucyjnego modelu życia systemu

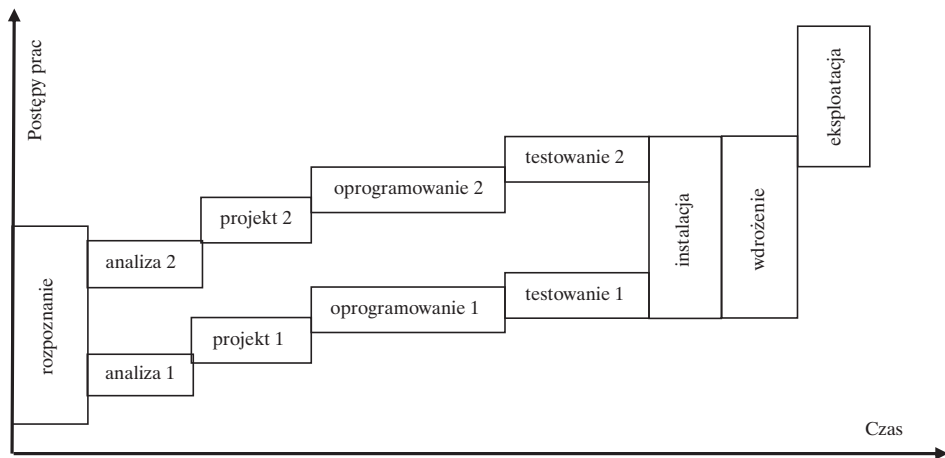
Źródło: opracowanie własne.

Jest to rozwiązanie tańsze w realizacji od poprzedniego, ponieważ może – po przesunięciu realizacji poszczególnych modułów w czasie – być obsługiwane przez mniejszą ilość personelu. Kiedy analitycy kończą pracę nad jednym modułem, przechodzą na następny, a ich miejsce zajmują programiści. Model ewolucyjny niestety nie zawsze spełniał pokładane w nim nadzieje. Poszczególne podsystemy, dla dużych projektów budowane przez różne zespoły, nawet pomimo tworzenia specjalnych procedur integracyjnych, nie zawsze w pełni dawały się zintegrować ze względu np. na różną wizualizację mechanizmów funkcjonowania. Brakowało w nich wyraźnie spójnych założeń dla całego systemu informatycznego. Dlatego właśnie model kaskadowy ewoluował dalej – do stworzenia modelu przyrostowego. Model przyrostowy charakteryzował się następującymi założeniami.

- przeprowadzane jest rozpoznanie i analiza dla całości systemu, dzięki której powstaje całościowa koncepcja wstępna systemu poparta ogólną analizą całego systemu;
- następnie – system podzielony na moduły realizacyjne jest projektowany, programowany i testowany kolejno dla każdego z nich przez dziedzinowe zespoły robocze wykonujące projekty techniczne dla każdego modułu i je testujące;

- spójność systemu zapewniają pierwotne założenia systemu oraz wspólne końcowe etapy instalacji i wdrożenia, w których przeprowadzana jest też pełna integracja systemu.

Jest to procedura, podobnie jak poprzednia, najlepiej nadająca się do zastosowania przy ograniczonych środkach, którymi dysponuje zespół realizujący system. Każdy z podsystemów może być realizowany oddzielnie, cały zespół pracuje jedynie na początku w trakcie tworzenia całościowej koncepcji rozwiązania systemu oraz na końcu, w trakcie integracji poszczególnych podsystemów w jedną organiczną całość. Procedura przyrostowa jest lepsza od ewolucyjnej, ponieważ mechanizmy integracyjne działają zarówno na początku, jak i na końcu procesu tworzenia systemu. Na początku cyklu, oprócz rozpoznania i analizy, tworzone są także wspólne normatywy i zasady wykonania projektu każdego modułu. Ułatwia to późniejszą integrację na etapach końcowych. W cyklu ewolucyjnym oraz przyrostowym można też w ograniczony sposób reagować na zmiany wymagań użytkownika.



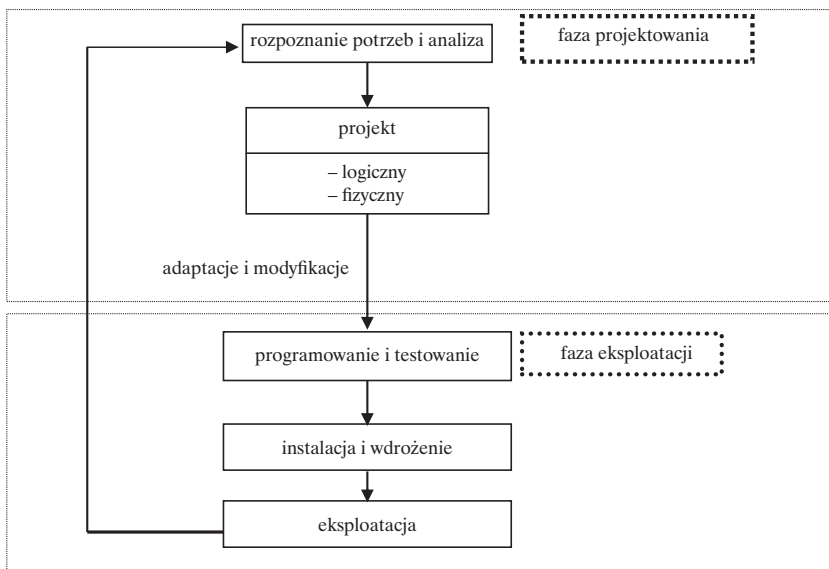
**Rysunek 5.** Schemat ideowy przyrostowego cyklu życia systemu

Źródło: opracowanie własne.

Odrębną systematykę zakłada model tworzenia struktury baz danych (Fry'ego) (Fuglewicz, Stąpor i Trojnar, 1995). W cyklu życia opartym na tworzeniu struktury bazy danych wyróżnia się na ogół następujące fazy:

- rozpoznania i analizy – gdzie następuje rozpoznanie i zebranie potrzeb informacyjnych użytkowników;

- projektu technicznego, składającego się z dwóch etapów dotyczących tworzenia:
  - projektu logicznego – czyli opisu modelu danych i przyszłych procesów w systemie;
  - projektu fizycznego – czyli projektu struktury bazy danych (zbiorów i relacji pomiędzy nimi), wzorców dokumentów, technologii przetwarzania, specyfikacji wewnętrznych itp.;
- oprogramowania i testowania – polegającego na stworzeniu bazy danych i oprogramowanie zastosowań (aplikacji) opartych na danych zawartych w bazie danych oraz testowaniu oprogramowania;
- wdrożenia – zainstalowanie oprogramowania w określonej platformie i konfiguracji sprzętowej i wprowadzenia parametrów określonych przez użytkownika w celu kustomizacji i przygotowania systemu do eksploatacji;
- eksploatacji użytkowej i kontroli – polegającej na roboczym użytkowaniu systemu oraz zapewnieniu poprawności z ustalonymi normatywami i wymogami użytkownika;
- potencjalnych modyfikacji i adaptacji – czyli udoskonalenia funkcjonowania w wyniku pojawienia się nowych potrzeb – w przypadku powrotu do etapów początkowych.



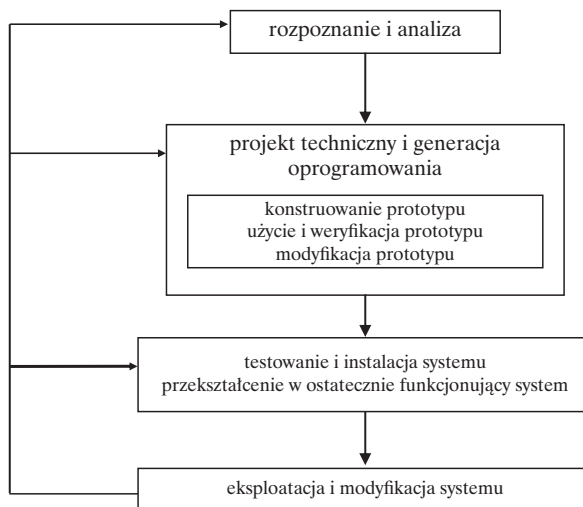
**Rysunek 6.** Schemat ideowy modelu tworzenia struktury baz danych

Źródło: opracowanie własne.

Zasadnicza idea tego cyklu życia polega na stworzeniu na początku projektu struktury bazy danych systemu i oprogramowania bazy danych – co umożliwi rejestrację podstawowych faktów z życia organizacji, a następnie obudowanie bazy danych oprogramowaniem aplikacyjnym, które wykorzystuje dla różnych aplikacji te same dane zawarte w bazie danych do obsługi, np. sprzedaży, dostaw, kontaktów z kontrahentami. W ten sposób powstawało i jest wykorzystywanych do dziś wiele złożonych systemów opartych na bazie danych. Jest to jednak metoda, gdzie rezultaty działania projektanta i programisty użytkownik końcowy ogląda dopiero po uruchomieniu bazy danych oraz pierwszych aplikacji. Wtedy też może sprawdzić ich zgodność z ustalonymi z zespołem projektowym założeniami. Znacznie szybciej uwidaczniane jest to podczas zastosowania modelu prototypowego.

Istotą modelu prototypowego jest oprogramowanie – na początku działań projektowych – schematu działania systemu, najlepiej razem z przyszłym użytkownikiem. Cel takiego postępowania jest następujący:

- redukcja czasu oczekiwania na rezultaty programowe i wykazanie się wobec użytkownika końcowego uchwytymi (dla niego) efektami realizacyjnymi;
- szybkie „sprzężenie” użytkownika z zespołem projektowym;
- ograniczenie liczby błędów oraz iteracyjnych (potencjalnych) rozmów z użytkownikiem, np. na temat wizualizacji systemu;
- większe zaangażowanie użytkownika w analizę i projekt.



**Rysunek 7.** Schemat ideowy modelu z tworzeniem prototypu

Źródło: opracowanie własne.

Postępowanie w modelu prototypowym składa się z następujących etapów:

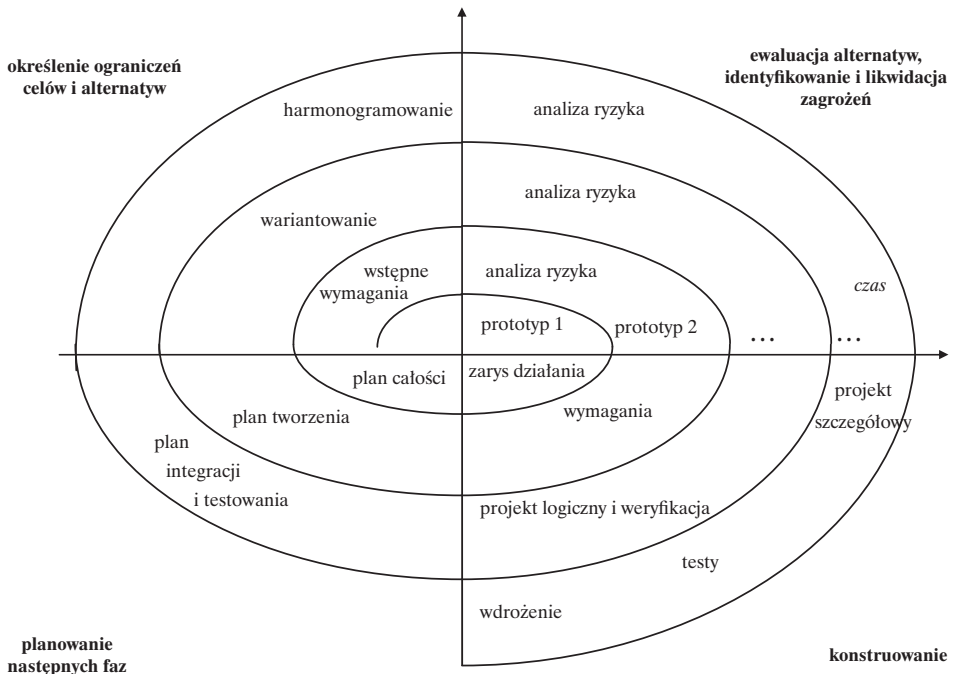
- rozpoznania i analizy potrzeb użytkownika;
- projektu technicznego wraz z generacją oprogramowania – w trakcie tego etapu powstaje prototyp, który następnie jest sukcesywnie poprawiany w interakcji z użytkownikiem;
- po testach, kolejnych wersji prototypu i jego udoskonaleniach oraz instalacji system jest przekształcany w wersję ostateczną, która przekazywana jest do eksploatacji i – w miarę potrzeb – modyfikowana za pomocą narzędzia, które posłużyło do jej stworzenia.

Sprawna realizacja systemu prototypowego wymaga zastosowania dobrego oprogramowania klasy CASE (*Computer Aided System Engineering*), co niestety powoduje, że jego bezpośrednie i pełne zastosowanie bywa ograniczone poprzez własności tego narzędzia i dodatkowo zwiększa koszty projektu. Dlatego, pomimo niewątpliwych zalet tego podejścia, wielu projektantów i programistów woli stosować jedno z podejść klasycznych. Najbardziej rozwiniętą formą modelu tradycyjnego jest natomiast model spiralny, którego fazy są następujące:

- ustalenie celów – dotyczy określenia konkretnych celów oraz uzasadnienia koncepcji i ustalenia planów ich realizacji; budowa modelu rozpoczyna się od inicjacji i definiowania projektu (ustalenia wstępnych wymagań), a następnie analizy wstępnej ryzyka realizacji projektu, na tej podstawie zaś buduje się pierwszy prototyp i tworzy konceptualny plan (harmonogram) całości projektu;
- rozpoznanie i redukcja zagrożeń – po kolejnej fazie analizy ryzyka pierwszego prototypu (identyfikacja najważniejszych zagrożeń, źródeł i sposobów zapobiegania zagrożeniom) oraz zbadaniu ewentualnych alternatyw budowany jest następny prototyp projektu i tworzy się wymagania dotyczące ograniczeń i zasobów projektu;
- tworzenie i zatwierdzanie – na tym etapie następuje proces rozwoju kolejnej części produktu, oparty na najbardziej odpowiednim modelu, wybranym na podstawie oceny zagrożeń; kolejno – powstaje plan realizacji projektu i odbywa się następny etap zakończony planem całości projektu;
- ocena i planowanie – to etap, w którym recenzowany jest postęp prac oraz planowana jest kolejna bądź ostateczna faza projektu; ostatni obieg cyklu przynosi projekt szczegółowy, testy i realizację projektu oraz jego zamknięcie.

Model ten różni się od pozostałych wielokrotnym powtarzaniem poszczególnych faz podstawowego modelu cyklu życia oraz – po każdym „podcyklu” – następuje analiza ryzyka zakończonej sukcesem wykonalności całego systemu w aktualnym momencie rozwoju (zob. rys. 8).





**Rysunek 8.** Schemat postępowania w spiralnym modelu cyklu życia projektu

Źródło: opracowanie własne na podstawie: Boehm, 1981.

Pomimo powyższych niedostatków, wysoka popularność metod tradycyjnych, zwłaszcza realizowanych w cyklu liniowym, wynika z głębokiego (historycznie rozpoznanego w różnych środowiskach) rozpoznania sposobów wykorzystania tej metodyki, prostoty logiki wykorzystania, głębokiej (niekiedy wynikającej z proceduralizacji wcześniej podejmowanych intuicyjnie decyzji) formalizacji (w poszczególnych technikach), możliwości ewolucji (stąd model przyrostowy i ewolucyjny) oraz możliwości ograniczonej adaptacyjności (stąd szerokie wykorzystanie podczas wdrożeń systemów powielarnych).

Bezpośredni, iteracyjny kontakt z końcowym użytkownikiem zaczyna występować dopiero w metodykach prototypowych, jest on jednak nadal ściśle reglamentowany, sformalizowany i podporządkowany początkowej koncepcji.

Jeszcze gorsza sytuacja kształtuje się w zakresie analizy ryzyka potencjalnych zmian w trakcie realizacji projektu. Przybliżone podejście do tego problemu występuje już, co prawda, w wariantach metody liniowej (w modelu ewolucyjnym, podzielonym na moduły łatwiej jest nadążać za zmieniającymi

się potrzebami użytkownika), ale jedynie w modelu spiralnym każdy cykl jest poprzedzany analizą możliwości osiągnięcia sukcesu.

Kwestie związane z wpływem czynnika ludzkiego na sukces projektu są rozpatrywane jedynie w grupie metod społecznych („miękkich”), jednak w warunkach środowiska, w których powstały nie mogły konkurować z pozostałymi metodami, ze względu na przyjmowane założenia, że bardziej istotne są czynniki techniczne („twarde”), łatwiejsze wówczas do pojęcia w aspekcie księgowym oraz fakt, że pozostałe metodyki były po prostu tańsze w wykonaniu.

Grupa metodyk tradycyjnych kształtowana jest od niemal pół wieku. Reżim, który większość z nich nakłada na rozwój projektu dyscyplinuje w pewnym sensie sposób postępowania w trakcie jego realizacji. Nie daje to jednak gwarancji, że projekt zakończy się sukcesem. Metody te są tak bardzo „sztywne” i ustrukturyzowane (po początkowym okresie intuicyjności), że przestrzeganie wszystkich kroków, formuł i procedur znacząco spowalnia cały proces rozwoju projektu, zwłaszcza dotyczącego systemów o średniej wielkości. Generalnie można powiedzieć, że charakteryzują się one następującymi cechami:

- przewidywalnym i powtarzalnym podejściem do procesu rozwoju projektu
  - w klasycznych metodykach zakładana jest zwiększająca się szczegółowość w miarę realizacji poszczególnych faz i etapów rozwoju projektu oraz obejmowanie całego okresu od początku do końca projektu; do analiz przeprowadzanych na bardzo niskim stopniu abstrakcji stosuje się na ogół deterministyczne techniki szczegółowe; wynik uzyskany za ich pomocą jest prawdziwy, dopóki przynajmniej jedno z założeń początkowych nie zostanie zmienione; zakłada się więc deterministyczny, niezmienny, nieadaptowany i mało elastyczny harmonogram, budżet i zasoby – na tej podstawie budowany jest pełen podział zadań dla każdego zespołu tworzącego produkt lub usługę;
- obszerną dokumentacją – tradycyjne podejście do realizacji projektu zakłada, że dokumentacja jest tworzona po każdej fazie, a często i etapie cyklu życia projektu; w najbardziej konwencjonalnej formie modelu kaskadowego przyjmuje się, że zakres oraz wymagania dotyczące użytkownika są zbierane i ustalane w pierwszej fazie cyklu (zob. powyżej), a następnie na ich podstawie tworzony jest produkt lub realizowana usługa; nie zmienia się tych założeń do końca cyklu życia projektu, część zebranych i przekształconych w zalecenia wykonawcze informacji i dokumentów wymaga przez to zmian w trakcie realizacji projektu;
- zorientowaniem na proces – celem klasycznych metodyk jest w zasadzie określenie procesu/ów, które będą uniwersalne i powtarzalne, czyli będą funkcjonowały w prawidłowy sposób (no i będą przydatne dla każdego, kto ich będzie używał), w każdej sytuacji, w której wydadzą się przydatne; każdy proces, składający się z zadań/czynności powinien być wykonywany

- według określonych z góry procedur przez określoną, przypisaną do niego i odpowiedzialną za niego grupę pracowników lub wykonawcę;
- zorientowaniem na narzędzia i techniki wspomagające realizację – do wykonania każdego określonego w projekcie zadania powinny być dostarczone odpowiednie narzędzia wspomagające zarządzanie.

### **1.5. Realizacja rozwoju nowoczesnych (zwinnych) metod projektowania systemów informatycznych w praktyce**

Niedostatki tradycyjnych metod projektowania systemów informatycznych są eliminowane albo przynajmniej ograniczane przez nowoczesne metody lekkie (zwinne – agile) projektowania. W pewnym sensie, biorąc pod uwagę również istniejące uwarunkowania zewnętrzne, prekursorami tych metod były miękkie metodyki społeczne. W *Maniście rozwoju zwinnego oprogramowania* (Beck, 2001) zwraca się bardziej uwagę na istotność zachowań ludzi i interakcji pomiędzy nimi niż procesów i narzędzi, działający rezultat projektu niż obszerną dokumentację, współpracę z klientem niż formalne ustalenia oraz reagowanie na zmiany niż podążanie za planem. Odpowiada to postulatowi wymienionych dotychczas zmian, które wynikają z ewolucji poglądów na same projekty i zarządzanie nimi. Głównym czynnikiem różniącym metody zwinne od tradycyjnych jest uznanie ludzi jako podstawowego czynnika sukcesu projektu, który połączony jest z intensywnym naciskiem na skuteczność i uwzględnienie zmian. Jest też to swoista odpowiedź na wyzwania biznesowe, powstałe w wyniku ukształtowania się szybko zmieniających się globalnych rynków.

Założenia metodyk zwinnych, które traktować można również jako główne różnice pomiędzy nimi a tradycyjnymi metodykami zarządzania projektami, są następujące (Awad, 2005):

- zorientowanie na interesariuszy projektu – jest to wg tych metodyk najważniejszy czynnik związany z rozwojem projektu, a jednocześnie zadanie dla kierowników zespołów „zwinnych” projektów; kładzenie większego nacisku na ludzi wraz z ich umiejętnościami, takimi jak: ambicje, zdolności oraz wzajemna komunikacja; jeżeli zespół nie jest zaangażowany w projekt, to żaden proces nie naprawi ich nieadekwatności;
- adaptacyjność – w tym podejściu kładzie się nacisk na zarządzanie zmianą; sprzyja to przekazaniu użytkownikowi własnej wiedzy większej niż minimalna zakładana projektem; zarządzanie zmianą zakłada ciągłą reakcję na ciągłe zmiany zachodzące projekcie; najtrudniejsze do oceny i reakcji są zewnętrzne zmiany środowiskowe – ponieważ nie jest możliwa ich eliminacja – należy dążyć do minimalizacji kosztów z nimi związanych;

- zgodność z rzeczywistością – zwraca się większą uwagę na zgodność otrzymanych rezultatów z uzyskanymi wynikami projektu niż zgodność z wynikami początkowo zakładanymi;
- elastyczność planowania – podstawowym problemem planowania projektu jest brak możliwości przewidywania implikacji rozwoju planów przedsięwzięć innowacyjnych, ponieważ środowisko, w którym powstają, jest wysoce dynamiczne; w „zwinnych” projektach wykonawcy muszą zastanowić się, jak mogą uniknąć nieodwracalności swoich decyzji – wymuszonych przyzwyczajeniem do praktyki szczegółowego projektowania tradycyjnego, które prowadzi do daleko posuniętej szczegółowości; zamiast próbować podjąć właściwe decyzje na początku każdego cyklu (projektowanie tradycyjne), lepiej ją podjąć w taki sposób, żeby w następnych etapach można było je odwrócić;
- oparcie się na procesach empirycznych – w metodach tradycyjnych procesy występują jako deterministyczne i liniowe, w metodach „zwinnych” jako proces empiryczny (probabilistyczny, źle lub słabo ustrukturalizowany) bądź nieliniowy; proces deterministyczny, taki, w którym od rozpoczęcia do zakończenia można za każdym razem spodziewać się takich samych wyników; projektów poprzez cechę wyjątkowości, jednorazowości itp. nie można zdefiniować jako procesów deterministycznych, ponieważ w czasie ich realizacji może się rozwijać i produkt i zespół projektowy; jest bardzo mało prawdopodobne, aby jakikolwiek zestaw predefiniowanych kroków mógł doprowadzić do pożądanego, przewidywalnego wyniku, ponieważ zmieniają się wymagania technologiczne oraz ludzie wewnątrz zespołu projektowego;
- wykorzystanie podejścia zdecentralizowanego – zdecentralizowany styl zarządzania może znacząco wpłynąć na projekt, ponieważ może on zaoszczędzić więcej czasu niż podejście autokratyczne; w metodykach lekkich deleguje się część zadań związanych z podejmowaniem decyzji do wszystkich członków zespołu (w rzeczywistości nie zawsze jest to jednak możliwe);
- prostota – w projektowaniu prowadzonym metodykami lekkimi zawsze wybierana jest najprostsza droga prowadząca do celu – zakłada się łatwe zmiany modelowe, które dostosowywane są do bieżących potrzeb i mogą następować w różnych terminach; nie wytwarza się większej funkcjonalności niż ta, która jest w danym momencie konieczna ani dokumentacji próbującej przewidzieć przyszłość projektu; to zmniejsza koncentrację na znalezieniu informacji potrzebnych do tych predykcji (Gibson i Hughes, 1994);
- komunikacja – oparcie się na współpracy z odbiorcą (użytkownikiem finalnym) i współpracy wewnętrznej – klient projektu powinien ściśle współpracować z zespołem projektowym, zapewniając mu wszelkie potrzebne informacje oraz zgłaszać bieżące uwagi i komentarze do projektu; ze względu na decentralizację zespół wykonawczy w metodykach „zwinnych” powinien się w sposób ciągły komunikować ze sobą;

- funkcjonowanie w małych, samoorganizujących się zespołach – gdzie obowiązki i zadania przekazywane są do zespołu jako całości, a zespół rozdziałając je, zapewnia najlepszy sposób ich realizacji; w małych zespołach najlepiej sprawdza się idea ciągłej komunikacji; struktura procesu i konkretne praktyki tworzą minimalne, elastyczne ramy strukturalne dla zespołów samoorganizujących; odpowiednie ich wykorzystanie znacznie zmniejsza ryzyko związane z czynnikiem ludzkim.

Do nowoczesnych (zwinnych, lekkich) metod projektowania zalicza się przede wszystkim: programowanie ekstremalne (XP – eXtreme Programming), Scrum, Feature Driven Development (FDD) – metodykę programowania rozwoju ukierunkowanego na wyróżnione cechy projektu (własności), Dynamic System Development Method (DSDM) – metodykę dynamicznego rozwoju systemu i Adaptive Software Development (ASD) – adaptacyjny rozwój oprogramowania. Idee, istota i praktyki wykorzystania tych metod, zaprezentowane poniżej, są zgodne z przedstawionymi założeniami, jednak nie zawsze pokrywają cały cykl życia systemu.

Podstawowe założenia i zaleceń metodyki XP (Beck i Anders, 2004):

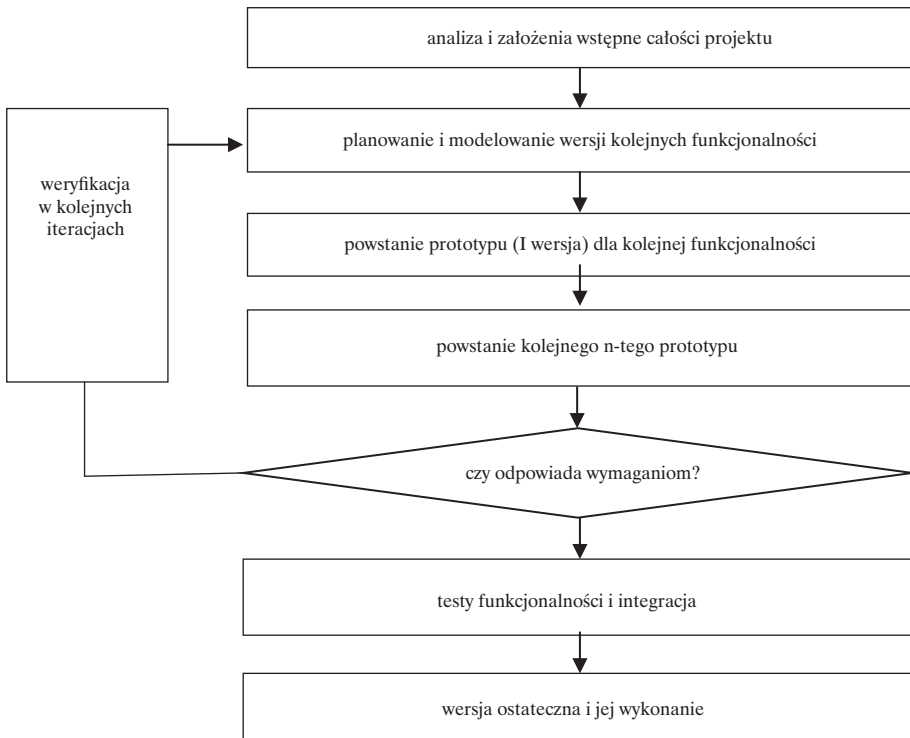
- specyfikacje projektowe są prawie zawsze niepełne, wieloznaczne, a nawet czasem sprzeczne wewnętrznie; co prawda zakłada się, że przy dobrym i stałym kontakcie ze świadomym klientem może się nawet obyć bez specyfikacji, ale dotyczy to tylko małych projektów;
- planowanie zwrotne – wykonawcy szacują czas niezbędny do realizacji zadań zgłoszonych przez klienta, klient je koryguje, po czym następują negocjacje, co do czasów i zadań, które można w tych okresach wykonać;
- iteracyjność – każda aplikacja tworzona jest w kolejno po sobie następujących iteracjach, każda z nich przybliża planowaną wersję do ostatecznych wymagań; system tworzy się w kolejnych iteracjach, planują tylko następną; po jej skończeniu i powstaniu wersji spełniającej przyjęte dla iteracji założenia planuje się dopiero następną iterację itd.;
- jednolity język komunikacji – opracowanie języka (zbiór, dziennik), w którym poszczególne kategorie mają to samo znaczenie dla wykonawcy i klienta;
- prostota architektury – uzyskiwany produkt powinien być jak najprostszyszy, a skomplikowane propozycje zastępowane mniej złożonymi; architektura jest jednak labilna – jeśli jej zmiana ma przyspieszyć bądź ułatwić realizację bieżącej iteracji i nie pogarsza wyników testowania, uzyskanych w uprzednich, to należy ją wprowadzić;
- refaktoryzacja – restrukturyzacja systemu poprzez usunięcie elementów dublujących się, poprawienie komunikacji czy uproszczenie modelu bez zmiany założonej funkcjonalności programu; wszelkie poprawki wykonywane są przed wprowadzeniem nowej funkcjonalności;

- praca dwójkami – naprzemienne wykonywanie przydzielonego wspólnego zadania, w celu poprawienia zastępowalności, wzajemnego uczenia się i kontroli poprawności, co zwiększa jakość wykonywanego zadania; obydwie osoby mają szansę dokładnie poznać kod źródłowy programu i po sobie poprawiać błędy (co eliminuje skutki informatycznego powiedzenia *...obys cudze programy poprawiał...*); może zmniejszyć wydajność pracy, przynajmniej początkowo; w praktyce często podział pracy w dwójkach jest inny niż zakładali autorzy metody (jedna osoba przygotowuje analizy i bieżące projekty; druga programuje);
- wspólna odpowiedzialność projektowa – z jednej strony każdy z członków zespołu może zmienić w dowolnym momencie poszczególne efekty dotychczasowych wyników projektu; z drugiej zaś – jest to poważną niedogodnością – brak jest jednej osoby odpowiedzialnej i każdy może „grzebać” w kodzie;
- natychmiastowe i ciągle integrowanie (*continuous integration*) nowych fragmentów pracy z powstającą całością oraz testowanie już zintegrowanych rozwiązań;
- samodyscyplina wyrażająca się przeznaczaniem określonego czasu na prace projektowe dla każdego członka zespołu zadaniowego oraz przestrzeganiem ustalonych na początku standardów wykonywania pracy – komunikacyjnych, formalnych i merytorycznych (w praktyce jest to często realizowane równie dowolnie, jak w metodzie ewolucyjnej);
- utrzymywanie stałego kontaktu z klientem – tworzone na podstawie analizy specyfikacje wymagań klienta są często wieloznaczne i niekompletne; należy zatem ciągle je korygować poprzez utrzymywanie kontaktów z klientami, którzy na bieżąco weryfikują uzyskane rezultaty.

Cykl życia projektu według modelu XP składa się z sześciu faz (zob. rys. 9):

- analiza i założenia wstępne – analiza opłacalności w świetle wyspecyfikowanych wymogów użytkownika i jego ograniczeń, budowa ogólnego modelu biznesowego i zadań stojących przed wykonawcą, wybór środowiska i narzędzi implementacji, negocjacje kontraktowe; etap wspólny dla całości projektu, nie zawsze uwzględniany w tej metodyce;
- planowanie i modelowanie wersji – przedstawienie dopuszczalnego wariantu/ów rozwoju projektu dla każdej funkcjonalności, w którym wykorzystywane są ustalenia poprzedniego etapu, rozpisanie projektu na zadania przedstawione przez klientów i przypisanie im priorytetów i umieszczenie w harmonogramie realizacyjnym;
- powstanie prototypu – na podstawie poprzedniego etapu, przedstawienie go klientowi, w kolejnych iteracjach po konsultacjach z klientem wprowadzenie zmian;

- powstanie kolejnego prototypu itd. aż do uzgodnienia wersji ostatecznej, w której następuje powstanie architektury i implementacji funkcji, zaprojektowanych w kolejnych wersjach;



**Rysunek 9.** Podstawowe fazy cyklu życia projektu w modelu XP

Źródło: opracowanie własne.

- testy funkcjonalności – kolejne wersje przedstawiane klientowi są testowane przed następnymi modyfikacjami, które może na tym etapie sugerować, po wypracowaniu ostatecznej postaci wersji każdej funkcjonalności następuje jej integracja z pozostałymi;
- dostarczenie ostatecznej wersji projektu i jej wykonanie – ostatnia iteracja doprowadza do stworzenia ostatecznej, kompletnej wersji projektu, która jest następnie realizowana.

Metodyka programowania ekstremalnego dotyczy głównie projektów małej i średniej wielkości, obciążonych dużym ryzykiem. Ponadto zamawiający projekt

(sponsor projektu) nie potrafi wyspecyfikować dokładnie ani celu tworzenia systemu informatycznego, ani jego pożądaných funkcjonalności, w związku z czym trudno jest stworzyć dokładny harmonogram projektu z przypisanymi do poszczególnych etapów zasobami. Metodą praktyczną postępowania w projekcie staje się więc działanie *per analogiam* (róbmy tak, jak w projektach podobnych, które zakończyły się sukcesem). Tak więc opiera się często na zastosowaniu w jednym projekcie czasem nawet sprzecznych przypadków „dobrych praktyk”, wykorzystaniu ich doświadczeń poprzez eliminację lub przynajmniej zmniejszenie błędów popełnionych przez poprzedników. Za największą wadę tej metodyki uważa się brak dokładnej specyfikacji dla całości projektu. Ma to być rekompensowane stałą obecnością i dostępnością przedstawiciela klienta, ale wydaje się, że może to prowadzić do rozchwiania systemu, gdy jest on osobą niekompetentną bądź tylko niezdecydowaną.

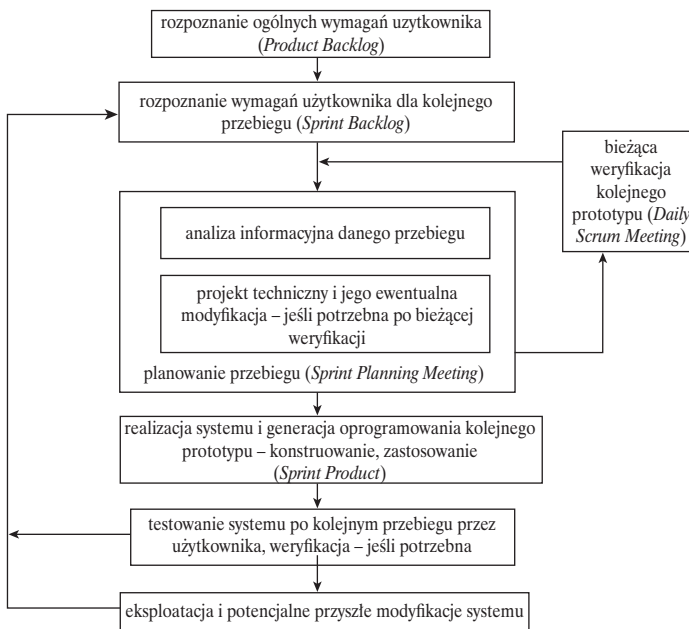
Nazwa metodyki Scrum nawiązuje do tzw. młyna w grze w rugby i sugeruje konieczność szybkiego dopasowywania się do zmieniających się dynamicznie okoliczności. Sprzyjają temu podstawowe, używane w metodzie Scrum, praktyki projektowania, które można sprowadzić do następujących:

- tworzenie rejestru zamówień (*Product Backlog*) – czyli lista wszystkich wymagań użytkownika: funkcji i ustalonych z realizatorem zmian wraz z priorytetami, czekająca na realizację (odpowiedzialny użytkownik końcowy – właściciel produktu (*Product Owner*));
- cykl pracy – przebieg (*Sprint*) – etap pracy zespołu projektowego (od jednego do sześciu tygodni, z zaleceniem regularności i jednolitości długości trwania każdej iteracji np. miesiąc); w każdym cyklu jest dostarczana użytkownikowi do przetestowania i oceny następną działającą wersją prototypu produktu;
- planowanie cyklu pracy – przebiegu (*Sprint Planning Meeting*) – składa się z dwóch części: analizy i projektu realizacji; w pierwszej – zespół projektowy, wraz ze wszystkimi użytkownikami ustala kompletny (w danym momencie) zbiór celów i funkcji systemu; w drugiej – kierownik projektu (*Scrum Master*) z zespołem uzgadnia najlepszy sposób realizacji produktu podczas danego przebiegu;
- tworzenie rejestru zamówień dotyczącego konkretnego przebiegu (*Sprint Backlog*) – lista nowych lub zmienionych funkcjonalności przypisana do kolejnego przebiegu; w momencie jej realizacji powstaje nowa wersja prototypu;
- weryfikacja postępu prac (*Daily Scrum Meeting*) – odbywa się w trakcie obowiązkowych codziennych spotkań zespołu projektowego; polega na identyfikacji niezbędnych zmian i określeniu warunków ich realizacji przez zespół (na zasadach samorealizacji).

Cykl życia w metodzie Scrum przebiega według następującego schematu (Schwaber i Beedle, 2001; Schwaber, 2005):



- rozpoznanie ogólnych wymagań i wstępna analiza informacyjna całości systemu,
- rozpoznanie i analiza dla kolejnego, bieżącego przebiegu,
- planowanie przebiegu (analiza i projekt techniczny),
- bieżąca weryfikacja założeń w trakcie codziennych spotkań,
- realizacja systemu i generacja oprogramowania kolejnego prototypu – konstruowanie i zastosowanie,
- testowanie systemu po kolejnym przebiegu,
- eksploatacja i potencjalne, przyszłe modyfikacje systemu.



**Rysunek 10.** Cykl życia systemu informatycznego w metodyce Scrum

Źródło: opracowanie własne, na podstawie: Schwaber i Beedle, 2001; Schwaber, 2005.

Przed każdym przebiegiem odbywa się spotkanie zespołu realizacyjnego z użytkownikami identyfikujące priorytetowe zadania (określenie zakresu, zawartości i intencji użytkownika) oraz konwertujące je w funkcjonalności przyszłego systemu. Na tej podstawie tworzony jest plan wykonania oprogramowania w bieżącej iteracji (priorytety, podział obowiązków, szczegółowe działania). Natomiast na początku każdego dnia przebiegu (w niektórych interpretacjach tej metodyki na zakończenie dnia) odbywa się – w trakcie spotkań zamkniętych zespołu projektowego – bieżąca weryfikacja realizacji zadań (stan wykonania,

problemy ogólne i szczegółowe realizacji oraz jednostkowe sposoby ich rozwiązania). Ma ona na celu koordynację i synchronizację dziennej pracy członków zespołu. Po każdej iteracji odbywa się natomiast spotkanie z użytkownikiem w celu prezentacji produktu kolejnej iteracji i określenia czy realizuje on kierunek zmian przez niego oczekiwany. Ma ono pomóc klientowi w określeniu czy i co w następnej kolejności powinno być wykonywane. Na podstawie wniosków z tego spotkania produkt zostaje przekazany do testowania i użytkowania lub w kolejnej iteracji do dalszych modyfikacji.

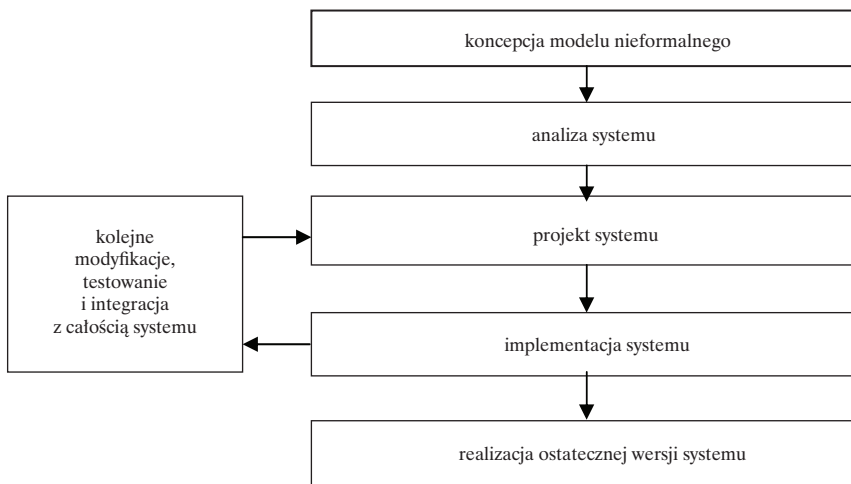
Założenia metodyki Feature Driven Development (FDD) – programowania zorientowanego na własności, skoncentrowanej głównie na fazie projektowania i implementacji dadzą się sprowadzić do następujących (Rising i Janoff, 2002, s. 26–32):

- głównym elementem jest cecha (*feature*) produktu – wydzielony zakres funkcjonalności projektu istotny z punktu widzenia klienta;
- lista cech jest budowana po stworzeniu ogólnego modelu biznesowego (obiektywne model nieformalny zawierający cele i ideę produktu oraz jego założenia i alternatywne rozwiązania); jej zawartość musi pokrywać się z wymaganiami dostarczonymi przez klienta (użytkownika);
- na tej podstawie konstruowany jest plan implementacji cech określający w jakiej kolejności będą realizowane cechy produktu oraz harmonogram jego realizacji wraz z przypisaniem poszczególnych zadań członkom zespołu projektowego;
- zastosowanie procedury „interpretacyjnej” polegającej na dostarczaniu kolejnych, działających wersji produktu w iteracjach polegających na przeplataniu się faz projektowania szczegółowego wybranych cech produktu oraz ich implementacji, aż do uzyskania konsensusu z użytkownikiem końcowym;
- realizacja tej procedury odbywa się poprzez przydzielenie cech zakwalifikowanych do wykonania w danej iteracji dynamicznie tworzonym małym (2–3 osobowym) zespołom projektantów i programistów; każdemu z nich przypisywana jest klasa biznesowa związana z funkcjonalnością danej cechy; pozostali członkowie zespołu testują napisany fragment oprogramowania i integrują z resztą produktu.

Poszczególne fazy cyklu tej metody są następujące (zob. rys. 11):

- budowa koncepcji nieformalnego modelu ogólnego – skonstruowanie opisanego na ogół modelu ogólnej architektury systemu, swoiste założenia budowy systemu;
- analiza systemu – polega na skonstruowaniu specyfikacji użytkowych dla wyróżnionych, niewielkich, użytecznych cech systemu, typu elementarnego, które są następnie grupowane w obszary funkcjonalne, a w razie potrzeby i dziedziny;

- projekt systemu – zgodnie z powyższym tworzony jest, w uzgodnieniu z klientem, plan konstrukcji oprogramowania według wyselekcjonowanych cech realizowanych zgodnie z priorytetami użytkownika końcowego; często na tym etapie dodatkowo szacuje się pracochłonności wykonania poszczególnych modułów oprogramowania oraz ryzyko związane z ich wykonaniem; prace nad projektem odbywają się w tworzonym na czas każdej iteracji zespole, złożonym z przedstawicieli (właścicieli) klas rozpatrywanej właśnie grupy cech; zespół ten modyfikuje bądź uszczegóławia bieżący projekt i dopuszcza do implementacji;
- implementacja systemu – kolejna wersja systemu dla kolejnego zestawu cech jest przedstawiana użytkownikowi; w następnej iteracji albo jest modyfikowana zgodnie z sugestiami użytkownika, albo akceptowana do realizacji;
- dwie ostatnie fazy powtarzają się iteracyjnie do końca projektu; po każdej iteracji klientowi dostarczana jest kolejna wersja oprogramowania.



**Rysunek 11.** Cykl życia systemu informatycznego w metodyce FDD

Źródło: opracowanie własne.

Dynamic System Development Method (DSDM) jest mieszanką i rozszerzeniem zwinnego podejścia do tworzenia oprogramowania i praktyk znanych z metodyk iteracyjnych (prototypowych). W swoim podejściu nawiązuje do metodyki RAD (Rapid Application Development – szybkie tworzenie aplikacji) opartej na budowie kolejnych prototypów w projekcie z gotowych komponentów. Pozwalało to na rzeczywiście szybkie uzyskanie efektów programistycznych już we wczesnej fazie projektu.

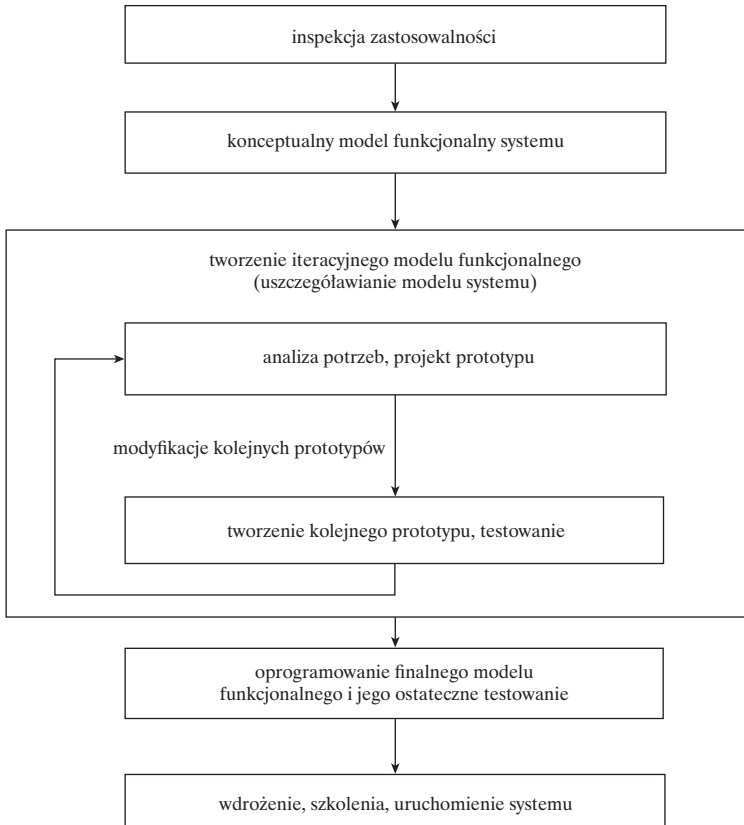
Główne założenia tej metodyki, opierające się na tym, że zadania systemu mogą podlegać ciągłym zmianom, są następujące:

- na początku projektu jednokrotnie przeprowadzana jest inspekcja zastosowalności, zawierająca uzasadnienie dla zastosowania tej metody oraz identyfikację potencjalnych zagrożeń dla jej pomyślnej realizacji;
- następnie tworzony jest model biznesowy obejmujący: opis (charakterystyka konceptualna systemu) i specyfikację zakresu systemu, zarys architektury systemu i plan prototypowania;
- na podstawie modelu biznesowego buduje się szczegółowy iteracyjny model funkcjonalny, polegający na naprzemiennym procesie analizy i budowy kolejnych (coraz lepszych, uzgadnianych z użytkownikiem końcowym) prototypów; wynikiem jest całościowy model funkcjonalny obudowany oprogramowanymi prototypami; w każdym przebiegu iteracyjnym tworzy się listę bieżąco opracowywanych funkcjonalności oraz dotyczących ich prototypów realizacyjnych; uwagi, komentarze, zalecenia – od użytkownika i/lub uzgodnienia z użytkownikiem; pozostałe wymagania niefunkcjonalne (organizacyjne, ekonomiczne, techniczne, psychologiczne, prawne itd.); analiza ryzyka opłacalności związanego z kontynuacją prac nad projektem;
- w kolejnych iteracjach powstają nowe, coraz bardziej zbliżone do wymagań użytkownika projekty;
- ostateczny model funkcjonalny zostaje oprogramowany, a przetestowane prototypy włączone w jego zakres (adaptowane); powstaje produkt zawierający uzgodniony wcześniej i przetestowany zestaw funkcjonalności;
- gotowy produkt w momencie wdrażania jest obudowywany dokumentacją końcową, instrukcjami i szkoleniami dla użytkowników.

Podstawową zaletą metodyki DSDM jest to, że na każdym etapie projektowania i budowy systemu produkt jest oceniany przez twórców i użytkowników, a uwagi wynikające z ich oceny opracowywane są w ramach kolejnych iteracji. Do innych zalet tej metodyki można zaliczyć również wysoką jakość i adaptacyjność wobec zmieniających się wymagań i, podobnie jak w innych zwinnych metodykach, krótki czas dostarczenia poszczególnych wersji produktu. Oprócz początkowej analizy ryzyka wykonalności, metodyka ta przypomina tradycyjne metody prototypowe.

Zasadniczym celem metody Adaptive Software Development (ASD) (Highsmith, 1999) – adaptacyjnego rozwoju oprogramowania jest wspieranie dużych, skomplikowanych i innowacyjnych oraz zmiennych w czasie projektów poprzez organizację pracy pozwalającą na łatwiejsze dostosowanie się do dynamicznego środowiska. Wykorzystywane jest przede wszystkim iteracyjne i przyrostowe tworzenie oprogramowania, w którym uwzględniono także „miękkie” elementy projektu – kulturę organizacji, negocjacje wymagań użytkownika, naukę itp.

Mniejszą wagę przywiązuje się do narzędzi wspomagających, formalnych procedur i wniosków wynikających z dotychczasowej praktyki projektowania.



**Rysunek 12.** Cykl życia projektu informatycznego w metodyce DSDM

Źródło: opracowanie własne.

Najważniejsze cechy charakterystyczne Adaptive Software Development, stosowanych do dynamicznych i często modyfikowalnych projektów informatycznych są następujące:

- oparcie się na dynamicznych spekulacjach (rozważaniach o możliwościach wariantowania i potencjalnych zmian), współpracy z użytkownikiem i wyciąganiu wniosków z bieżącej sytuacji (nauce);
- identyfikacja i wyjaśnienie wszelkich założeń niezbędnych do realizacji projektu (spekulacje);

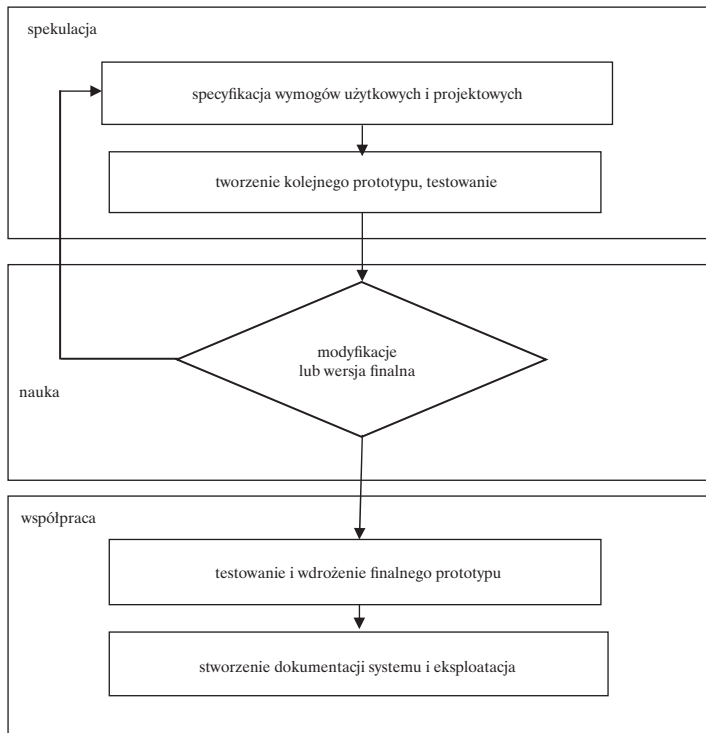
- ścisła współpraca oparta na natychmiastowej, szybkiej i efektywnej komunikacji pomiędzy członkami zespołu projektowego (i ewentualnie podzespołami);
- szybkie reagowanie na błędy i odchylenia od ustaleń projektu oraz ewentualne zmiany wymagań (nauka).

W odróżnieniu od podejścia tradycyjnego, w którym odchylenie od planu (spowodowane przyczynami obiektywnymi lub wymogami użytkownika) jest traktowane jako błąd do korekty, w podejściu adaptacyjnym, kreowanym przez ASD, takie odchylenia prowadzą do poprawnych rozwiązań, ponieważ są przyjmowane za poprawne i z góry pożądate. ASD nie posiada tak szczegółowych zasad i procedur, jak inne nowoczesne metody, a stanowi jedynie zespół wskazówek, pewne podejście do tego, jak należy zachęcać członków zespołu do współpracy i uczenia się w ramach projektu. Zamienia statyczny plan działania, na zmieniający się w czasie – dynamiczny, zależny od wyników poprzednich etapów (nadażny za zmieniającym się celem).

Cykl życia projektu informatycznego jest podzielony pomiędzy trzy zające się, iteracyjnie wykonywane fazy, podzielone na standardowe etapy projektu (Highsmith, 2004):

- spekulacji (inicjalizacja (rozpoznanie); definicja celu; ustalenie czasu trwania projektu, określenie maksymalnej liczby realizowanych w iteracjach przybliżeń kolejnych prototypów i czasu ich trwania; wybór zakresu funkcjonalnego i celu działania każdego prototypu; prototypowanie, dyskusja rezultatów każdej iteracji, dalsze modyfikacje lub zatwierdzenie postaci ostatecznej;
- współpracy – przetestowanie i wdrożenie ostatecznej postaci systemu, stworzenie dokumentacji w ścisłej współpracy z użytkownikiem systemu, dzielenie się z nim wiedzą i wspólne podejmowanie decyzji wdrożeniowych;
- nauki – po każdej iteracji następuje etap oceny jakości prototypu z punktu widzenia użyteczności dla klienta, innowacyjności i jakości zastosowanych rozwiązań technicznych, efektywności funkcjonowania zespołu projektowego oraz bieżącego stopnia wykonania projektu (jego statusu).

W zastosowaniu metodologii ASD podkreślana jest konieczność ścisłej i intensywnej w czasie projektu współpracy pomiędzy ostatecznym użytkownikiem i realizatorami projektu, co ma zapewnić tworzenie systemu informatycznego bliższego ostatecznemu, ustalonemu w trakcie adaptacyjnego procesu projektowania, zakresowi wymagań klienta. Swoista otwartość procedury postępowania wymaga, co prawda dla celów bezpieczeństwa sukcesywnej realizacji projektu, okresowej (zgodnej z cyklem iteracji) analizy ryzyka, ale pozwala to zachować możliwość wprowadzania zmian wynikających z procesu uczenia się. Takie podejście zwiększa ponadto pewność prawidłowego zakończenia projektu informatycznego.



**Rysunek 13.** Cykl życia projektu informatycznego w metodyce ASD

Źródło: opracowanie własne.

Podsumowując, należy dopatrywać się wielu podobieństw zarówno w poszczególnych, szczegółowych metodykach nowoczesnych, jak i wcześniej opisywanych metodykach tradycyjnych:

- pomimo deklarowanych różnic podobieństwo faz cyklu życia projektu i całej procedury projektowej;
- pomimo deklarowanej (*Manifesto for Agile...*) innowacyjności i odrębności „czerpanie” całą garścią z dorobku poprzednich metodyk twardych bądź socjopsychologicznych;
- pomimo wymyślnego nazewnictwa podobieństwo charakteru procedur postępowania;
- pomimo deklaratywnej szczegółowości podobieństwo w opieraniu się na danych przybliżonych, co wielokrotnie prowadzi do rozminięcia się oczekiwań użytkownika z wykonawcą.

Różny jest natomiast stopień objęcia typowych etapów cyklu życia projektu przez poszczególne metodyki. Proces ten przedstawia tabela 4.

**Tabela 4.** Pokrycie etapów cyklu życia systemu informatycznego przez wybrane metodyki zwinne

Etap cyklu życia/metody zwinne	Inicjacja/koncepcja	Analiza	Projektowanie	Oprogramowanie	Testy	Wdrożenie
Adaptive Software Development		X	X	X	X	
Dynamic System Development Method	X	X	X	X	X	X
Extreme Programming		X	X	X		
Feature-Driven Development		X	X	X		
SCRUM		X	X	X		

Źródło: opracowanie własne.

Przeprowadzona analiza wykazała, że istnieje wiele metodyk zarządzania projektami informatycznymi i brak jest jednej metody uniwersalnej, którą można by zastosować we wszystkich przypadkach, jakie przynosi nam praktyka gospodarcza. Wynika to nie tylko z faktu, że metodyki są tylko pewnym zbiorem wzorców, zasad oraz formuł, które pomagają w uniknięciu błędów, lecz zupełnie ich nie usuwają. Różnorodność przypadków generowanych przez rzeczywistość i zmienność wymogów użytkownika powoduje, że nie jest nawet możliwe dokładne dopasowanie metody do grupy przypadków. Niemniej jednak już sama świadomość istnienia różnorodnych sposobów rozwiązywania problemów powstałych podczas projektowania systemów informatycznych pozwala przynajmniej przybliżyć sposób postępowania zespołu projektowego do optymalnego oraz daje poczucie kontroli przedsięwzięcia, utrzymywania pełnego zaangażowania wszystkich zainteresowanych stron i uzasadnia poczucie bezpieczeństwa realizacji strategii biznesu sponsora.

Rozwój metod projektowania systemów informatycznych poszedł ponownie w kierunku wykorzystania podejść, chciałoby się powiedzieć bardziej intuicyjnych. Ale nie jest to stwierdzenie w pełni prawdziwe. Wypadałoby raczej powiedzieć, że nastąpiła, umożliwiona przez rozwój technologii informacyjnych, formalizacja miękkich (a więc i intuicyjnych) czynników decydujących o sukcesie projektu, takich jak współpraca, zaufanie, dobra komunikacja wzajemna itp. Trzeba też sobie powiedzieć, że nie dla każdego rodzaju projektu tego typu



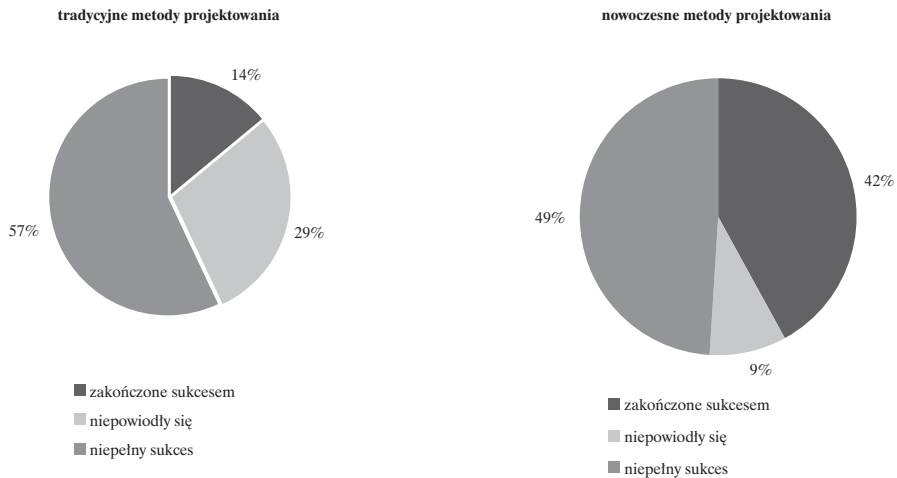
metodyki są możliwe do wykorzystania. O możliwościach zastosowań nowych rozwiązań w tym kierunku będą decydowały takie czynniki, jak:

- rodzaj i harmonogram finansowania prac zawarty w umowie z klientem – jeżeli zakłada się wykładniczy sposób finansowania, to należy stosować metodyki zwinne; w metodykach tych przyjmuje się, że czynnikami ważniejszymi niż negocjowanie kontraktów są zaufanie i współpraca partnerów biznesowych (przynajmniej deklaratywnie); w trakcie podpisywania umowy i wstępnego ustalania harmonogramu albo przyjmuje się realizację budżetu od początku korzystną dla klienta (powoli rosnąca na początku projektu, wykładniczo wraz z przekazywaniem uzgodnionych, przyjmowanych przez użytkownika produktów), albo opartą na kontraktach o niewielkim stopniu ryzyka dla realizatora projektu (np. umowy z refundowanym kosztem wg cen stałych); jeżeli natomiast zakłada się logarytmiczną funkcję realizacji budżetu (wysoką w stadiach początkowych, stabilizującą się w późniejszych fazach projektu), to – zwłaszcza dla firm o ugruntowanej pozycji na rynku opłacalne jest przyjęcie jednej z metodyk klasycznych;
- typ budżetu – nierównomiernie rozłożony w czasie budżet projektu, wymuszający różną intensywność prac oraz wysoki stopień niepewności, wskazuje na efektywniejsze użycie metodyki zwinnej; natomiast budżet finansujący projekt w sposób równomierny i stały w czasie sugeruje wykorzystanie metodyki klasycznej;
- charakter ustaleń harmonogramu – narzucone w harmonogramie sztywne założenia czasowe w stosunku do zaplanowanych zadań przemawiają na korzyść zastosowania metodyk klasycznych, przybliżone lub relatywne terminy realizacji – sugerują użycie metod zwinnych;
- ilość i poziom wymaganej dokumentacji oraz poziom jakości oprogramowania – spełnienie założonych wymagań związanych z formalnymi standardami dokumentacji, licencjami czy certyfikacji wskazuje na konieczność zastosowania klasycznej metodyki zarządzania projektem; natomiast projekt, w którym nacisk kładziony jest na jakość oprogramowania i możliwości jego rozwoju jest zgodny z filozofią metodyk zwinnych;
- podejście do ryzyka projektowego – niskie ryzyko projektowe obsługują lepiej metodyki zwinne; wysokie ryzyko w projekcie wymusza stosowanie metodyk klasycznych, w których zakłada się tworzenie planów minimalizacji ryzyka lub sytuacji awaryjnych;
- komunikacja z klientem – metodyki zwinne zakładają ciągłą oraz bezpośrednią komunikację z klientem; zmniejszane są więc wszelkie czynniki związane z niezrozumieniem realizowanych zadań; w metodykach klasycznych preferuje się rzadszy kontakt z klientem, co prowadzi bardzo często do nieporozumień w momencie prezentacji i przekazania zrealizowanego systemu;

- struktura organizacyjna konieczna do realizacji projektu – przedsiębiorstwa o strukturze hierarchicznej lub pokrewnych, zawierające w swoich strukturach ściśle wyspecjalizowane jednostki będą preferować metodyki klasyczne; przedsiębiorstwa o strukturze np. macierzowej, projektowej lub innej, w której mogą sobie pozwolić na delegowanie zadań projektowych do mniejszych zespołów, w których nie ma ściśle zdefiniowanej hierarchii organizacyjnej, powinny zdecydować się na wybór metodyki klasycznej;
- sektor/branża realizacji projektu – związana często ze strukturami organizacyjnymi też znacząco wpływa na realizację projektu; branże, które na wejściu mają niewielką liczbę surowców i materiałów, a na wyjściu gotowych produktów – lepiej nadają się do użycia w trakcie projektowania systemu metod klasycznych;
- wielkość projektu – część środowisk programistycznych związanych z wdrożeniami wielkich systemów uważa, że metody klasyczne lepiej nadają się do realizacji tego typu projektów; w metodykach zwinnych, trudno przy wielkich projektach, nawet powtarzalnych, skoordynować działania wielu małych grup realizujących projekt;
- rodzaj systemu, dla którego projekt jest realizowany – systemy eksperckie lub oparte na bazie wiedzy wymagają bardzo często specjalistycznej wiedzy o realizowanym zagadnieniu; jeśli są to systemy z założenia konstruowane jako samouczące zaleca się stosowanie metodyk nowoczesnych, w przeciwnym wypadku – dla systemów opartych na ustalonych wzorcach postępowania – metodyk klasycznych;
- czynniki psychologiczne – np. doświadczenie zespołu realizacyjnego w stosowaniu metodyk zwinnych lub klasycznych; zaufanie organizacji, w której projekt jest realizowany do zespołu projektowego; wysoki stopień wykorzystania w zespołach mieszanych specjalistów ze strony przyszłego użytkownika itp.

Cytowane już statystyki Standish Group (<http://www.controlchaos.com/storage/S3D%20First%20Chapter.pdf>) pokazują, że prawidłowe podejście do projektowania systemów informatycznych z użyciem metodyki nowoczesnej (agile) daje sukces w 42%, podczas gdy zastosowanie metod tradycyjnych tylko 14% (zob. rys. 14). Uważa się, że projekty sterowane metodami nowoczesnymi umożliwiły szybsze reagowanie na zmieniające się potrzeby klientów, pozwoliły na skuteczniejsze ograniczenie ryzyka i lepsze dostosowanie oprogramowania do wymagań organizacji.

Przeprowadzona analiza wykazała, że istnieje wiele metodyk projektowania i nie ma jednej metodyki uniwersalnej. Metodyka jest tylko pewnym zbiorem wzorców, zasad oraz formuł, które pomagają uniknąć błędów, lecz zupełnie ich nie usuwają. Jednakże konsekwentnie wdrożona metodyka daje poczucie kontroli przedsięwzięcia, utrzymywania pełnego zaangażowania wszystkich zainteresowanych stron oraz uzasadnia poczucie bezpieczeństwa realizacji strategii biznesu sponsora.



**Rysunek 14.** Porównanie sukcesu projektów realizowanych za pomocą metod tradycyjnych i nowoczesnych

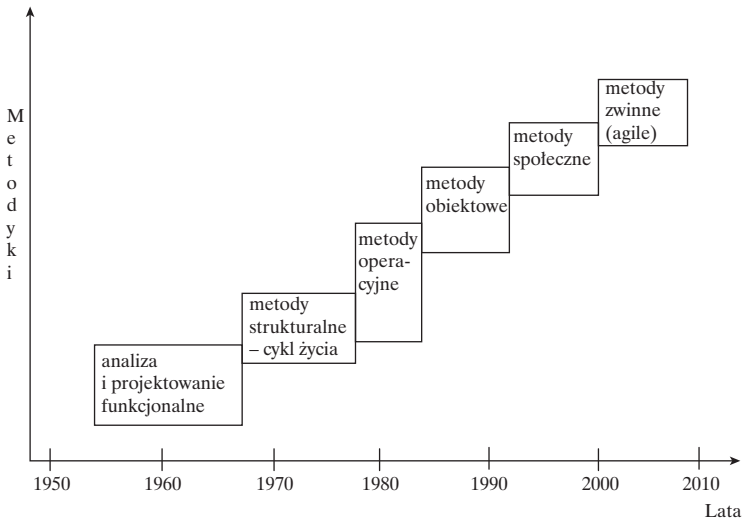
Źródło: <http://www.controlchaos.com/storage/S3D%20First%20Chapter.pdf>, za: Schwaber i Sutherland, 2012, rys. 1.2.

Dlatego też coraz więcej organizacji utrzymujących się z realizacji projektów jest zainteresowanych zarządzaniem przez projekt, polegającym w praktyce (upraszczając sytuację) głównie na równoległym prowadzeniu jak największej ilości projektów. Umożliwia to porównanie ich cech charakterystycznych (harmonogramów, kosztów, zasobów) realizowanych dla różnej wielkości firm, w wielu branżach, w wielu lokalizacjach. Pozwala to na wielowymiarową ocenę (w tym porównawczą, efektywności itd.) prowadzonych projektów i dostarcza kierownikom projektów wzorców dobrych praktyk zarządzania oraz umożliwia ocenę efektywności poszczególnych jednostek organizacyjnych prowadzących projekty. Analizy wykonania projektów dostarczają też przesłanek do współdzielenia zasobów projektów (w tym technologii, know-how itp.), rozliczenie czasu i kompetencji wykonawców oraz wycenę generowanych w projekcie produktów lub usług. Zarządzanie przez projekty jest stylem zarządzania doskonalącym przedsiębiorczość i będącym swoistym „złotym środkiem” pomiędzy rzeczywistymi potrzebami firmy zleceniodawcy (określonej przez jej specyfikę) a wiedzą (teoretyczną i praktyczną) oraz metodami projektowania (Stabryła, 2006). Taka pragmatyka podejścia umożliwia tworzenie wzorców postępowania podczas realizacji poszczególnych rodzajów projektów, co ułatwia tworzenie strategii nie tylko organizacji utrzymujących się z tworzenia i wdrażania projektów (i coraz częściej nie tylko informatycznych). To podejście do zarządzania zbiorami projektów może mieć różne formy i postaci.

W literaturze spotyka się następujące rodzaje nowoczesnego zarządzania zbiorami projektów, które, jak się obecnie wydaje, wyznaczają wiodące kierunki rozwoju zarządzania projektami (Pańkowska, 2010, s. 238–252; Cohen, Mandelbaum i Shub, 2004):

- zarządzanie programami, czyli uporządkowanymi zbiorami współzależnych projektów nadzorowanych przez instytucję zewnętrzną – cele projektów są narzucane przez sponsora (zarządzającego), projekty te są różne, ale wzajemnie zgodne w zakresie realizacji, realizowane przez różne, niezależne od siebie zespoły; wszelkie zasoby są ustalane i wyznaczane przez sponsora lub jego agendy, zarządzanie ryzykiem i jakością odbywa się na poziomie poszczególnych projektów i całego programu (Wysocki i McGary, 2003);
- zarządzanie multiprojektami – czyli grupą projektów, które są powiązane oceną efektów projektowych określanych przez odpowiednią strukturę organizacyjną (np. biuro/centrala zarządzania projektami); projekty są autonomiczne, ale mają uzgodnioną hierarchię celów, podobnie jak i zakres; występuje centralne rozdzielnictwo zasobów, przy czym decyzje budżetowe są proporcjonalne w stosunku do potencjalnych zysków poszczególnych projektów; ryzyko realizacji jest zarządzane z poziomu multiprojektu; zmniejszenie poziomu ryzyka jest możliwe dzięki przesuwaniu zadań pomiędzy projektami realizowanymi przez ściśle współpracujących ze sobą partnerów; jakość projektu zarządzana jest zarówno z poziomu centralnego, jak i poszczególnych projektów;
- zarządzanie portfelem projektów – polega, podobnie jak zarządzanie multiprojektem, na zarządzaniu kolekcją projektów, ale konkurencyjnych, dobieranych w taki sposób, by optymalizowały korzyści wynikające z zarządzania całym zbiorem; projekty są przyjmowane do portfela według rankingu określonego kryteriami ekonomicznymi zgodnymi z priorytetami całości projektu; zakres tych projektów jest nieuzgadniany wzajemnie; decyzje dotyczące budżetu są podejmowane w odniesieniu do zawartości całego portfela, a projekty rywalizują o zasoby; ryzyko jest oceniane na poziomie projektów, preferowane są zaś projekty jak najbardziej pewne w realizacji; projekty prowadzone są równoległe, a ich jakość oceniana jest niezależnie od portfela;
- projekty typu *roll-out* (porównania do wzorcowego produktu) – zestawy projektów o zakresie i określonych celach podobnych do wzorca (Flasiński, 2006), projekty są częściowo podobne, różnią się specyficznymi cechami środowiska, w którym są realizowane (np. wdrożenia powtarzalnych systemów informatycznych); każdy sponsor ustala swój budżet odrębnie; projekty mogą być realizowane sekwencyjnie lub równoległe (w tym przypadku rywalizują o zasoby); oceniane są zarówno przez sponsorów, jak i wykonawców, kontrolowane są jednak przez wykonawców.

Zarysowane i zasygnalizowane kierunki rozwoju zarządzania projektami wynikają z dynamicznych zmian, które w ostatnich latach następują w środowisku projektowym. Prowadzą one głównie do zwrócenia większej uwagi na zarządzanie zmianami prowadzące do zwiększenia dynamizmu, elastyczności i adaptacyjności projektów oraz transformacji relacji do zasadniczych interesariuszy projektów. Dzięki rozwojowi technologii umożliwiają integrację różnego rodzaju projektów innowacyjnych, które mogą być zarządzane w sieci projektów. Nie rozwiązują jednak wszystkich dylematów związanych z rozwojem zarządzania projektami zarysowanych w powyższym rozdziale.



**Rysunek 15.** Kierunki rozwoju metodyk projektowania systemów informatycznych

Źródło: opracowanie własne.

Z powyższych rozważań wynika więc raczej, że rozwój nowoczesnych metod projektowania systemów informatycznych wspomagających zarządzanie idzie raczej w kierunku uzupełniania i doskonalenia metodyk projektowania systemów w tych obszarach, które stosunkowo niedawno pojawiły się w rzeczywistości gospodarczej (np. systemy mobilne, systemy e-biznesu itp.), niż w kierunku całkowitego zastępowania klasycznych, tradycyjnych metod. Niezbędne zmiany, które w nich muszą nastąpić powodowane są raczej szybkim rozwojem technologii informacyjnych, a najbardziej systemów informatycznych typu CASE wspomagających procesy projektowania.

## 2

### Przegląd systemów usprawniania procesów biznesowych

---

Jak zauważono w poprzednim rozdziale, zastosowana poprawnie metodyka pomaga w zarządzaniu przedsięwzięciem informatycznymi, nie eliminując jednocześnie wszystkich potencjalnych błędów wynikających ze zróżnicowania procesów informacyjnych występujących w organizacjach gospodarczych. Dlatego właśnie metodyki obudowywane są szeregiem proceduralnie powiązanych technik i metod szczegółowych, zaopatrzonych niekiedy w daleko posunięte w swej dokładności przykłady zastosowań do polepszania procesów i skomplikowaną strukturę, stanowiąc wyrafinowane systemy usprawniania procesów biznesowych i organizacyjnych firmy. Niektóre z nich – trochę na wyrost, inne zdobywając krajowe lub regionalne certyfikaty jakości – są określane „standardami” projektowania, modelowania lub tworzenia procesów biznesowych. Jednak tak jak nie istnieje jedyna słuszna metodyka projektowania systemów informatycznych, tak nie istnieje jeden wyjątkowy i właściwy system usprawniania procesów biznesowych. A wszystkie korzystają pełną garścią z dorobku (ideologicznego i teoretycznego) metodyk uprzednio przedstawionych i związanych z nimi technik, co najwyżej wprowadzając drobne ulepszenia, niekiedy mniej istotne zmiany niewpływające na kwintesencję problemu. Pomimo agresywnej propagandy instytutów, producentów i dystrybutorów poszczególnych systemów usprawniania, ciągle pojawiają się nowe minimalizujące w określonych przypadkach błędy poprzednich i generujące – niestety – nowe, które w kolejnych systemach są eliminowane lub przynajmniej ograniczane zgodnie z nowymi technologiami. Na tym mechanizmie oparty jest postęp innowacyjny, nie tylko w tej dziedzinie. Dodatkowym problemem jest wysoka niejednorodność tych systemów – niektóre sytuują się na bardzo wysokim poziomie abstrakcji (wskazówki metodyczne), inne schodzą na poziom automatyzacji języków graficznych komunikacji projektanta z użytkownikiem. Nadal nierozwiązane są do końca, wspomniane wcześniej problemy definicyjne i klasyfikacyjne – zawsze wydaje się, że jak najbardziej pełna formalizacja procesów zapewnia kontrolę i zarządzanie zmianami, jednakże nasuwa się pytanie: czy jeszcze są to więc projekty (jako zbiór innowacyjnych procesów i ich relacji),

czy tylko zbiór znanych i przewidywalnych, często powtarzalnych procesów? Dlatego twórcy i propagatorzy niektórych systemów usprawniania zamiast sformułowania „projektowanie systemów informatycznych”, chętniej używają dla określenia ich przeznaczenia mniej zobowiązującego i bardziej bezpiecznego – „modelowania procesów w systemach informatycznych”. Modelowanie zawsze jest jedynie takim odzwierciedleniem rzeczywistości, które ma dać nam nową, a zintegrowaną i jednorodną wiedzę o jej cechach charakterystycznych w danej dziedzinie i służyć określonym celom np. wspomaganie procesów decyzyjnych. Modelowanie – póki nie jest przeniesione do świata rzeczywistego, nie pociąga za sobą skutków ekonomicznych – a na pewno jest tańsze niż eksperymentowanie na rzeczywistości. Może dostarczyć wiele wariantów do wyboru, które po rozpatrzeniu dodatkowych kryteriów, np. ekonomicznych, mogą potem być (lub nie) wykorzystane w procesie projektowania systemów. Skutkiem zaś projektowania systemów informatycznych jest określone oprogramowanie, po weryfikacji (testy, poprawki itp.) i wdrożeniu funkcjonujące definitywnie w danym momencie w określonej organizacji. Modelowanie jest postrzegane tu jako jedna ze złożonych technik projektowych, wyposażonych w dodatkowe, liczne, często zautomatyzowane narzędzia wspomagające procesy analizy i projektowania systemów informatycznych zarządzania.

Usprawnianie procesów biznesowych jest kategorią bardzo szeroką, która zawiera w sobie zarządzanie projektami informatycznymi, ponieważ standardowe metodologie zarządzania projektem można stosować w każdej dowolnej branży, czy dziedzinie ludzkiej działalności. A szczegółowe, konkretne procesy w ramach projektu zawsze można dopracowywać już za pomocą takich węższych lub uzupełniających metodologii i technik usprawniania procesów, jak CMM (Capability Maturity Model – model dojrzałości organizacyjnej), CMMI (Capability Maturity Model Integration – zintegrowany model dojrzałości organizacyjnej), *Six Sigma*, Prince2, czy ISO9000. Zarządzanie projektami jest metodologią strukturalną do oceniania, definiowania i zarządzania projektami – czyli jednym z najszerszych aspektów usprawniania procesów biznesowych (Snedaker, 2007). Wszystkie opracowywane i sprzedawane systemy zarządzania projektami stosują te same podstawowe zasady postępowania, bez względu na to, który z nich jest używany, mogą się różnić w szczegółach i uzupełnieniach. Można też wraz z nimi używać innych programów kontroli i poprawy jakości procesów.

Jednym z najistotniejszych warunków sprawnego i efektywnego wykonania projektu jest stosowanie jednorodnych procedur, wspólnej metodyki postępowania i zastosowanie ujednoczonej wiedzy ogólnej na temat tworzenia i realizacji projektu. Wzrasta też rola ujednoczonych procedur i metodyk postępowania, które nawet w projektach organizacji wirtualnych (gdzie poszczególne elementy projektu są odległe przestrzennie), gwarantują spójność rozwiązania.

Wypracowane i zweryfikowane metodyki realizacji systemów informatycznych, stają się narzędziem pomocnym w osiągnięciu sukcesów w zarządzaniu projektami informatycznymi (Szyjewski, 2004). Jak wspomniano, coraz częściej firmy informatyczne na własne potrzeby wypracowują oryginalne metodyki realizacji prac projektowych lub adaptują już znane istniejące i powszechnie akceptowane.

Metodykę określa się jako „...zbiór zasad dotyczących sposobów wykonywania jakiejś pracy...” (Bralczyk, 2008, s. 395), metodologię zaś – jako naukę o użyteczności, jakości i skuteczności badań naukowych. Oznacza to, że metodyka, w odróżnieniu od metodologii, stara się pokazać jak najlepiej należy tę pracę wykonać, a nie co należy robić – jakie są relacje zastosowanej metodyki do innych stosowanych w danej dziedzinie, jaka jest jej innowacyjność, w jakich przypadkach jest najbardziej efektywna w użyciu, czy jaka jest jej wartość poznawcza.

Można też ją określić jako „...ustandaryzowane dla wybranego obszaru podejście do rozwiązywania problemów. Metodyka abstrahuje od merytorycznego kontekstu danego obszaru, a skupia się na metodach realizacji zadań, szczególnie metodach zarządzania...” (<http://pl.wikipedia.org/wiki/Metodyka>).

Każda metodyka zbudowana jest na bazie przez wiele lat gromadzonej wiedzy praktycznej zebranej przez firmy. Można ją uznać za preliminarz działań obejmujący swym zasięgiem wszelkie czynności wykonywane w czasie trwania projektu. To formalny zapis, plan czynności oraz narzędzie pomagające kontrolować działania w projekcie. Metodyka jest też praktycznym przewodnikiem wskazującym, jakie czynności trzeba wykonać na poszczególnych etapach procesu projektowego. Stanowi ona także podstawę do weryfikacji działań projektowych, ich kontroli oraz rozliczania; to swoisty przewodnik po projekcie od stanu obecnego do stanu pożądanego (Chmielarz i Klineciewicz, 2010).

Zarządzanie procesami może być rozumiane jako zestaw technik i rozwiązań pozwalających sprawniej realizować założone cele projektów, ale również może być zastosowane jako metoda zarządzania organizacją. W pierwszym przypadku występuje jako zbiór dobrych<sup>1</sup> praktyk zarządzania, przydatnych, gdy pozwalają na osiągnięcie celów projektu bez istotnych zmian funkcjonowania organizacji. W drugim – zarządzanie przez procesy zastępuje niejako zarządzanie funkcjonalne organizacją, co wymaga szerokich zmian w organizacji. Najczęstszą zmianą jest ustanowienie właścicieli procesów oraz – na nowo – określenia ich miejsca i roli odgrywanej w organizacji. Towarzyszy temu spłaszczenie organizacji oraz powstanie nowych struktur wspomagających zarządzanie procesami (np. centra doskonałości Business Proces Management – BPM). Zarządzanie procesowe ma tu na celu również zapewnienie technik i mechanizmów pomagających,

<sup>1</sup> Czy naprawdę dobrych, to się w przypadku projektu często okazuje *ex post* (przyp. autora).



w oparciu o znajomość strategii organizacji, ustalać docelowy – pożądaný – stan organizacji (w tym jej procesów) oraz sposoby dojścia do niego ze stanu obecnego. Z kolei przejście od stanu bieżącego do hipotetycznego stanu końcowego może również być traktowane jako projekt, w którym metodyki zarządzania procesami mogą pomóc w osiągnięciu zakładanego celu, a realizacja projektu przejść do kolejnego etapu rozwoju organizacji.

Przedstawione w dalszej części rozdziału metodyki należą do zestawów praktyk w zakresie zarządzania projektami informatycznymi, rekomendowanych i serwowanych przez różne firmy lub instytuty badawcze oraz stosowanych w zróżnicowanych warunkach gospodarczych. Obejmują w zasadzie całość lub poszczególne etapy, a czasem i obszary procesu analizy i tworzenia systemu informatycznego. Różnią się również stopniem abstrakcji, na którym przedstawia się ich podstawowe założenia, poziomem zastosowania w organizacji, jak również kierunkami rozwoju.

Powracające w opracowaniu zblizenie metod zarządzania projektami i metod zarządzania procesami (na bazie postępującej formalizacji) sprawia zaś, że metody usprawniania procesów (w szczególności zarządzania nimi) coraz częściej stosuje się do zarządzania procesami unikatowymi i innowacyjnymi, czyli projektami. Dlatego, o ile jedne z nich uważa się za metodyki (a czasem ze względu na szeroki zakres badań i publikacji na ten temat, związany z ciągłym rozwojem – za metodologie zarządzania projektami), o tyle inne są traktowane jako metodyki zarządzania procesami, występujące samodzielnie bądź we współpracy z innymi w usprawnianiu procesów.

## 2.1. Metodyka Project Management Institute (PMI)

Project Management Institute (PMI) jest uważany za absolutnego lidera w zakresie zarządzania projektami w Stanach Zjednoczonych, który ustala standardy w tej dziedzinie. Jego szczególna popularność spowodowana jest olbrzymią ilością członków (od 301 tys. członków w 2009 r.<sup>2</sup>; 334 tys. w 2010 r.<sup>3</sup>; a już ponad 600 tys. w 2011 r. z 184 krajów<sup>4</sup>; a w 2012 ponad 650 tys. z 185 krajów<sup>5</sup>), szerokim i różnorodnym spektrum działania oraz możliwością wydawania certyfikatów w zakresie zarządzania projektami.

<sup>2</sup> <http://www.pmi.org/en/About-Us/About-Us-Annual-Report.aspx>; <http://entangled.com/2010/07/13/pmp-certification-growth-rate/>

<sup>3</sup> <http://www.pmi.org/en/About-Us/~media/PDF/Media/PMI%20AR%202010%20Combined%20FINAL.ashx>

<sup>4</sup> <http://www.pmi.org/en/About-Us/~media/PDF/Media/PMI%202011%20Annual%20Report%20-%20FINAL.ashx>

<sup>5</sup> <http://www.pmi.org/About-Us.aspx>

PMI koncentruje się na systematyzowaniu, klasyfikowaniu i rozwijaniu wiedzy oraz umiejętności z dziedziny zarządzania projektami. Wiele firm tworzy własne autorskie modele i systemy zarządzania projektami, jednak wszystkie z nich – jak się wydaje – bazują na podstawowych założeniach, metodach i procedurach wypracowanych i wciąż doskonalonych w Instytucie. Ogólność prowadzonych rozwiązań i stworzone ramy realizacyjne nadają się bowiem do zastosowania w jakichkolwiek projektach, bez względu na ich branżę, rozmiar, złożoność czy budżet. W ogóle większość powszechnie stosowanych (a szczególnie tych uznanych za standardy rekomendowane przez instytucje certyfikujące) metodyk w zakresie zarządzania projektami informatycznymi w wystarczający sposób normuje sposób kierowania projektem w zakresie opisanego poszczególnych jego faz oraz czynności, które należy w nim zaplanować i zrealizować. Istnieją także zbiory najlepszych praktyk w postaci kompendiów wiedzy dotyczącej zarządzania projektami, które mogą być pomocne w ich skutecznej realizacji (Kasperek, 2011). Nadal jednak za pierwszy, wiodący i najbardziej abstrakcyjny w sensie ideologicznym uważany jest produkt PMI – PMBOK Guide (ang. *A Guide to the Project Management Body of Knowledge*) (PMI, 2003), wydany początkowo w 1983 r. jako tzw. biała księga<sup>6</sup>, w 1996 roku jako oficjalne wydanie pierwsze, 2000 r. – drugie, piąte wydanie jest planowane na 2013 r.<sup>7</sup>. Jest on wciąż uzupełnianym zbiorem standardów i rozwiązań w dziedzinie zarządzania projektami, wykonanym i opublikowanym przez członków PMI (Project Management Institute). Standard PMBOK Guide to zbiór powszechnie uznanych praktyk znajdujących zastosowanie w zarządzaniu projektami.

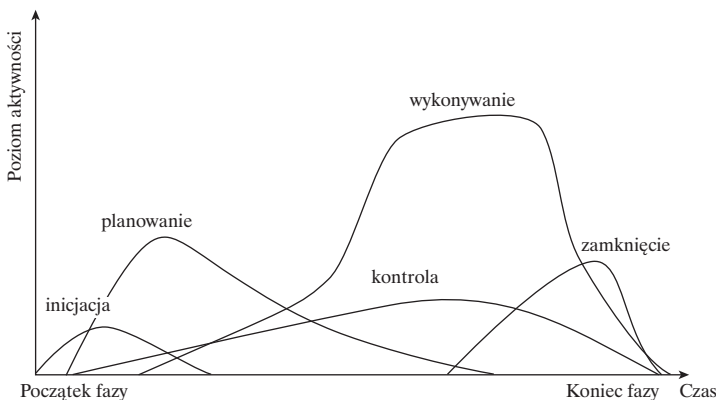
Dokument ten grupuje procesy dotyczące zarządzania projektami w pięć grup procesów (z podprocesami) i dziewięć obszarów wiedzy. Ze względu na brak wymogu sekwencyjności procesów w projekcie grupy procesów, choć prezentowane w określonej kolejności, mogą nachodzić na siebie w trakcie projektu. Grupy procesów i odpowiadające im podprocesy wymieniono poniżej (na podstawie: *PMBOK Manual...*):

- 1) procesy rozpoczęcia – procesy, które służą zdefiniowaniu i zatwierdzeniu projektu lub jego etapu w organizacji; w tym podprocesy: opracowanie dokumentu otwarcia, opracowanie wstępnego zakresu projektu;
- 2) procesy planowania – określanie i precyzowanie celów projektu oraz wybór najlepszego z dostępnych sposobów działania, pozwalających osiągnąć cele projektu – są to procesy odpowiadające na pytanie: jak, w jaki sposób zrealizować zamierzone cele, jakimi środkami, kiedy, w jakiej kolejności itp.; w tym podprocesy: opracowanie planu zarządzania projektem, planowanie zarządzania zakresem projektu, definiowanie zakresu projektu, opracowa-

<sup>6</sup> Biała księga – autorytatywny raport lub podręcznik pomagający czytelnikom zrozumieć problem, rozwiązać problem, lub podjąć właściwą decyzję.

<sup>7</sup> [http://en.wikipedia.org/wiki/A\\_Guide\\_to\\_the\\_Project\\_Management\\_Body\\_of\\_Knowledge](http://en.wikipedia.org/wiki/A_Guide_to_the_Project_Management_Body_of_Knowledge)

- nie struktury podziału prac, zdefiniowanie czynności, porządkowanie czynności, szacowanie zasobów czynności, szacowanie czasu trwania czynności, opracowanie harmonogramu, szacowanie kosztów, budżetowanie kosztów, planowanie jakości, planowanie zasobów ludzkich, planowanie komunikacji, planowanie zarządzania ryzykiem, identyfikacja ryzyka, jakościowa analiza ryzyka, ilościowa analiza ryzyka, planowanie reakcji na ryzyko, planowanie zaopatrzenia, planowanie kontraktów;
- 3) procesy realizacji – koordynacja ludzi i innych zasobów w celu wykonania przyjętego planu; w tym podprocesy: kierowanie i zarządzanie realizacją projektu, zapewnienie jakości, przyjmowanie członków zespołu, rozwój zespołu, dystrybucja informacji, gromadzenie ofert od sprzedawców, wybór sprzedawców;
  - 4) procesy kontroli – dbałość o realizację celów projektu poprzez systematyczne monitorowanie i mierzenie wykonania projektu, pozwalające na wykrycie odchyłeń od planu i podjęcie działań korygujących; w tym podprocesy: monitorowanie i nadzór nad pracami projektu, zintegrowane zarządzanie zmianami, weryfikacja zakresu, sterowanie zakresem, nadzór nad harmonogramem, nadzór nad kosztami, kontrola jakości, zarządzanie zespołem, raportowanie postępu prac, zarządzanie udziałowcami (interesariuszami), monitorowanie i nadzór nad ryzykiem, administrowanie kontraktem;
  - 5) procesy zakończenia – formalna akceptacja rezultatów uzyskanych w projekcie lub w danych etapie projektu oraz prawidłowe zakończenie projektu lub jego etapu; w tym podprocesy: zamknięcie projektu, zamknięcie kontraktu.



**Rysunek 16.** Nakładanie się grup procesów w ramach danego etapu projektu

Źródło: PMI (2004).

Poszczególne fazy nie są wykonywane w opisanej kolejności. W trakcie poszczególnych projektów bądź ich etapów przenikają się i nachodzą na siebie. Przykładowy sposób ich realizacji ukazany jest na rysunku 16.

Figurujące w wytycznych PMBOK procesy wykonania projektu są logiczną konsekwencją działań, które mogą być zrealizowane w trakcie każdego projektu. W zależności od rodzaju projektu, jego stopnia złożoności czy innowacyjności niektóre z nich mogą przybrać charakter bardziej szczegółowy niż inne, a w końcowych przypadkach nawet zostać pominięte. Chociaż trudno kwestionować poszczególne procesy (czy podprocesy) albo wskazać inne postulaty wynikające z zarządzania organizacją z punktu widzenia istniejących w organizacji procesów.

Nieco inaczej są opisywane obszary wiedzy projektu. Mimo że prezentują one w miarę logiczne, konsekwentne i spójne podejście do realizacji projektu, nie zawierają jednak szczegółowych porad wynikających z procesowego zarządzania firmą, które by je łączyły bezpośrednio z wytycznymi dotyczącymi procesów. Obszary te kształtują się w sposób następujący:

- 1) zarządzanie integracją projektu,
  - 2) zarządzanie zakresem projektu,
  - 3) zarządzanie czasem w projekcie,
  - 4) zarządzanie kosztami projektu,
  - 5) zarządzanie jakością w projekcie,
  - 6) zarządzanie zasobami ludzkimi w projekcie,
  - 7) zarządzanie komunikacją w projekcie,
  - 8) zarządzanie ryzykiem w projekcie,
  - 9) zarządzanie zamówieniami w projekcie.
1. Obszar zarządzania integralnością projektu obejmuje opracowanie planu projektu, realizację tego planu oraz zintegrowaną kontrolę zmian. W PMBOK występuje specyfikacja informacji wejściowych dla tworzenia planu projektu, tak by stanowił integralną całość i był zgodny z pozostałymi przedsięwzięciami realizowanymi w organizacji. Analiza tych informacji (istniejąca dokumentacja, informacja historyczna, wytyczne organizacyjne, ograniczenia w realizacji projektu, założenia projektowe) nie obejmuje jednak procesów biznesowych i ich wpływu na przebieg projektu nowego systemu.
  2. W skład zarządzania zakresem projektu wchodzi: zapoczątkowanie projektu, planowanie zakresu, doprecyzowanie zakresu, weryfikacja zakresu projektu oraz kontrola zmian zakresu. Brak zarządzania integralnością projektu w odniesieniu do przebiegu procesów również skutkuje brakiem integralności zarządzania zakresem (a więc i wymaganiami) z przebiegiem biznesowych procesów organizacji.
  3. Zarządzanie czasem trwania projektu składa się z: identyfikacji działań, określania kolejności działań, estymacji czasu trwania działań, opracowywania

harmonogramu i kontroli harmonogramu. Brak jest wytycznych, aby w każdej z powyższych kategorii należało przeprowadzić analizę wpływów i zależności, odniesionych do przebiegu procesów biznesowych organizacji. Czasami uwzględnienie okoliczności wynikających z przebiegu procesów będzie bardziej oczywiste (np. proces księgowania i zamknięcia roku bywa zwykle brany pod uwagę przy planowaniu wdrożeń systemów księgowych), lecz zachodzący w zupełnie innym obszarze firmy proces cyklicznie przeprowadzanej akcji marketingowej, powodujący zwiększoną ilość księgowania wpływających na zaangażowanie zasobów projektowych, może już nie być dla osób biorących udział w zarządzaniu czasem projektu tak oczywisty (Wojtkiewicz, 2012).

4. Zarządzanie kosztami projektowymi zawiera: planowanie zasobów, estymację kosztów, budżetowanie kosztów i kontrolę kosztów. PMBOK dostarcza szczegółowych wytycznych zarządzania tym obszarem, ponieważ stanowi on jeden z najważniejszych elementów każdego projektu. Nałożenie planowanych kosztów osobowych na ogólną mapę przebiegu procesów biznesowych organizacji może jednak dostarczyć bardzo cennej wiedzy o planowanym rozpoczęciu procesu szkolenia, co może wpłynąć na np. konieczność zapłaty nadgodzin. Czynniki kosztowe ma też znaczenie przy analizie opłacalności projektu, która poprzedza zwykle decyzję o jego rozpoczęciu.
5. Zarządzanie jakością obejmuje: planowanie jakości, zapewnianie jakości i kontrolę jakości. PMBOK zaleca ciągłą analizę produktów projektowych pod kątem jakości. Jednak każdy projekt powinien zostać osadzony we wcześniej przyjętych miernikach jakości procesu zarządzania projektem. W zależności od nich, za dobry jakościowo projekt może zostać uznany ten, który doprowadził do zakładanego wsparcia procesów biznesowych lub ten, który pozostawił pewne założenia funkcjonalne niezrealizowane, ale zakończył się w przewidywanym terminie i wyznaczonym budżecie.
6. Zarządzanie zasobami ludzkimi składa się z podobszarów planowania organizacyjnego, pozyskiwania personelu i kształtowania zespołu. Zgodnie z PMBOK członkowie zespołu projektowego powinni posiadać odpowiednią wiedzę, doświadczenie, dobre cechy interpersonalne pasujące do całości zespołu. Jak również być odpowiednio zmotywowani do realizacji projektu. PMBOK proponuje kształtowanie zespołu projektowego przez pryzmat umiejscowienia funkcjonalnego w strukturach przedsiębiorstwa. Dostarczyć to ma, według omówionej metodyki, sił odpowiednio wykwalifikowanych do realizacji konkretnego projektu, a kształtowanie zespołów na bazie kompetencji funkcyjnych ich członków i obsadzanie kluczowych ról informatykami powoduje, że projekty wdrożeń informatycznych realizowane są przez zespoły cechujące się brakiem zainteresowania sukcesem projektu, ponieważ nie mają możliwości poprawy procesów biznesowych, w których uczestniczą.

7. Zarządzanie komunikacją, na które składa się: planowanie komunikacji, dystrybucja informacji, sprawozdawczość wyników oraz zamknięcie administracyjne. Metodyka PMBOK zakłada, że w projekcie zostaną wytworzone kompetencje zapewniające komunikację wewnętrzną oraz zewnętrzną decyzji podejmowanych w projekcie, postępów prac projektowych i przekazywanie innych informacji umożliwiających wpływ na otoczenie projektu.
8. Zarządzanie ryzykiem, w skład którego wchodzi: planowanie zarządzania ryzykiem, identyfikacja ryzyka, jakościowa analiza ryzyka, ilościowa analiza ryzyka, planowanie reakcji na ryzyko oraz monitorowanie i kontrola ryzyka. PMBOK określa ryzyko jako wszelkie wydarzenia, które mogą negatywnie wpłynąć na realizację projektu zarówno ze względu na zdarzenia wewnętrzne, jak i zewnętrzne. Wśród nich wyróżnia zdarzenia niepozwalające wykorzystać wszystkich możliwości projektowych oraz negatywnie wpływające na zaplanowane działania. Na identyfikację ryzyka wpływają: tolerancja interesariuszy projektu na negatywne zdarzenia, zidentyfikowane źródła ryzyka, potencjalne zdarzenia inicjujące wystąpienie ryzyka, szacunki kosztów wystąpienia ryzyka oraz wpływ tych czynników na harmonogram projektu. W PMBOK występują zaawansowane analitycznie metodyki w zakresie oceny prawdopodobieństwa wystąpienia ryzyka i analizy planów zapobiegających i naprawczych.
9. Zarządzanie zamówieniami – zawierające: planowanie zamówień, planowanie zapytań, zebranie ofert, wybór dostawców, administracja kontraktem oraz zamknięcie kontraktu. PMBOK dostarcza wnikliwej metodyki w zakresie procesu realizacji zamówień. Zasadniczo można ją wykorzystać we wszystkich procesach zakupowych przedsiębiorstwa. Na ogół jednak firmy posiadają już rozwinięte procedury zamówieniowe i to właśnie zarządzanie projektem musi je uwzględniać, a nie odwrotnie.

W przypadku zarządzania projektami informatycznymi, w których występuje tworzenie oprogramowania, metodyka PMBOK jest uzupełniana zasadami dotyczącymi inżynierii oprogramowania dostarczonymi w ramach opisanej poniżej metodyki CMMI.

## **2.2. Metodyka Modelu Poziomów Dojrzałości (CMM – *Capability Maturity Model*) oraz Zintegrowanego Modelu Poziomów Dojrzałości (CMMI – *Capability Maturity Model Integration*)**

Model CMI powstał na zlecenie Departamentu Obrony USA w celu rozwiązania problemów wynikających z jakości oprogramowania tworzonego na ich potrzeby w latach osiemdziesiątych dwudziestego wieku. Ze względu na

subiektywnie narzuconą pięciostopniową skalę dojrzałości do modelu tego zgłaszano liczne zastrzeżenia. Dlatego był nieustannie rozwijany, co w latach dziewięćdziesiątych zaowocowało stworzeniem systemu zintegrowanego, którego postać z 2006 r. (*CMMI – DEV (for Development)*) zawiera dwie poprzednie wersje ciągłą (*CMMI – CR – Continous Representation*) i etapową (*CMMI – SR – Staged Representation*) (Czarnacka-Chrobot, 2009). CMMI–DEV należy traktować jako zestaw najlepszych praktyk służących do tworzenia i wykorzystania oprogramowania, w całym cyklu życia produktu od powstania koncepcji, projekt, oprogramowanie, wdrożenie, po dostarczenie i utrzymanie. W obecnym modelu rozszerzonym (*IPPD – Integrated Product and Process Development*) rozpatrywana jest również rola zespołów projektowych w procesach zintegrowanych.

Metoda służy do całościowej oceny tworzenia i zastosowania oprogramowania dla organizacji poprzez zakwalifikowanie jej na jeden z pięciu (reprezentacja etapowa) lub sześciu (reprezentacja ciągła) poziomów dojrzałości. W modelu o reprezentacji etapowej są one następujące:

- Poziom 1. Początkowy (Wstępny – ang. *Initial*) – firma podejmuje wysiłki w celu zarządzania projektami IT (projektowanie, rozwój, integrowanie, testowanie);
- Poziom 2. Zarządzany (ang. *Manager*) – w firmie są podstawy zarządzania projektowego (zarządzanie wymaganiami, planowanie projektowe, monitoring i kontrola projektów, zarządzanie umowami z dostawcami, zapewnienie jakości projektów),
- Poziom 3. Zdefiniowany (ang. *Defined*) – w firmie są wystandaryzowane procesy zarządzania projektami informatycznymi (rozwój wymagań, definiowanie procesu, zarządzanie ryzykiem, integracja środowiska IT, zarządzanie dostawcami),
- Poziom 4. Mierzony-zarządzany ilościowo (*Quantitative Management*) – występuje zarządzanie ilościowe i mierzenie procesów zarządzania jakością projektów informatycznych;
- Poziom 5. Optymalizowany (ang. *Optimizing*) – w firmie zachodzi ciągle usprawnianie procesów zarządzania projektami informatycznymi (wdrażanie innowacji, ciągła kontrola i ulepszanie).

W obecnie obowiązującej wersji modelu CMMI wyróżnia się 22 obszary procesu (Software Engineering Institute, 2006; Flasiński, 2006). Każdemu z nich podporządkowano specyficzne cele, a każdemu celowi specyficzne – charakterystyczne dla każdego procesu – praktyki zarządzania. Oprócz nich jest grupa celów i praktyk, które nie są podporządkowane do żadnego pojedynczego procesu. Aby zrealizować określony obszar, trzeba osiągnąć przypisane mu wszystkie cele specyficzne i ogólne, natomiast osiągnięcie wszystkich celów dla zestawu obszarów procesu oznacza osiągnięcie określonego poziomu dojrzałości.

### 2.3. Metodyka Prince2 (*Projects In a Controlled Environment*)

PRINCE2 jest standardem najpopularniejszym w Europie, gdzie obecnie po licznych uogólnieniach, stanowi alternatywę dla PMBOK. Metodyka ta została oparta na doświadczeniach uzyskanych przez kierowników projektów z krajów anglosaskich. Powstała w latach siedemdziesiątych XX wieku pod nazwą PromptII (Project Resource Organisation Management Planning Technique II). W 1989 roku po jej sprzedaży firmie LBMS, brytyjska agenda rządowa Central Computer and Telecommunications Agency (CCTA, obecnie będąca częścią Rządowego Biura Handlu – Office of Government Commerce) opublikowała nowy standard – PRINCE (*Projects IN Controlled Environments*), określając go jako zbiór najlepszych praktyk zarządzania projektami informatycznymi. Prace nad Prince2 zostały ukończone marcu 1996, kiedy Biuro Wydawnictw Rządowych (*The Stationery Office*) opublikowało podręcznik pt. *Zarządzanie projektami zakończonymi sukcesem przy pomocy Prince2 (Managing Successful Projects with Prince2)* (<http://www.best-management-practice.com/Knowledge-Centre/Publication-Reviews/PRINCE2/?DI=622360>), a w 2009 roku została opublikowana najnowsza wersja tego standardu.

W Wielkiej Brytanii Prince2 jest faktycznym standardem prowadzenia wszystkich rodzajów projektów. Wykorzystywany on jest zarówno w sektorze prywatnym, jak i publicznym w taki sposób, by zadowolić obecnych użytkowników i wciąż doskonalić w oparciu o zbierane na bieżąco doświadczenia. Został opracowany dla projektów informatycznych, ale jego elastyczność powoduje, że jest stosowany również w projektach z innych dziedzin (<http://www.prince-officialsite.com/>; <http://www.best-management-practice.com/Publications-Library/IT-Service-Management-ITIL/ITIL-2011-Edition/>). Opracowanie i rozwój tej metodyki spoczywa w rękach konsorcjum specjalistów od zarządzania oraz ponad 170 organizacji prywatnych i publicznych (*Review Panel*). Trudno jest ocenić zasięg użytkowników – biorąc pod uwagę liczbę zdających egzaminy na Prince2 Foundation i Practitioner oceniano ich w 2009 r. na ponad 50 tys. osób (<http://www.best-management-practice.com/Knowledge-Centre/An-Interview-with/?DI=571463>).

Popularność tej metodyki wynika z zastosowania w niej podejścia procesowego do przygotowania, wykonania i zamknięcia projektu, umożliwiającego jednocześnie stosunkowo łatwą adaptację do konkretnych warunków i skali przedsięwzięcia informatycznego. Każdy etap prac realizowanych zgodnie z metodyką Prince2 wyznaczany i rozliczany jest ze względu na osiągnięte rezultaty. Każdy projekt w tej metodyce jest opisywany za pomocą elementów stanu wejścia oraz wyjścia wraz ze zdefiniowanymi celami, które mają zostać osiągnięte oraz działaniami na określonych etapach przejścia pomiędzy poszczególnymi stanami (OGC, 2010, s. 5–78). Metodyka ta obecnie znajduje zastosowanie w projektach realizowanych przez światowe korporacje oraz może



być użyta w projektach wykonywanych przez małe firmy usługowe. W postaci z 2009 roku jest bardzo elastyczna – pozwala odpowiedzieć na pytania *co?* i *dlaczego?* powinno być zrobione w projekcie, umożliwia natomiast szeroką interpretację (w sensie działań i technik) sposobów osiągnięcia celu (nie mówi *jak?* ma być uzyskany). Umożliwia strukturyzację projektu – jego podział na mniejsze, a więc łatwiejsze do zarządzania części składowe. Dużą wagę przywiązuje się do kwestii finansowych, począwszy od uzasadnienia biznesowego projektu, skończywszy na jego rozliczeniu ([http://mfiles.pl/pl/index.php/Og%C3%B3lna\\_charakterystyka\\_metodykiPrince\\_2](http://mfiles.pl/pl/index.php/Og%C3%B3lna_charakterystyka_metodykiPrince_2)). Metodyka Prince2 definiuje projekt jako: „...środowisko zarządzania stworzone w celu dostarczenia jednego lub większej liczby produktów biznesowych stosownie do specyficznych wymagań stawianych ze strony użytkowników świata biznesu...”. Rozwój metodyki podąża w kierunku ([http://www.best-management-practice.com/gempdf/PRINCE2\\_2009\\_Overview\\_Brochure\\_June2011.pdf](http://www.best-management-practice.com/gempdf/PRINCE2_2009_Overview_Brochure_June2011.pdf)):

- mniejszej nakazowości i większej elastyczności;
- mniejszego oparcia na teorii, a większego na dobrych praktykach zarządzania;
- zasady używane w metodyce stają się mniej sztywne i powierzchowne, a bardziej zbliżone do rzeczywistości;
- dostosowanie metody do poszczególnych przypadków jest wyraźnie określone poprzez udostępnienie wytycznych dotyczące sposobu tej adaptacji;
- produkt jest zintegrowany z innymi programami firmy zapewniającymi zarządzanie udanymi produktami (*Managing Successful Programmes*), zarządzanie ryzykiem (*Management of the Risk*) itp., a połączenie z innymi dostępnymi użytkownikowi standardami i zasobami wiedzy jest jasno uwidocznione;
- rozwijane są i podkreślane miękkie aspekty zarządzania projektami;
- metodyka staje się mniej biurokratyczna – bardziej istotne stają się w niej wymagane informacje i sposób podejmowania na ich podstawie decyzji niż dokumenty i spotkania robocze.

W wersji z 2009 roku Prince2 składa się z następujących motywów (komponentów):

- uzasadnienie przedsięwzięcia,
- organizacja projektu,
- zapewnienie jakości,
- hierarchia planów,
- zarządzanie ryzykiem,
- zarządzanie zmianą,
- kontrola postępów projektu.

W uzasadnieniu przedsięwzięcia (*business case*) podaje się powody prowadzenia danego projektu, dopuszczalne warianty osiągnięcia celu projektu,

spodziewane korzyści z jego wykonania, listę zagrożeń i barier w realizacji, zestawienie kosztów projektu, harmonogram oraz spodziewane efekty ekonomiczne jego wykonania (ocenę opłacalności). Ostatnie działanie jest iteracyjnie powtarzane, aby mieć pewność, że pomimo zmieniających się celów biznesowych (użytkownik uczy się w trakcie projektu i zmienia czasem wymagania) działanie zakończy się osiągnięciem zysku.

W komponencie nazwanym organizacja projektu definiuje się zakres ról poszczególnych aktorów projektu (zarządzających i realizujących) oraz sugeruje (w wersji 2009 r.) zakres ich kompetencji i odpowiedzialności z niego wynikający. Obecnie dla Komitetu Sterującego oprócz obowiązków określa się również zachowania organizacyjne.

Metodyka określa osiem podstawowych ról w projekcie::

- Komitet Sterujący (*Project Board*),
- kierownik projektu (*Project Manager*),
- nadzór projektu (*Project Assurance*),
- kierownik zespołu specjalistycznego – rola opcjonalna (*Team Manager*),
- wsparcie projektu – rola opcjonalna (*Project Support*),
- główny użytkownik (*Senior User*),
- główny dostawca.

Komitet Sterujący reprezentuje projekt na zewnątrz wobec zarządu i składa się z osób, które prowadzą strategiczne sterowanie projektem i decydują o przydziale zasobów do projektu (szczególnie finansowych) oraz nadzorują projekt.

Przewodniczący Komitetu Sterującego reprezentuje interesy wspólne dostawcy i odbiorcy produktu, rozstrzyga spory, przewodniczy posiedzeniom Komitetu oraz przewodzi projektowi. Personalnie odpowiada za jego prawidłową realizację.

Kierownik projektu jest odpowiedzialny za: planowanie i zarządzanie projektem, opracowanie i aktualizację planów, dostawy produktów, powierza wykonanie zadań, dokonuje odbioru, prowadzi dokumentację, rejestruje zadania projektowe, prowadzi aktualizację planów, prowadzi analizy zmian, przedkłada raporty Komitetowi Sterującemu, przewodzi i motywuje pracowników zaangażowanych w projekt.

Kierownik zespołu specjalistycznego odpowiada za opracowanie planów oraz sterowanie pracami, a także składa raporty kierownikowi projektu z osiągnięcia kamieni milowych (punktów kontrolnych) projektu.

Członkowie zespołu specjalistycznego powinni posiadać wiedzę i umiejętności konieczne do wykonania produktu, wytwarzać je według założonych harmonogramów, odpowiadać za przygotowanie do kontroli jakości i zgodność jakości wytwarzanych produktów z jakością normatywną.

Wsparcie projektu – udziela pomocy administracyjnej, nadzoruje standardy zarządzania projektem i jakością, prowadzi dokumentację, dokonuje rejestracji

zagadnień projektowych, aktualizuje plany i analizuje skutki zmian, sporządza projekty dokumentów, protokoły z narad i przeglądów jakości.

Główny użytkownik produktu – reprezentuje interesy przyszłych użytkowników projektu, nadaje priorytety kolejnym produktom, dokonuje oceny i akceptacji produktów, nadzoruje spełnienie zasad funkcjonalności i jakości produktów, odpowiada za efekty i spodziewane korzyści akceptowanych produktów oraz sprawuje cząstkowy nadzór nad częścią projektu, która go dotyczy.

Główny dostawca produktu – reprezentuje interesy wykonawców produktu, przydziela i zwalnia niezbędne zasoby, odpowiada za integrację planowania i wykonawstwa specjalistycznych zadań projektowych oraz sprawuje specjalistyczny nadzór nad projektem i przyszłą eksploatacją i utrzymaniem produktu.

W komponencie „hierarchia planów” występują trzy poziomy plany: nadrzędny (plan projektu – *Project Plan*), cząstkowy (plan etapu – *Stage Plan*), szczegółowy (plan pracy zespołu – *Team Plan*). Oprócz tego występuje tzw. plan naprawczy (*Exception Plan*) w przypadku etapu, w którym pojawią się odchylenia przekraczające założone limity.

Zarządzanie ryzykiem – polega na eliminowaniu, minimalizacji lub utrzymywaniu ryzyka w założonych granicach. Składa się z takich elementów, jak identyfikacja ryzyka – (rozpoznawanie i rejestracja) oraz zarządzanie ryzykiem (minimalizacja negatywnych wpływów). Zarejestrowane rodzaje ryzyka są analizowane w trakcie postępu prac i zgodnie z planem redukcji ryzyka ograniczane.

Zarządzanie jakością polega na określeniu długookresowej polityki, która polega na dopilnowaniu (kontroli jakości), aby produkty spełniały założone wcześniej standardy kryteriów jakości założone (gwarancja jakości).

Zarządzanie zmianami w projekcie dotyczy drobnych modyfikacji, nienaruszających koncepcji biznesowych i architektonicznych. Obejmują: składanie wniosków o zmianę (*request for change*) – dotyczące zmiany w wymaganiach albo produkcie, rejestru odstępstw (*off specification*), sugestii zmian (*suggestions*), zapytań (*queries*) oraz definiowania problemów ogólnych związanych z modyfikacjami (*general issue*).

Zarządzanie postępowaniem – nie występowało w poprzednich wersjach, zastąpiło w zasadzie sterowanie projektem (*Control*) i zarządzanie konfiguracją (*Configuration Management*), wchłaniając ich funkcje. Z jednej strony ma zapewnić prowadzenie projektu zgodnie z zakładanym harmonogramem i budżetem, z drugiej – kontrolować integrację tworzenia produktów. Zawiera elementy oceny etapowej i końcowej.

Prince2 z 2005 roku nie pokrywał zakresu środowiska projektu. Obecny zawiera bogate wytyczne dotyczące dostosowania projektu do różnorodnych ograniczeń środowiskowych i rozszerzeń realizacyjnych, takich jak np.: projekty powiązane w programy, projekty zarządzania relacjami z klientami i dostawcami, projekty zintegrowane z innymi oraz działania w projektach o różnej skali.

- W metodzie Prince2 (2009 r.) wyróżniono siedem podstawowych procesów:
- rozpoznanie i przygotowanie założeń projektu (rozruch – *Starting up a project*),
  - zarządzanie strategiczne (ogólne) projektem (*Directing a project*),
  - inicjalizacja (uszczegółowienie) projektu (*Initiating a project*),
  - planowanie (*Planning a project*),
  - kontrola wykonania etapu projektu (*Controlling a stage*),
  - zarządzanie zakresem etapu projektu (*Managing a Stage Boundary*),
  - zamykanie projektu (*Closing a project*).

Istniejący w poprzedniej wersji proces planowania projektu został zlikwidowany, a jego podprocesy przydzielone i włączone do każdego z pozostałych procesów, z etapem ostatnim włącznie.

Rozpoznanie i przygotowanie założeń projektu (rozruch) polega na rozpoznaniu czy, w jaki sposób i w jakich warunkach finansowych projekt może być zrealizowany. Realizuje się go poczynając od zdefiniowania celów projektu i powołania zespołu kierowniczego, który te cele będzie osiągał, na uzasadnieniu ekonomicznym i określeniu wstępnych parametrów projektu (rozłożenie w czasie, budżet, elementy sterowania, uzgodnienia formalne) kończąc.

Zarządzanie strategiczne (ogólne) projektem sprowadza się do koordynacji i podejmowania decyzji kluczowych dla rozwoju projektu przez Komitet Sterujący, w szczególności czy należy projekt kontynuować. Decyzje bieżące są w kompetencjach kierownika projektu, z wyjątkiem (zarządzanie przez wyjątki – *management by exception*) sytuacji kryzysowych – wyjścia poza zakres dopuszczalnej tolerancji odchyień. W związku z czym rola Komitetu Sterującego sprowadza się głównie do wydawania zezwoleń na: inicjowanie projektu, wykonanie projektu lub etapu planu wyjątkowego, podejmowanie decyzji wynikających z zagrożeń i decyzji o zamknięciu projektu.

Inicjalizacja (uszczegółowienie) projektu – składa się z dokładnego zaplanowania jego realizacji i wyszacowania zasobów (środki finansowe, zasoby ludzkie, wyposażenie) niezbędnych do osiągnięcia celu. Jest to dalsze uszczegółowienie założeń początkowych, które znajdzie się później w podstawowym dokumencie tego procesu, tj. Dokumencie Inicjującym Projekt. Z tego wynikają podstawowe podprocesy, takie jak np.: doprecyzowanie uzasadnienia projektu, planowanie projektu, planowanie jakości, zdefiniowanie elementów decyzyjnych, struktury informacyjnej i dokumentacji projektowej oraz stworzenie Dokumentu Inicjującego Projekt.

W procesie kontroli wykonania etapu projektu realizowane jest bieżące zarządzanie projektem przez kierownika projektu. Ilość etapów zależy od branży, w której projekt jest realizowany, zakresu podjętych prac, skali przedsięwzięcia, poziomu i wartości ryzyka oraz ilości i znaczenia tzw. kamieni milowych – najważniejszych punktów decyzyjnych, w których podejmowane są decyzje

o opłacalności kontynuacji projektu. Po wyrażeniu zgody na wykonanie grupy zadań, odbywa się okresowa ocena postępów prac na podstawie przeglądu stanu danego etapu i jego zgodności z analizą zarejestrowanych zagadnień projektowych. W przypadkach udokumentowanych i uzasadnionych odchyłeń od zakładanych wartości podejmowanie decyzji o korekcie, zmianach, modyfikacjach bieżących, tworzenie raportów o istotnych wydarzeniach i odbieranie wykonanej grupy zadań.

Zarządzanie zakresem etapu projektu – każdy z etapów metodyki musi być zakończony i przyjęty przez Komitet Sterujący, przed przejściem do etapu następnego. Obejmuje opracowanie planu bieżącego zakresu etapu, jego uaktualnianie wraz z potencjalną korektą uzasadnienia biznesowego oraz podejmowanego ryzyka, opracowanie – w razie potrzeby planu naprawczego oraz tworzenie raportu o zakończeniu etapu.

Zamykanie projektu – polega na sprawdzeniu czy wszystkie zamówione produkty w postaci zgodnej z normatywami zostały wykonane i dostarczone użytkownikowi i przez niego zaakceptowane. Według metodyki Prince2 powinno się to odbywać w sposób uporządkowany i kontrolowany. Zbierane i rejestrowane są doświadczenia z projektu. Po jego zakończeniu przeprowadzany jest przegląd oceniający projekt, a wyciągane wnioski na przyszłość umieszczane są wraz z analizą z oceny w raporcie o dokonaniach projektu.

**Tabela 5.** Elementy metodyki Prince2 (2009 r.)

Procesy	Komponenty	Techniki
1. Rozpoznanie i przygotowanie założeń projektu (rozruch – <i>Starting up a project</i> ) 2. Zarządzanie strategiczne projektem ( <i>Directing a project</i> ) 3. Inicjalizacja (uszczegółowienie) projektu ( <i>Initiating a project</i> ) 4. Kontrola wykonania etapu projektu ( <i>Controlling a stage</i> ) 5. Zarządzanie zakresem etapu projektu ( <i>Managing a Stage Boundary</i> ) 6. Zamykanie projektu ( <i>Closing a project</i> )	1. Uzasadnienie przedsięwzięcia ( <i>Business Case</i> ) 2. Organizacja projektu ( <i>Organization</i> ) 3. Zapewnienie jakości ( <i>Quality</i> ) 4. Hierarchia planów ( <i>Plans</i> ) 5. Zarządzanie ryzykiem ( <i>Risk</i> ) 6. Zarządzanie zmianą ( <i>Changes</i> ) 7. Kontrola postępów projektu ( <i>Progress</i> )	Dwie podstawowe: 1. Planowanie oparte na produktach ( <i>Product Based Planning</i> ) 2. Kontrola jakości ( <i>Quality Review</i> )  oraz szereg innych, odwołujących się do zasobów wiedzy projektowej np.: 3. Zarządzanie zmianami ( <i>Changes Management</i> ) 4. Zarządzanie przez wyjątki ( <i>Exception Management</i> )

Źródło: opracowanie własne na podstawie: *Managing and Directing Successful Project...*, 2011.

Planowanie jest procesem trwającym przez cały cykl życia projektu, co sprawia, że jego podprocesy występują we wszystkich pozostałych procesach. Obejmuje więc przede wszystkim takie kroki, jak: przeprowadzenie identyfikacji, ustalenie kolejności wykonania czynności w celu osiągnięcia odpowiednich własności produktu (sekwencjonowanie), zdefiniowanie kolejnych zdarzeń w projekcie i ich wzajemnych relacji, szacunki pracochłonności, opracowanie harmonogramu działań wraz z oceną ryzyka za pomocą odpowiednich narzędzi.

Podsumowanie najważniejszych składowych metodyki Prince2 (procesów, komponentów i technik) znajduje się w tabeli 5.

Zalety metodyki Prince2 są następujące:

- zawiera uogólnioną, usystematyzowaną, częściowo sformalizowaną ekspercką wiedzę z zakresu zarządzania projektami;
- stanowi dobre medium komunikacji – jest metodyką powszechnie znaną, co ułatwia porozumiewanie się pomiędzy wszystkimi interesariuszami projektu i zespołami projektowymi, ponadto organizuje dobrze działające kanały łączności pomiędzy wszystkimi interesariuszami projektu;
- występuje w niej możliwość lepszego kontrolowania zużycia zasobów przez projektanta i odbiorcę projektu;
- umożliwiona jest poprawna strukturalizacja zadań od inicjacji po zakończenie;
- możliwy jest regularny monitoring postępu w pracach i automatyzacja kontroli odchyleń od ustalonego planu;
- rozłożenie punktów decyzyjnych w projekcie jest mniej sformalizowane niż w innych metodykach;
- zapewnia zaangażowanie Komitetu Sterującego poprzez odciążenie od zadań bieżących i daje kierownikowi projektu, dzięki zasadzie zarządzania przez wyjątki, możliwość działania bez zbędnych ingerencji Komitetu Sterującego, przy jego poparciu w sytuacjach nadzwyczajnych;
- dzięki jej stosowaniu występuje wysoka powtarzalność i standaryzacja, zwłaszcza projektów uniwersalnych, o wspólnej terminologii, podejściu i dokumentacji, co z jednej strony umożliwia powtórne zastosowanie w podobnych przypadkach, a z drugiej zapewnia możliwość doskonalenia kompetencji;
- standaryzacja jest wzmocniana przez dostarczanie kompletnych wzorców i wręcz szablonów dokumentacji o pełnej spójności i porównywalności;
- pomimo standaryzacji zawiera mechanizmy ograniczonej adaptacyjności do specyficznych wymogów organizacji lub projektu;
- opiera się na najlepszych praktykach zarządzania (w granicach rozsądku);
- jest powszechnie dostępna, co ogranicza wkład własny w tworzony projekt;
- nie wymaga opłat autorskich za stosowanie w projektach.

Wady metodyki Prince2, jakie dostrzeżono w trakcie korzystania z niej podczas realizacji projektów, wydają się być następujące:

- wyrywkowość stosowania metody, bez zwracania uwagi na podstawowe zasady w niej występujące;
- powszechna dostępność – co oprócz korzyści przynosi zaniechanie własnych wysiłków w celu doskonalenia metod zarządzania projektami;
- praktyka nadmiernego dokumentowania w procesach kontroli sposobu realizacji projektu (od której odchodzi się już od wersji z 2009 r. począwszy);
- występująca niekiedy nadmierna komunikacja (regularna wymiana informacji pomiędzy interesariuszami) w projekcie, gdzie spotkania robocze są tak częste, że przeszkadzają tokowi prowadzonych prac;
- brak bezpośrednio definiowanej analizy wymagań użytkownika, co z jednej strony przyspiesza prace projektowe, z drugiej zaś – może prowadzić do przyjęcia fałszywych bądź niepełnych założeń;
- nadmierna skala pracochłonności spowodowana zbyt literalnym przyjęciem zasad metodyki Prince2, przyjęta dla małych projektów może spowodować brak opłacalności jej zastosowania.

Prince2 przewiduje skomplikowaną, hierarchiczną organizację projektu, zakładającą istnienie wielu szczebli decyzyjnych, w tym podejmujących strategiczne decyzje projektowe. Struktura ta jest także zaangażowana w procesy monitorowania, analizy ryzyka i zapewniania jakości w projekcie. Metodyka zakłada, że projekt przez cały czas trwania musi udowodniać zasadność swojego istnienia: projekty, które straciły uzasadnienie, powinny być przerywane niezależnie od wielkości poniesionych już nakładów.

## 2.4. Metodyka ITIL (*Information Technology Infrastructure Library*)

ITIL (*Information Technology Infrastructure Library*) należy traktować jako modelową bibliotekę modeli będących najlepszymi praktykami w zakresie zarządzania usługami informatycznymi (*IT Service Management*), którą stosuje się jako pomoc w ich wdrażaniu. W skład tej biblioteki wchodzi określona mapa procesów oraz ich dodatkowych charakterystyk (pojęć, relacji, ról organizacyjnych, mierników) zawierających wiedzę ekspercką na temat całego cyklu życia każdej wyspecyfikowanej usługi informatycznej z perspektywy klienta oraz firmy informatycznej. Jest to de facto zbiór uogólnionych zaleceń i rekomendacji branży informatycznej, określających sposób efektywnego i skutecznego oferowania i wdrażania usług informatycznych. Z powyższych uwag wynika, że jest to metodyka:

- opisująca jedynie najlepsze praktyki zarządzania usługami IT, na podstawie których dopiero opracowuje się specyficzne dla danej firmy procedury; nie

- wdraża się natomiast gotowych instrukcji przygotowanych dla każdego typu organizacji;
- nie jest to bowiem narzędzie, a biblioteka procesów opisująca kompleksowo całą organizację IT, zawierająca ekspercką wiedzę na temat zarządzania usługami IT;
  - jest więc to zestaw porad, wytycznych i wskazówek jak unikać podstawowych błędów w zarządzaniu usługami IT i jakie w związku z tym wprowadzać kompleksowe rozwiązania sprawdzone w innych organizacjach, pamiętając przy tym o silnych relacjach między poszczególnymi obszarami występującymi w ITIL.

Materiały dotyczące metodyki ITIL ukazały się na rynku już w latach osiemdziesiątych jako zestaw około czterdziestu luźno powiązanych publikacji dotyczących wybranych najlepszych praktyk zarządzania. Pierwsza książka *HelpDesk* ukazała się w 1989 r. i wydana została przez rządową agencję Wielkiej Brytanii o nazwie OGC (*Office Government Commerce*) ([http://pl.wikipedia.org/wiki/Information\\_Technology\\_Infrastructure\\_Library](http://pl.wikipedia.org/wiki/Information_Technology_Infrastructure_Library); <http://www.itil.org/en/vomkennen/itil/index.php>; <http://www.15000.net/>; <http://www.itlife.pl/>). W następnych latach, współpracując z IT Management Forum, opracowano w 2001 r. drugą wersję metodyki, zebraną w siedmiu podręcznikach. Na podstawie przedstawionych w nich najlepszych praktyk opracowano brytyjską normę BS15000 *Service Management*, która z kolei stała się w 2005 r. podstawą normy ISO/IEC20000 (*Service Management*) (<http://itsm.itlife.pl/content/view/10037/82/>). W ostatnich latach wersja trzecia, opracowana w 2007 r., będąca aktualizacją poprzedniego modelu, odzwierciedlająca najnowsze poglądy na temat efektywnego zarządzania IT staje się jednym z uznanych standardów w tej dziedzinie (<http://www.itil-officialsite.com/AboutITIL/WhatisITIL.aspx>). Oficjalnym wydawcą OGC, odpowiedzialnym między innymi za druk i dystrybucję publikacji bibliotek ITIL, jest The Stationery Office (TSO). Przed opublikowaniem muszą one uzyskać akceptację The International Publications Executive Sub Committee (IPESC).

Metodyka ITIL, w wersji trzeciej, składa się z pięciu zintegrowanych tematycznie pozycji analizujących po kolei wszystkie etapy jej cyklu życia. Zgodnie z tym założeniem podręczniki zatytułowane są w następujący sposób ([http://pl.wikipedia.org/wiki/Information\\_Technology\\_Infrastructure\\_Library](http://pl.wikipedia.org/wiki/Information_Technology_Infrastructure_Library); <http://www.itil.org/en/vomkennen/itil/index.php>; <http://www.15000.net/>; <http://www.itlife.pl/>):

- *Strategia usług (Service Strategy)*,
- *Projektowanie i przeprojektowywanie usług (Service Design)*,
- *Wdrażanie usług i zarządzanie zmianą (Service Transition)*,
- *Eksploracja usług (Service Operation)*,
- *Nieustanne doskonalenie usług (Continual Service Improvement)*.



Strategia usług (*Service Strategy*) – faza początkowa cyklu życia usługi, podczas której wyznaczane są cele strategiczne, które mają być uzyskane przez realizację oferowanych usług oraz zasady działania wraz z zasobami niezbędnymi do realizacji zakładanych celów (Hanna, Windebank, Adams, Sowerby, Rance i Cartlidge, 2009).

Do procesów strategii usług zaliczamy:

- zarządzanie portfelem usług – proces odpowiedzialny za określenie potrzeb użytkownika i stworzenie dla niego oferty usług IT, uwzględniające jego korzyści; portfel usług składa się z takich elementów, jak: planowanie dostarczanych usług, katalog usług oraz rejestru wycofanych usług (zarchiwizowane w bazie wiedzy, specyficzne, zakończone już usługi, udostępniane w podobnych przypadkach) (zob. <http://itilservicestrategy.blogspot.com/>); zarządzanie portfelem usług składa się z czterech etapów:
  - definiowania (lista usług, studium przypadków, weryfikacja danych w portfelu);
  - analizy (proponowane wartości, uzgadniania, negocjowanie i ustalanie priorytetów, bilansowanie podaży i popytu);
  - zatwierdzanie (wyznaczanie i autoryzacja uzgodnionego portfela usług);
  - projektowanie (przekazanie decyzji i przydział zasobów do realizacji portfela usług IT);
- zarządzanie finansami – dostarczanie informacji o wartościach usług IT oraz prognozach ich wielkości (planowanie budżetu dostawcy usług IT, księgowanie kosztów dostarczenia usług, naliczanie opłat za usługi IT);
- zarządzanie popytem – identyfikacja wzorców zachowań użytkowników oraz polityki zróżnicowania opłat za pożądane zachowania klienta – tak, aby przynosiły największe korzyści dla obu stron (użytkownika oraz firmy dostarczającej usługi IT); klient nie będzie chciał korzystać z oferty, z której nie ma korzyści.

Projektowanie i przeprojektowywanie usług (*Service Design*) – na podstawie wymagań biznesowych oraz wytycznych z poprzedniej fazy odbywa się projektowanie nowych lub zmiana już istniejących usług, które będą wdrażane do eksploatacji. Zasadniczymi celami tego etapu są: projektowanie usług i obsługujących je procesów, projektowanie zasad bezpieczeństwa, projektowanie miar oceny (efektywności, użyteczności itd.), identyfikacja ryzyka projektu, tworzenie i/lub aktualizacja dokumentacji, projektowanie udoskonalania jakości usług IT.

Do procesów występujących na etapie projektowania i przeprojektowywania usług zaliczamy:

- proces udostępniania usług do wykonania – realizowany zgodnie z *Katalogiem usług* dzielącym się na: katalog biznesowy (zawierający informacje o usługach IT przekazywane i widoczne dla klienta) mapujących (nakładających) usługi

IT na procesy biznesowe oraz katalog techniczny (zawierający informacje o usługach dostarczanych klientowi, ale dla niego niewidocznych), mapujących usługi IT na infrastrukturę teleinformatyczną;

- proces zarządzanie dostępnością – zestaw działań odpowiedzialnych za efektywne udostępnienie, analizę, planowanie oraz doskonalenie dostępnością usług IT; jego realizacja polega na zapewnieniu zgodności udostępnionych, realizowanych usług z założonymi celami projektu w dwóch sferach pasywnej i aktywnej; w sferze pasywnej wykonywane są takie czynności, jak: monitorowanie, mierzenie efektów i analiza raportów dostępności usług oraz badanie ewentualnej niedostępności niektórych usług oraz działań potencjalnych naprawczych; w sferze aktywnej wyróżnia się: ocenę i zarządzanie ryzykiem, planowanie i projektowanie nowych lub zmienionych usług, wdrażanie czynności zapobiegawczych, przegląd nowych lub zmienionych usług, przeprowadzenie testów dostępności oraz odporności;
- proces zarządzania bezpieczeństwem informacji – zestaw działań odpowiedzialnych za: prawidłową dostępność (na żądanie), poufność (dla osób upoważnionych), integralność zasobów organizacji (modyfikację wyłącznie przez upoważnione osoby, informacja jest pełna i dokładna) oraz uwierzytelnianie oraz niezaprzeczalność (zaufanie transakcji);
- proces zarządzanie pojemnością – procedury doprowadzające do efektywnej (w zakresie kosztów i odpowiedniego czasu) zgodności charakterystyk technicznych (zdolności do wykonania usług) dostarczanych usług informatycznych i infrastruktury IT z uzgodnionym z klientem poziomem usług; w ramach tego procesu następuje analiza i przydział środków do świadczonych usług IT z trzech perspektyw: zarządzania zdolnościami dla realizacji usług dla klientów, zarządzania wykonalnością usług przez dostawcę, zarządzania charakterystykami komponentów ([http://www.itlibrary.org/index.php?page=Capacity\\_Management](http://www.itlibrary.org/index.php?page=Capacity_Management));
- proces zarządzanie dostawcami – czynności zmierzające do zapewnienia uzgodnionego wypełnienia zobowiązań przez dostawców;
- proces zarządzanie ciągłością usług IT – zarządzanie ryzykiem, narażającym na zakłócenie wykonania danej usługi; doprowadza do zapewnienia klienta, że dostawca zawsze jest w stanie dostarczyć usługę na poziomie akceptowalnym; cykl życia tego procesu składa się z pięciu etapów:
  - inicjacji (ustalenia polityki zapewnienia ciągłości, zdefiniowania zakresu, zainicjowania uruchomienia projektu);
  - analizy wymagań i ustalenia strategii ciągłości działań (analiza wpływu na biznes, ocena ryzyka, opracowanie strategii ciągłości usługi IT);
  - planowania ciągłości i wdrożenia (projektowania planów ciągłości usługi IT, projektowania planów IT, planowania organizacji, opracowania strategii testowania);

- uruchomienia i eksploatacji (szkolenia, podnoszenia świadomości użytkownika, przeglądu i audytu, testowania, zarządzania zmianami).

Wynikiem etapu projektowania usługi jest pakiet projektu usługi (*Service Design Package*), który zawiera wszystkie informacje dotyczące nowej lub zmienionej usługi, w szczególności dotyczące jej użyteczność oraz gwarancję działania.

Etap wdrażania usług i zarządzania zmianą (*Service Transition*) – polega na zarządzaniu i koordynacji wszystkich elementów potrzebnych do zbudowania, testowania i wdrożenia usługi u klienta, zgodnie z ustalonymi warunkami. Składa się z takich elementów, jak:

- zarządzanie zmianą – proces koordynowania modyfikacji w trakcie wdrażania usług IT z ustalonymi celami i harmonogramem; obejmuje on takie czynności, jak: rejestracja zmian, ocena ich istotności, wyznaczanie priorytetów realizacji, planowanie zmian, testowanie zmian, wdrażanie zmian i prowadzenie dokumentacji bieżącej tych czynności; w modelu ITIL wyróżnia się zmiany normalne, standardowe (wcześniej przewidziane, o niskim stopniu ryzyka, łatwe do realizacji, niskie koszty) oraz awaryjne (nagły, nieprzewidywalny i istotny błąd systemu, szybki tryb wprowadzenia poprawek);
- zarządzanie komponentami usług i konfiguracją – proces dostarczenie szczegółowych informacji na temat zasobów oraz zapewnienia kontroli nad zasobami i infrastrukturą organizacji; w trakcie procesu wykonywana jest sekwencja czynności, takich jak: zarządzanie i planowanie elementów usług i konfiguracji, jej identyfikacja i kontrola, obsługa bieżącego stanu i dostarczanie na ten temat raportów, weryfikacja oraz audyt komponentów i konfiguracji;
- zarządzanie wiedzą – proces polegający na: rejestracji, analizie wykorzystania oraz dystrybucji informacji i wiedzy w organizacji; podstawowym celem jest uniknięcie powtarzalności zdobywania wiedzy organizacyjnej;
- planowanie i wspieranie wdrażania usług i zarządzania zmianą – proces odpowiedzialny za zaplanowanie wszystkich procesów wdrażania usług oraz koordynację zasobów, dla ich realizacji; głównymi celami tego procesu są kontrola wraz z identyfikacją i zarządzaniem ryzykiem wystąpienia błędu wykonania usługi, a także planowanie i koordynacja zasobów realizacyjnych;
- ocena – proces polegający na ocenie zmodyfikowanych lub nowo wprowadzonych usług IT, wg wcześniej określonych miar i kryteriów, co uzasadnia wprowadzenie zmiany i jest weryfikacją poprawności podjętej w tym zakresie decyzji;
- zarządzanie wersjami i wdrożeniami – proces kontroli i planowania udostępnienia wersji testowego i eksploatacyjnej oraz przekazania nowego sprzętu, oprogramowania itp. niezbędnej do realizacji usługi IT;
- weryfikacja i testowanie usług – proces, który ma zapewnić poprawność usługi i jej zgodność z podpisanymi zobowiązaniami.

Eksploatacja usług (*Service Operation*) – faza cyklu życia polegająca na zarządzaniu systemami i infrastrukturą informatyczną, w celu dostarczenia usług IT użytkownikom oraz klientom na poziomie uprzednio uzgodnionym. Eksploatacja usług odbywa się poprzez realizację pięciu procesów:

- zarządzania incydentami – proces przywracania poprawnego działania usługi oraz usunięcie negatywnych skutków, jakie mogły się w międzyczasie pojawić;
- realizacja (spełnienie) wniosków – proces wspomagający użytkownika w dostarczaniu standardowych usług i skarg na ich niewypełnienie;
- zarządzanie uprawnieniami dostępu – proces przydzielania użytkownikom praw dostępu i uprawnień do wykonywania funkcji w ramach przydzielonych im usług IT;
- zarządzanie zdarzeniami – główny proces w tej fazie polegający na wykrywaniu, reakcji na zdarzenia, które występują w trakcie cyklu życia danej usługi;
- zarządzanie problemami – proces zapobiegający pojawianiu się kłopotów, a w momencie ich pojawienia powodujący minimalizację skutków; składa się z dwóch części pasywnej (identyfikacja i specyfikacja) i aktywnej (doskonalenie, tak aby zidentyfikowane problemy więcej nie wystąpiły).

Procesy występujące w tej fazie są realizowane w trakcie czterech podstawowych funkcji:

- pulpit usług – główne medium komunikacji do zgłaszania nieprawidłowości w działaniu systemu informatycznego oraz wniosków o udostępnienie określonej usługi;
- zarządzanie techniczne – wspieranie technicznych usług zarządzania systemem (*main frame*, serwerów, sieci, Internetu, pamięci masowej, baz danych, usług katalogowych, desktopu itd.);
- zarządzanie aplikacjami – odpowiada za prawidłowe działanie różnych rodzajów aplikacji oraz identyfikuje potrzeby podczas wdrażania aplikacji;
- zarządzanie eksploatacją IT – zarządzanie infrastrukturą informatyczną, kontrola eksploatacji (obsługa konsoli, przetwarzanie wsadowe, archiwizacja, wydruki) oraz zarządzanie wyposażeniem (centra danych, zapasowe centra danych, konsolidacja itp.).

Nieustanne doskonalenie usług (*Continual Service Improvement*) – jest to faza, która może wystąpić w każdym momencie cyklu życia realizowanych usług, odpowiadająca za podnoszenie jakości usług oraz związanych nią procesów, które są następujące:

- mierzenie jakości danej usługi IT – ustalenie mierników i prawidłowego sposobu mierzenia dla poszczególnych komponentów usługi IT;
- opracowywanie i raportowanie jakości dotyczących poszczególnych usług – publikacja wyników analiz mierzenia jakości usług;

- siedmiostopniowy proces doskonalenia jakości usług – składający się z identyfikacji strategii ulepszania (wizja, wymagania i ograniczenia, cele taktyczne i operacyjne wraz ze strategią ich osiągnięcia), identyfikacja ocenianych komponentów, zebranie danych dotyczących oceny (o wykonawcy, sposobie wykonania, terminarzu, wyznaczenie kryteriów spójności danych, wyznaczenie celów operacyjnych oraz miar jakości usług), przetwarzanie danych (ustalenie częstotliwości, formatu, narzędzi i systemów oraz dokładności obliczeń), analiza wyników (wynikające z niej trendy, nowe cele i kierunki działania, ewentualne wymagane działania usprawniające), wykorzystanie oceny (podsumowanie wyników i plan działań) oraz wdrożenie działań doskonalących.

### Zalety metodyki ITIL

Podstawowe zalety biblioteki polegają na (<http://itsm.itlife.pl/content/view/10005/50/>):

- możliwości wykorzystania zasobów zgromadzonej w niej wiedzy o prawidłowej realizacji procesów realizacji usług IT w innych organizacjach;
- minimalizacji kosztów wdrożenia – organizacja nie ponosi wysokich kosztów związanych z wdrożeniem tej metodyki, poza kosztami szkoleń dotyczących możliwości jej wykorzystania;
- polepszeniu komunikacji wzajemnej – używanie wspólnego, w pewnym sensie znormalizowanego języka oraz praca w dziale IT według wspólnych reguł;
- polepszenie relacji z klientem – skupienie uwagi na korzyściach ekonomicznych klienta już na początkowych etapach stosowania metodyki;
- zarządzanie jakością – ciągłe doskonalenie jakości organizacji oraz wykonywanych w niej usług informatycznych.

### Wady metodyki ITIL

Największą wadą według użytkowników jest fakt, że metodyka ta jest tym, czym jest, a więc biblioteką – listą czynności do wykonania w określonych przypadkach, nie daje zaś wskazówek co do tego, jak prawidłowo stosować najlepsze praktyki, które są w niej opisane. Każda organizacja musi więc, stosując zasady ITIL, opracować sobie własne procesy wraz z ich obsługą. ITIL jest więc tylko schematem postępowania, ramą, którą następnie należy wypełnić specyficzną dla danej organizacji treścią – nie każda organizacja ma pracowników, którzy są w stanie sami to zrobić. Firmy zatrudniają więc do konsultacji tych samych ludzi, którzy tworzyli zasady ITIL – zatem redukcja kosztów związana z zastosowaniem ITIL jest tylko pozorna. Poza tym stosowanie ITIL wymaga przeprowadzenia dedykowanych szkoleń, za które trzeba płacić,

a dostępne podręczniki opisujące metodykę są relatywnie drogie z perspektywy zastosowań niekomercyjnych.

## 2.5. Pozostałe metodyki wspierające usprawnianie procesów w organizacji

### 2.5.1. Metodyka *The Open Group Architecture Framework* (TOGAF)

Projektowaniem, utrzymywaniem i wprowadzaniem zmian do architektury IT zajmuje się metodyka *The Open Group Architecture Framework* (TOGAF) Version 9. W metodyce TOGAF występuje otwarta ramowa struktura implementacji – w zasadzie jest to szczegółowo opisany zestaw narzędzi wspomagających opracowanie architektury korporacyjnej. Może być używany w każdej organizacji pragnącej rozwijać architekturę korporacyjną (<http://pubs.opengroup.org/architecture/togaf9-doc/arch/>). TOGAF występuje w postaci ram architektonicznych (lecz nie jest to architektura), niezależnych od dostawców, powstałych w drodze konsensusu wypracowanego w ramach *Open Group*, które zawierają rygorystyczną metodę tworzenia architektury (*Architecture Development Method* – ADM), pozwalającą na przejście od architektury generycznej do architektury dedykowanej konkretnej organizacji, składa się z zestawu usług, standardów, pojęć projektowych, komponentów i konfiguracji (Sobczak, 2012).

TOGAF został opracowany i jest rozwijany przez członków tzw. *Open Group*. Pierwsza wersja metodyki TOGAF powstała w 1995 roku w oparciu o zasady *Technical Architecture Framework for Information Management* (TAFIM), opracowane przez amerykański Departament Obrony USA. Departament Obrony udzielił *Open Group* pozwolenia na przejęcie metodyki i jej rozwój w postaci kolejnych wersji już pod nazwą TOGAF.

W metodyce TOGAF występują następujące poziomy projektowania architektury informacyjnej przedsiębiorstwa (Wojtkiewicz, 2012):

- faza wstępna – określenie ustalenia zakresu działania organizacji, dla którego będzie realizowane przedsięwzięcie związane ze zwiększeniem innowacyjności (w tym wdrożenie rozwiązań IT) oraz identyfikowany jest sponsor tych prac;
- wizja architektury – utworzenie wizji organizacji, która prezentować będzie zakładany poziom dojrzałości w obszarze innowacyjności (w tym IT); następuje identyfikacja interesariuszy związanych z zapewnieniem realizacji celów i określenie ich potrzeb; rozpoznawane są główne czynniki ryzyka dla planowanego przedsięwzięcia, a także ustalany jest plan komunikacji dla tego przedsięwzięcia;
- architektura biznesowa – opisuje strukturę przedsiębiorstwa w odniesieniu do jego zadań biznesowych, struktury zarządczej, procesów biznesowych

- i informacji biznesowej; jest to wprowadzenie do ustalania architektury korporacyjnej;
- architektura danych (część architektury systemów informatycznych) – przedstawia strukturę i charakterystykę danych zaangażowanych do wsparcia architektury biznesowej (zadań biznesowych, procesów biznesowych itp.) oraz wskazuje systemy informatyczne używane do przetwarzania tych danych;
  - architektura aplikacji (część architektury systemów informatycznych) – ukazuje strukturę systemów informatycznych, ich wzajemne relacje i lokalizację w otoczeniu, a także informuje, w jaki sposób uczestniczą one w zadaniach opisanych w ramach architektury biznesowej i architektury danych;
  - architektura technologii – charakteryzuje technologie użyte do wsparcia elementów architektury biznesowej i architektury danych; określa też strukturę komponentów technologicznych i ich wzajemne relacje;
  - rozwiązania IT – prezentują sposób przejścia od stanu obecnego architektury do jej stanu docelowego; zawierają opisy niosące opis docelowy na opis stanu obecnego (architekturę przejściową), pozwalając określić plan działań;
  - plany migracji przedstawiają szczegółowy plan wdrożenia nowych systemów z uwzględnieniem scenariuszy migracji danych, zgodnie z rozwiązaniami określonymi na poprzednim poziomie;
  - zarządzanie wdrożeniem – opisuje sposób wdrożenia nowego systemu IT do architektury przedsiębiorstwa, uwzględniając powiązania z innymi komponentami architektury;
  - zarządzanie zmianą architektury IT – ma na celu utrzymywanie spójnego, aktualnego i zgodnego z przyjętym standardem opisu architektury podczas zmian dokonywanych na którymkolwiek z jej poziomów (<http://pubs.opengroup.org/architecture/togaf9-doc/arch/>).

Cztery ostatnie poziomy pokrywają się zakresem z opisanymi uprzednio wyspecjalizowanymi metodykami dotyczącymi zarządzania informatycznym projektem wdrożeniowym. Szczegółowość metodyki zostanie przedstawiona poniżej na przykładzie opisu poziomu architektury danych. Oczywiście architektura danych jest tylko jednym z poziomów opisu architektury przedsiębiorstwa wyodrębnionym przez metodykę TOGAF. Wydaje się jednak, że zbieranie i przetwarzanie danych stanowi jeden z najistotniejszych elementów budowania przewagi konkurencyjnej, a w organizacjach niekonkurujących na wolnym rynku jest często najważniejszym elementem umożliwiającym wykonywanie działań strategicznych. Architektura danych w metodyce TOGAF zawiera kilka poziomów opisu: konceptualny, logiczny i fizyczny. Poniżej przedstawiono ich krótką charakterystykę, zaczynając od poziomu najwyższego:

- konceptualny model danych (*Conceptual Data Model* – CDM) jest to rodzaj słownika porządkującego koncepcję architektoniczną organizacji; znaleźć

w nim można szeroki zakres kategorii danych (np. rejestr klientów, dostawców, inwentaryzację zakupów) użytkowanych w organizacji, stanowiących główne obiekty porządkujące informacje zbierane przez firmę; model ten zawiera również opis relacji pomiędzy danymi określony z perspektywy biznesowej, a więc niezależnie od sposobu, w jaki te dane są kategoryzowane przez przetwarzające je systemy informatyczne;

- logiczny model danych (*Logical Data Model*) to słownik zapisu danych; jest kolejnym etapem uszczegółowienia w stosunku do wyższego poziomu; przedstawia dane opisane na poziomie modelu konceptualnego w postaci konkretnych atrybutów danych, identyfikuje format tego zapisu i pokazuje powiązania pomiędzy nimi (np. nazwa, adres, rodzaj, powiązanie); by uzyskać ustrukturyzowany zapis, model ten normalizuje atrybuty danych w zakresie całej architektury danych; proces normalizacji polega na takim zorganizowaniu i opisaniu atrybutów danych, które zapewniają spójność i jednorodność tego zapisu, a także eliminują w opisie duplikaty atrybutów zawierających tę samą wartość informacyjną;
- fizyczny model danych (*Physical Data Model*) jest specyficznym słownikiem, tłumaczącym atrybuty modelu logicznego na ich techniczne odpowiedniki, które mogą zostać użyte w systemach zarządzania danymi.

Prawidłowo udokumentowana architektura danych ma ogromne znaczenie dla zarządzania zmianą w firmie. Pozwala na bieżąco uzyskiwać wiedzę o efektywności procesów biznesowych, wymieniać informacje ze światem zewnętrznym, przeprowadzać wdrożenia nowych komponentów informatycznych, wreszcie skutecznie zarządzać zmianą na wszystkich poziomach opisu architektury przedsiębiorstwa (Wojtkiewicz, 2012).

Korzyści z zastosowania metodyki TOGAF, widoczne w rachunku zysków i strat organizacji są przedstawiane w następujący sposób (<http://pubs.opengroup.org/architecture/togaf9-doc/arch/>):

- bardziej efektywne działanie firmy wyrażające się:
  - niższymi kosztami działalności gospodarczej,
  - elastycznością i adaptacyjnością organizacji,
  - udostępnieniem korzyści całej organizacji,
  - obniżeniem kosztów zarządzania zmianami,
  - zwiększeniem elastyczności zatrudnienia,
  - poprawą wydajności pracy,
- bardziej wydajne zastosowanie IT w organizacji:
  - zmniejszenie kosztów oprogramowania, wsparcia, utrzymania systemów informatycznych,
  - zwiększona mobilność aplikacji,
  - poprawa interoperacyjności i łatwiejsze zarządzanie systemem i siecią,



- poprawa zdolności do rozwiązania najważniejszych problemów w skali całego przedsiębiorstwa, takich jak np. bezpieczeństwo systemów informatycznych,
- łatwiejsza wymiana oprogramowania na nowsze wersje i wprowadzanie modyfikacji,
- zwiększenie zysków z bieżących inwestycji, zmniejszenie ryzyka dla przyszłych:
  - redukcja złożoność biznesu i IT,
  - maksymalizacja zwrotu z inwestycji w istniejącej działalności i infrastruktury IT,
  - zwiększenie elastyczności wyboru pomiędzy outsourcingiem a możliwością zakupu rozwiązań IT,
  - zmniejszenie ryzyka inwestowania,
- szybsze, prostsze i tańsze, zamówienia publiczne:
  - decyzji o zakupie są prostsze, ponieważ informacja o zamówieniach jest łatwo dostępna,
  - proces zamówień jest szybszy – występuje maksymalizacja prędkości oraz elastyczności tworzenia i składania zamówienia bez utraty spójności architektonicznej,
  - zdolność do tworzenia otwartych systemów wielu dostawców (*multi-vendor*),
  - możliwość do zwiększenia potencjału gospodarczego organizacji.

### 2.5.2. Metodyka *Six Sigma*

Sześć sigma (*Six Sigma*) jest to metoda zarządzania projektami skupiająca się głównie na zarządzaniu jakością produktów i/lub usług, poprzez minimalizację ilości niedociągnięć, i w rezultacie na usprawnieniu działania organizacji. Obecnie, jak wszystkie tego typu metodyki, obrasta ona w kwestie ideologiczne i filozoficzne. Zasadniczym, pierwotnym celem metody jest zwiększenie jakości do poziomu 99,99966% zgodności z założonymi, oczekiwanymi standardami. Odpowiada to średniej ilości 3,4 usterek na milion (liczba defektów na milion szans ich zaistnienia – *Defect per Million Opportunities* – DPMO), co stanowi sześć odchyień standardowych (stąd nazwa – sześć sigma, sigma litera najczęściej w statystyce używana do określenia właśnie odchylenia standardowego) od rozkładu normalnego wynikającego z obserwacji danego zjawiska (StatSoft Polska, 2012). Poziom jeden sigma – oznacza około 700 tys. defektów na milion, a dla firmy oznacza... katastrofę. Poziom trzy sigma jeszcze trzydzieści lat temu był uznawany za wystarczający standard jakości. Sześć sigma to doprowadzenie jakości niemal do perfekcji. Czyli, jak z tego wynika, celem *Six Sigma* jest zapewnienie niemal stu procentowej zgodności zaplanowanych rezultatów z rzeczywistymi wynikami. Wiadomo jednak, że większość zjawisk gospodarczych nie chce mieć rozkładu normalnego. Tak więc, ta nadwyżka

nad poziomem trzy sigma ma widocznie rekompensować zaburzenia założeń teoretycznych metody. Od strony filozoficznej *Six Sigma* stawia na zarządzanie zmianą: „...lepiej jest zarządzać problemem niż go potem eliminować...” (zob. [http://www.4pm.pl/artukul/czym\\_jest\\_six\\_sigma-26-1462.html](http://www.4pm.pl/artukul/czym_jest_six_sigma-26-1462.html)). Czyli – na wykrywanie błędów i przyczyn ich powstania oraz jak najpełniejsze ich eliminowanie zanim jeszcze się pojawią. Oczywiście, aby tym problemem zarządzać, trzeba dysponować dużą ilością mierzalnych, a najlepiej powtarzalnych obserwacji. Stąd drugie poważne założenie metodyki – to mierzalność zjawisk, które są brane pod uwagę przy podejmowaniu decyzji. Od strony ideologicznej (i zapewne marketingowej) chwytliwa dla klienta jest eliminacja niemal wszystkich błędów – tylko jaka kosztowna!

Metodyka ta została stworzona w połowie lat osiemdziesiątych XX w. w firmie Motorola przez B. Gavina, po odkryciu przez niego zależności pomiędzy ilością błędów w procesie produkcji a całkowitym kosztem wytworzenia produktów i usług. Identyfikacja defektów i eliminacja ich przyczyn spowodowała znaczące oszczędności, uzyskane dzięki zmniejszeniu kosztów wykrywania i usuwania usterek oraz potencjalnych, związanych z tym reklamacji. Od tego czasu jest ona stale doskonała i stosowana nie tylko do usprawniania procesów produkcji (Motorola<sup>8</sup>, Honeywell, Alstom, 3M, Microsoft, General Electric<sup>9</sup>), lecz także do wspomaganie administracji, finansów, ubezpieczeń itp. Powszechnie znana i uznana, daje najwyższy i najszybszy zwrot z inwestycji.

W metodyce *Six Sigma* występuje sześć podstawowych zasad (Barney i McCarty, 2005):

- skupienie się na kliencie i jego wymaganiach – rozpoznanie zidentyfikowanych i niezidentyfikowanych potrzeb klienta i ich mierniki;
- zarządzanie oparte na informacjach – określenie wskaźników poziomu działania firmy, kluczowych kryteriów oceny i optymalizacja rezultatów;
- zarządzanie procesami i ich ulepszanie;
- aktywne zarządzanie – wykonywanie czynności najlepiej zanim się coś wydarzy, antycypowalność zdarzeń i ich skutków;
- nieograniczona współpraca – zrozumienie potrzeb końcowych użytkowników i opanowanie przepływów dokumentacji (*workflow*) w całej organizacji;
- celem jest doskonalenie procesów, przy tolerancji dla niepowodzeń – ciągłe samodoskonalenie oraz uznanie dla konieczności popełniania błędów w procesie zarządzania.

<sup>8</sup> „...Motorola, lider konsorcjum organizacji, które rozwinęły podejście Sześć sigma, relacjonuje oszczędności w wysokości 11 mld dolarów, uzyskane w ciągu dwunastu lat od pierwszych implementacji Sześć sigma w jej zakładach na podstawie...”; zob. <http://www.statsoft.pl/sixsigma.html>

<sup>9</sup> Przykład: „...General Electric, który dzięki implementacji *Six Sigma* zaoszczędził 750 milionów dolarów w 1995 roku...”; zob. Snedaker, 2007, s. 23–24.

Jednak istotą *Six Sigma* nie jest tylko zastosowanie statystycznych metod do zmniejszenia błędów w produkcji. Podstawą jej stosowania jest procedura metodyczna, nazwana od pierwszych liter poszczególnych jej faz DMAIC:

- definiuj (*Define*) – na tym etapie następuje inicjacja i identyfikacja, poznanie i opisanie procesów oraz ich słabych stron, określenie celów, ograniczeń oraz obszarów występowania procesów, identyfikacja cech krytycznych i zadań do rozwiązania (czyli określenie problemów i procesów, które są przyczyną defektu i skoncentrowanie się na tych, których poprawa przyniesie największą korzyść); technikami analitycznymi na tym etapie mogą być diagramy przyczynowo-skutkowe, metoda ABC i kalkulator *Six Sigma*;
- mierz, wymiaruj (*Measure*) – to poznanie w sposób jasny, precyzyjny i dokładny obiektywnej prawdy o obiekcie, a wyniki pomiaru powinny być jasno i jednoznacznie interpretowane; polega na zebraniu informacji o aktualnym stanie procesu, dla ustalenia poziomu odniesienia i skali problemu, czyli zbioru cech procesu i działań nieakceptowalnych dla klientów (liczba błędów i szans na ich popełnienie), określeniu metod pomiaru wymagań klientów oraz przeprowadzeniu szczegółowych pomiarów (ocena potrzeb klienta lub rynku, porównanie z konkurencją itp.); techniki analityczne używane na tym etapie to: statystyki opisowe, wykres Ishikawy, wykresy z szeregów czasowych itp.<sup>10</sup>;
- analizuj (*Analyse*) – na tym etapie przeprowadzana jest analiza danych w celu zidentyfikowania przyczyn defektów oraz sposobu ich uniknięcia, wskazanie krytycznych powodów problemów i ich wpływu na proces, analiza i ocena pomiarów, specyfikacja odchyłeń i metod ich eliminacji (badanie różnych opcji procesu); obliczamy ile zyskamy na poprawie poszczególnych procesów przez eliminację ewentualnych błędów; korzysta się tu z diagramów przyczynowo-skutkowych, testów dla prób zależnych i niezależnych, analizy przekrojowej;
- poprawiaj, usprawniaj, udoskonalaj (*Improve*) – na tym etapie dokonywane jest wskazanie procesów do poprawienia i wprowadzenie rozwiązań usuwających wyspecyfikowane problemy, poszukiwanie nowych rozwiązań, minimalizacja odchyłeń, kreowanie zmian i ich wdrażanie lub projektowanie procesu – tak, aby spełniał oczekiwania klienta, ingerencja w tok procesu po to, by obniżyć poziom występowania błędu, poprzez zmniejszenie odchyłeń; elementem tego etapu jest planowanie, wykonanie i analiza wyników doświadczeń dotyczących kluczowych kategorii procesu; technikami analitycznymi są tu – karty kontrolne, analiza Pareto, kalkulator *Six Sigma*, analiza przekrojowa, wykresy przekrojowe;

<sup>10</sup> Galileo Galilei: „...Licz to, co policzalne, mierz to, co mierzalne, a to, co niemierzalne, uczynj mierzalnym...”.

- kontroluj, sprawdzaj (*Control*) (DMAIC, 2012) – oznacza tu zakończenie, weryfikację skuteczności podjętych działań i – w wyniku tego wprowadzenie nowych postaci procesów, identyfikację i monitoring wyników, w celu uzyskania wysokiego poziomu jakości poprzez określenie zakresu zadań oraz ról i odpowiedzialności za jakość właścicieli procesów lub testowanie sprawdzające czy proces spełnia wymagania klienta; najczęstszą techniką działania jest tu stosowanie kart kontrolnych.

W tradycyjnej metodyce zarządzania projektem występuje podział na etapy rozpoczęcia (inicjacji), planowania (definiowanie i analiza), realizacji, kontroli oraz zamknięcia analogiczne do etapów metody *Six Sigma*. Poszczególne z nich są wspomagane specjalnie stworzonymi i przystosowanymi do tej metody narzędziami dodatkowymi:

- ukierunkowanymi na klienta w celu rozpoznania jego potrzeb (VOD – *Voice of the Customer*) – określenie potrzeb klienta i jego preferencji oraz śledzenie ich zmian; prowadzenie monitoringu sytuacji;
- ukierunkowanymi na jakość przekazu – konwertującymi wymagań klienta na język zrozumiały dla projektanta systemu oraz – zwrótnie – projektanta i programisty na język użytkownika i rynku (QFD – *Quality Function Deployment*);
- definiującymi procesy i role w projekcie (diagram SIPOC – skrót od *Supplier, Input-Process-Output, Customer*) – rozpoznanie i długotrwała analiza procesów, miejsca i roli wykonawcy oraz użytkownika w projekcie;
- identyfikującymi ryzyko (FMEA – *Failure Mode Effect Analysis*) – identyfikacja możliwości popełnienia błędu (ryzyka) oraz ich hierarchii i sposobów przeciwdziałania, na tyle na ile jest to możliwe;
- podsumowującymi wyniki (*Balaced Score Card*) – mierzenie efektywności projektu oraz trendów rozwojowych.

W metodyce tej uznaje się swoistą, a sygnalizowaną wcześniej, nadrzędność wymagań klienta i zarządzania nimi nad pozostałymi ograniczeniami „złotego trójkąta” – to klient ostatecznie, a zgodnie ze swymi potrzebami, decyduje o możliwości zmian zakresu, budżetu oraz terminarzu realizacji projektu.

Zadaniem VOC jest rozpoznanie i monitorowanie wymagań klienta odnośnie do projektu. Polega ono na przeprowadzaniu wywiadów, dyskusji i analiz uzyskanych z badań bieżących. Dodatkowym medium analiz staje się Internet z dostępnymi na tej płaszczyźnie narzędziami (formularze, ankiety, newslettery itp.). Identyfikacja potrzeb klienta formalnych i ukrytych oraz jego miejsca i roli w projekcie jest tylko pierwszym etapem działania VOC – następnym staje się określenie sposobów ich zaspokojenia, aby uzyskać odpowiednie efekty końcowe projektu. Dobrze zdefiniowane i wyrażone potrzeby klienta powinny

być powiązane z produktem końcowym projektu i ocenione za pomocą mierzalnych kryteriów.

Zasadniczym celem tworzenia diagramu SIPOC jest określenie miejsca i oczekiwań klienta w realizowanym procesie. Wykorzystywany jest on do przedstawienia przepływu dokumentacji (*workflow*), a jego nazwa pochodzi od pierwszych liter elementów diagramu:

- dostawca (*Supplier*) – osoba dostarczająca zasoby wykorzystywane w procesie,
- wkład (*Input*), wejście – zasoby ograniczające proces,
- proces (*Process*), zespół czynności – algorytm – przetwarzający wkład (wejście), po to by zwiększyć jego wartość,
- efekt końcowy (*Output*), wyjście – produkty pochodzące z procesu,
- klient (*Customer*) – osoba lub proces uzyskująca efekty końcowe.

Diagram ten można stworzyć zarówno dla pojedynczego procesu, jak i dla całej organizacji. Opracowanie takiego diagramu przynosi następujące korzyści ([http://www.4pm.pl/arttykul/six\\_sigma\\_czesc\\_4-26-27-2.html](http://www.4pm.pl/arttykul/six_sigma_czesc_4-26-27-2.html)):

- zrozumienie przez klienta swojej roli oraz terminologii stosowanej w projekcie, w poszczególnych procesach,
- zaangażowanie klienta w projekcie,
- opracowanie metod wzajemnej komunikacji w projekcie,
- identyfikacja niewyspecyfikowanych wcześniej ukrytych potrzeb klienta.

Zbudowanie diagramu SIPOC zależy bezpośrednio od dokładności przeprowadzonych wcześniej analiz dotyczących rozpoznania wymagań klienta. Wyniki tego procesu są bowiem początkiem dalszych działań. A polegają one na zwiększaniu jakości – precyzowaniu stanu pożądanego przez klienta. Problem braku zrozumienia stron projektu, w szczególności programisty/projektanta i użytkownika końcowego był w informatyce niemal od początku istnienia oprogramowania jednym z kluczowych. Natomiast rozbieżności tworzonych na tej konfrontacyjnej płaszczyźnie rozwiązań są zgorą funkcyjnych systemów. Stąd liczne próby sprowadzenia tego porozumienia na płaszczyznę języków graficznych bardziej zrozumiałych dla obu stron. Podobne działania podjęto w QFD – narzędziu metodyki *Six Sigma* i nazwano je „domem” jakości ([http://www.4pm.pl/arttykul/six\\_sigma\\_czesc\\_4-26-27-2.html](http://www.4pm.pl/arttykul/six_sigma_czesc_4-26-27-2.html)). Metoda ta zaczyna się od budowy macierzy przeciwstawiającej i przekładającej wymagania klienta na końcowy produkt informatyczny, w kilku krokach przybliżających je do wymagań projektowania i wykonania systemów informatycznych poprzez przełożenie:

- początkowych wymagań na zwymiarowany produkt, umiejscowiony w rynkowych warunkach konkurencyjności,
- produkt na jego szczegółową specyfikację,

- specyfikacji na logiczny i techniczny projekt procesu,
- projektu procesu na szczegółowe działania określające jego przebieg i ocenę.

W zasadzie jest to sformalizowanie czynności sugerowanych i wykonywanych w wielu innych metodykach zarządzania projektami. Konfrontowanie wymagań klienta z możliwościami zrozumienia go przez wykonawców, jego wizji z możliwościami firmy, w której pracuje – to wszystko odbywało się często w iteracyjnym procesie wielu cykli życia przedstawionych w poprzednim rozdziale. Różnica tkwi w szczegółach.

Tak zwany pierwszy „dom” jakości budowany jest, by idee klienta dotyczące projektu skonfrontować z możliwościami realizacyjnymi firmy. W tym celu buduje się macierz, której kolumny odzwierciedlają możliwości działania firmy projektowej (łatwość obsługi, szybkość działania, dopasowanie do firmy, łatwe uaktualnienia, samowystarczalność, łatwość, zgodność ustaleń, sprawność działania w projekcie, bezkonfliktowość, sprawność działania w projekcie), wiersze zaś pokazują wymagania klienta (znajomość podsystemów, fachowość obsługi, dostępność, terminowość, wrażliwość, innowacyjność, elastyczność, zarządzanie zmianami, przejrzystość itp.). Do poszczególnych wierszy przypisuje się wspólnie z klientem subiektywne wagi o ustalonej skali (przeważnie 1–5) określające jego preferencje, co do możliwości wykonania projektu przez firmę projektową. Następnie graficznie określa się oczekiwane tendencje rozwojowe (kierunki usprawnień) w stosunku do możliwości wykonawcy (wzrastające, malejące (strzałki góra/dół), stałe (okrąg)). Wielu użytkowników zaznacza wyraźnie (na wszelki wypadek) tendencje rosnące. Kolejnym krokiem jest wypełnienie tabeli graficznie przygotowanymi ocenami, wg następującej skali:

- pusty trójkąt (1 punkt), słaba korelacja pomiędzy danym rodzajem oczekiwań a możliwościami jego realizacji,
- pusty okrąg (3 punkty) – średnia korelacja,
- wypełniony okrąg (9 punktów) – pełna korelacja z określoną możliwością realizacji firmy wdrożeniowej.

Jest to etap przygotowawczy – monitoring projektu pokazujący podejście do jego realizacji z punktu widzenia dwóch stron – realizatorów i późniejszych użytkowników. Zaleca się powtarzanie tego działania, prowadzące w iteracyjny sposób do uściślenia mapy rozbieżności oczekiwań w wykonaniu projektu.

Na tej podstawie buduje się tzw. drugi dom jakości. Opierając się na ustalonych ocenach i algorytmach, oblicza się istotność rozpatrywanych parametrów i poddaje kolejnej analizie – analizie porównawczej wykonawcy z firmami konkurencyjnymi. Następnie ponad macierzą zależności buduje się macierz relacji wzajemnych (dach domu) pomiędzy cechami charakterystycznymi – możliwościami wykonawców (stymulacja, kompromis, szkodliwość wzajemna) na pod-

stawie wywiadów z pracownikami lub klientami firmy. Na tej podstawie z kolei wyciąga się wnioski dotyczące działań firmy użytkownika i firmy wykonawcy na poziomie operacyjnym i strategicznym.

Przedstawione narzędzia są metodami gromadzenia, uporządkowania i sprecyzowania danych źródłowych pod kątem optymalnej realizacji tworzonych na ich podstawie założeń projektu, a w rezultacie usprawnienie jakości jego wykonania. W praktyce realizacja kolejnych faz za pomocą wymienionych narzędzi oznacza ciągły, trwający często przez wiele miesięcy proces, składający się z wielu czynności wykonywanych przez współpracujące ze sobą zespoły wykonawcze.

Wśród czynników osiągnięcia sukcesu i wad metodyki *Six Sigma* można wymienić (Modliński, 2005; StatSoft Polska, 2012):

- 1) sukces możliwy jest jedynie w przypadku właściwego procesu realizacji, czyli przestrzegania szczegółowych, detalicznych i bardzo restrykcyjnych zaleceń metodycznych i prawidłowej adaptacji do specyficznych warunków panujących w firmie; ponieważ metoda opiera się na zjawiskach mierzalnych o rozkładzie normalnym, a w gospodarce rzeczywistości taki rozkład rzadko występuje, to wyniki – pomimo najlepszych intencji – jedynie nadążają za wielkością sześć sigma, a raczej nigdy jej nie osiągają; drugim problemem jest tu dążenie do minimalizacji odległości od wielkości planowanych – tańsze jest i chyba częściej spotykane w innych metodykach założenie o minimalnym poziomie zaspokojenia tej zgodności z oczekiwaniami klienta;
- 2) jest to metoda kosztowna – ponieważ występuje konieczność przeszkolenia własnych pracowników w zakresie wdrożenia metody lub skorzystania z profesjonalistów w tym zakresie, ale jednocześnie poniesione nakłady zwracają się najczęściej już po pół roku;
- 3) podobnie jak inne metodyki zarządzania projektami, dla osiągnięcia sukcesu *Six Sigma* potrzebuje: zaangażowania w projekcie najwyższego kierownictwa oraz pracowników – odpowiednio zmotywowanych (zwłaszcza dla zadań wykraczających poza czas pracy lub wymagających podniesienia kwalifikacji), wykorzystania w nim najbardziej utalentowanych pracowników do stworzenia kadry wdrażającej (tzw. mistrzów czarnych pasów (*master black belts*) i noszących czarne pasy (*black belts*)), stworzenia infrastruktury zarządzania dostosowanej do wspierania wdrożenia, ciągłej kontroli zachodzących zmian oraz odpowiedniej polityki informacyjnej – szerzenia wiedzy na temat zasad funkcjonowania *Six Sigma*;
- 4) osiągnięcie poziomu sześć sigma jest szalenie trudne (polskie firmy posiadają procesy na poziomie 2–3), poprzez doskonalenie procesów wystarczy jednak dojść do poziomu cztery sigma, by konkurować na rynkach światowych (to już daje ilość popełnianych błędów w procesie, np. produkcyjnym, na poziomie mniejszym niż 1%),

- 5) metodyka ta jest nie lepsza niż TQM, chociaż olbrzymią rolę odgrywają w niej zaawansowane metody statystyczne wykorzystywane dla poprawy jakości; problem polega jednak jak zwykle na właściwym jej zastosowaniu – jeżeli takie nastąpi, to można powiedzieć, że *Six Sigma* ma krótszy horyzont czasowy dla uzyskania efektów, dzięki lepszemu zdyscyplinowaniu i wyższemu tempu realizacji;
- 6) zakres zastosowania metody jest bardzo szeroki – można ją stosować wszędzie tam, gdzie występują procesy (przedsiębiorstwa produkcyjne, banki, urzędy itp.); przed jej wdrożeniem zawsze należy jednak badać specyfikę danej branży (np. kwestie zachowania bezpieczeństwa) oraz efektywność ekonomiczną jej wdrożenia – relację potencjalnych efektów do poniesionych nakładów;
- 7) metodyka ta może być zastosowana zarówno w całej firmie, jak i w jednym jej dziale lub dla usuwania problemów związanych z jednym procesem; mogą to być procesy związane zarówno ze sferą materialną (np. produkcją) lub z informacyjną (administracja);
- 8) metodyce zarzuca się: brak oryginalności, z powodu oparcia na zdroworozsądkowych założeniach praktycznych; wątpliwe zyski z zastosowania metody (dużo firm rezygnuje z jej stosowania), spadek zadowolenia pracowników (nadmierny formalizm wymusza dokładność działania i nie daje możliwości twórczego potraktowania projektu, co powoduje tłumienie innowacyjności) oraz skupienie się na produkcie, który będzie przynosił zyski.

### 2.5.3. System zarządzania jakością według standardów ISO

Podobnie jak uprzednio omówione metodyki, standardy ISO (*International Organization for Standardization*)<sup>11</sup> mają za zadanie wypracowanie powszechnych, powtarzalnych i jasno zdefiniowanych procesów, których zadaniem jest poprawa jakości realizowanych produktów i usług.

„...Norma jest to dokument przyjęty na zasadzie konsensusu i zatwierdzony przez upoważnioną do tego jednostkę organizacyjną, ustalający – do powszechnego i wielokrotnego użytkowania – zasady, wytyczne lub charakterystyki odnoszące się do różnych rodzajów działalności lub ich wyników i zmierzający do uzyskania optymalnego stopnia uporządkowania w określonym zakresie. Zaleca się, aby normy były oparte na osiągnięciach zarówno nauki, techniki, jak i praktyki oraz miały na celu uzyskanie optymalnych korzyści społecznych...” (*Przewodnik 2 ISO/IEC...*, za: Niemiec i Grocholski, 2005, s. 250).

<sup>11</sup> Wśród których szczególnie ważne dla projektowania systemów informatycznych są te z serii 9000.



Z przytoczonej definicji wynikają główne zadania normy: stworzenie punktu nowoczesnego odniesienia dla realizacji projektów, ale też zagrożenia – norma powinna być ciągle monitorowana, poprawiana i unowocześniana, żeby mogła prawidłowo odgrywać swą rolę i spełniać oczekiwania użytkowników. Norma może zostać odniesiona do wielu działań realizowanych przez organizację od projektowania, po procesy produkcyjne czy też dystrybucję. Przyporządkowane omawianym normom zarządzanie jakością bezpośrednio odnosi się do zarządzania procesami. Procesy i wszelkie czynności składające się na nie muszą być szczegółowo analizowane, co w rezultacie ma doprowadzić do wykrycia potencjalnie w nich występujących błędów i ich eliminacji bądź przynajmniej minimalizacji. Norma służy jako pewien punkt odniesienia, niespełnienie jej oznacza jednocześnie niespełnienie wymogów jakości.

Normy mają różny charakter, związane są z analizą, projektowaniem i wdrażaniem systemów informatycznych, np. ISO 9001:2000 jest normą organizacyjną, a nie techniczną. Ze względu na uniwersalny charakter musi być dostosowana do konkretnego obiektu – uwzględniać bieżące i przyszłe potrzeby klienta, propagować sposoby utrzymania zaangażowania zespołów zadaniowych, propagować podejście procesowe, dynamizować proces usprawniania organizacji, sformalizować proces podejmowania decyzji – oparcie ich na bazie faktograficznej oraz uwzględniać prawidłowe relacje z kontrahentami. Spełnienie norm jest stwierdzane w procesie certyfikacji. Poprzedzony on jest:

- zaprojektowaniem systemu zarządzania jakością i jego udokumentowaniem,
- wdrożeniem systemu zarządzania jakością, z możliwością oceny jego skuteczności,
- przeprowadzeniem certyfikacji (audyt teoretycznych założeń systemu zarządzania jakością i jego dokumentacji, audyt zgodności założeń i dokumentacji z faktycznym działaniem wdrożonego systemu zapewniania jakości).

Metodyka usprawniania działań jest więc zawarta w normie lub grupie norm i ma na celu nie tylko podniesienie jakości wyrobu bądź usługi, lecz także samych procesów zarządzania organizacją w trakcie jej uzyskania.

Główne zalety korzystania z systemu normalizacji mogą być więc sformułowane w następujący sposób:

- rozpoznawalność i uznanie w świecie,
- przedstawienie w postaci wymagań i zaleceń całości prac od wymagań użytkownika począwszy, poprzez projektowanie, wykonanie, wdrażanie, eksploatację i ciągle doskonalenie,
- rozdziela prace systemowe od merytorycznych,
- wprowadza formalizację prac oraz procesów (struktury organizacyjnej, obowiązków, kompetencji, procesów podstawowych, procesów pomocniczych),
- zaleca uporządkowanie infrastruktury związanej z projektem,

- wprowadza konieczność oceny okresowej satysfakcji klientów, systemu zarządzania jakością, poziomu i kwalifikacji kontrahentów itp.,
- wprowadza możliwość odtworzenia w dowolnym czasie zdarzeń uprzednich, wpływających na jakość wytwarzanego produktu lub usługi.

Do podstawowych ograniczeń stosowania standardów ISO należy zaliczyć m.in.:

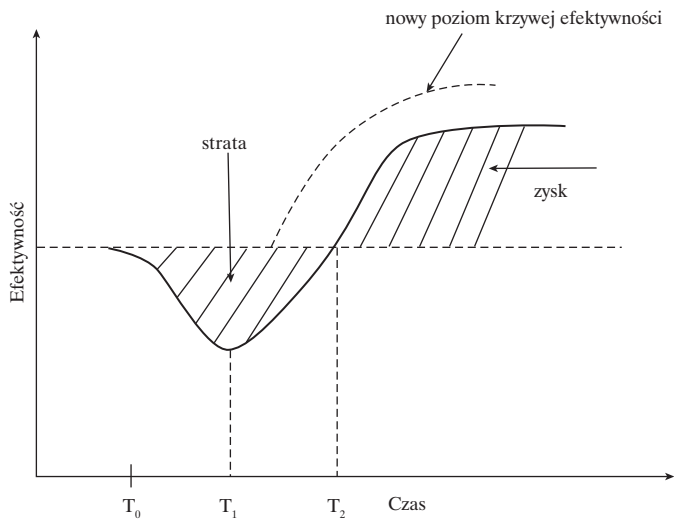
- zbytnią formalizację postępowania, co zwiększa koszty funkcjonowania organizacji;
- wysokie koszty zaprojektowania, udokumentowania, wdrożenia i certyfikacji systemów zapewnienia jakości;
- wydłużenie czasu wprowadzania nowych członków do zespołów zadaniowych (muszą się uczyć systemu zarządzania jakością opartego na normie).

Wydaje się jednak, że potencjalne korzyści z wprowadzenia systemów normalizacji przeważają nad potencjalnymi niedostatkami, dlatego rekomenduje się metodykę normalizacji do zbioru metod wspomagających zarządzanie projektami.

\*

Komercyjne, zintegrowane metodyki zarządzania projektami, korzystające z wielu metod i technik umożliwiają bardzo często spójne zbudowanie, a następnie wdrożenie nowoczesnych systemów informatycznych wspomagających zarządzanie, zwłaszcza typu *Business Information Systems*, opartych nie tylko na hurtowniach danych (wraz z hurtowniami branżowymi), lecz także połączone z nimi pakiety *Business Analytics*, zawierające mechanizmy i wzorce zarządzania wiedzą. Zwłaszcza, że zastosowanie nowej technologii informacyjnej w organizacji nie od razu zaczyna przynosić widoczne korzyści. Na początku – czyli po wdrożeniu, w momencie rozpoczęcia eksploatacji systemu, efektywność wykonywania zadań przez stosujących ją pracowników spada, w stosunku do poziomu poprzedniego, co jest związane z adaptacją nowego rozwiązania przez użytkowników. To powoduje – zamiast spodziewanych zysków – początkową stratę. Wraz z postępem procesu uczenia się klientów nowego systemu efektywność ich pracy zaczyna stopniowo rosnać. Najważniejsze jest, żeby moment osiągnięcia poprzedniej wydajności pracy nastąpił jak najszybciej, aby minimalizować straty dla przedsiębiorstwa, a nowa technologia zaczęła generować korzyści. Sytuację tę ilustruje rysunek 17. Zyski pojawiają się, gdy pole pod krzywą do poziomu prostej wyznaczającej poprzedni średni poziom efektywności przewyższy pole nad krzywą wyznaczającą straty. Pojawienie się kolejnej innowacji powoduje przesunięcie krzywej efektywności na nowy, wyższy poziom, co oznacza, że przyrost efektywności spowodowanej wdrożeniem kolejnych systemów może nie mieć końca.

Sytuacja ta wymusza na użytkowniku przyjęcie takich procedur zarządzania projektem, które zapewniają jak najmniejszy spadek efektywności po wdrożeniu, jak najszybszy punkt osiągnięcia poprzedniego poziomu i jak najwyższy kąt nachylenia krzywej po przekroczeniu tego punktu (ponieważ przynosi to szybsze generowanie skumulowanego zysku). Osiągnięcie tych warunków przyspiesza użycie zintegrowanych metod zarządzania projektami.



**Rysunek 17.** Skutki wdrożenia systemu informatycznego wspomagającego zarządzanie, generacja strat i zysków w okresie powdrożeniowym

Źródło: opracowanie własne.

# 3

## Tendencje rozwojowe systemów informatycznych zarządzania

---

W poprzednich rozdziałach przedstawiono kolejno: definicje, założenia, ogólne metodyki zarządzania projektami informatycznymi oraz metodyki komercyjne dostarczane przez firmy konsultingowe do rozwiązywania problemów zarządzania projektami. Zwrócono też uwagę na:

- zmiany, które systematycznie zachodzą w obszarze zarządzania projektami, rozszerzanie tego pojęcia, wraz z cyklicznymi zmianami zakresu od formalizacji do intuicyjności;
- upodobniania procedur unowocześniania metodyk, pomimo zróżnicowanego nazewnictwa, terminologii i opisu graficznego;
- wad i zalet związanych ze stosowaniem każdej z nich.

Obecnie wydaje się, że nadszedł czas na pokazanie cech charakterystycznych zróżnicowanych produktów, które powstają w wyniku procesu projektowego lub projektowo-wdrożeniowego. Tym produktem są systemy informatyczne wspomagające zarządzanie. Szeroka gama stosowanych systemów, implikacje historyczne ich nieustającego rozwoju i zależność od poziomu bieżącej innowacyjności powodują pewien bałagan klasyfikacyjny potęgowany zjawiskami przenikania się sfery tradycyjnej gospodarki z tzw. sferą elektroniczną. Szeroki wybór wciąż niedoskonałych metodyk zarządzania projektami informatycznymi, rosnący stopień ich komplikacji i wysoka dynamika zmian również nie sprzyjają jasności i zrozumieniu problemów powstających w tym zakresie. Stąd zamieszczony w niniejszym rozdziale pomysł uporządkowania tej sfery, który może przyczynić się także do uzmysłowienia projektantom implikacji ich działań w obszarze technologii informacyjnych. Uporządkowanie poddano weryfikacji – poprzez „umocowanie” poszczególnych analizowanych i charakteryzowanych rodzajów systemów w kontekście uwarunkowań ich rozwoju historycznego, determinowanego wzrostem wynalazczości i innowacyjności nie tylko w sferze technicznej, lecz także organizacyjnej, ekonomicznej i prawnej.

### 3.1. Przegląd literaturowy podejścia do tendencji rozwojowych systemów informatycznych

Systematyką rozwoju systemów informatycznych na potrzeby zarządzania organizacją zajmowano się w zasadzie od momentu powstania pierwszego komputera. Najczęściej przeplatała się z metodyką projektowania i tworzenia systemów informatycznych. Zwłaszcza, że obie te sfery były przecież i są nadal bardzo od siebie uzależnione. Tematyka ta – w sensie klasyfikacji, typologii, uproszeń i standaryzacji niezbędnych dla uporządkowania tej sfery stanowiła wielokrotnie podmiot i przedmiot rozważań wielu autorów, również na polskim rynku wydawniczym. I – jak się wydaje – cały czas wносиła do tej sfery coś nowego. Tradycja pisania na ten temat została zapoczątkowana w środowisku wrocławskim przez Z. Hellwiga (1971; 1975) książką pod jego redakcją *Automatyczne przetwarzanie informacji* i późniejszymi wydaniem książki *Maszyny cyfrowe i ich zastosowanie*. Wtórne niejako w stosunku do tego pomysłu czy też po prostu aktualniejsze w stosunku do bardzo szybko zmieniającej się rzeczywistości, czy rozszerzające tę tematykę o systemy i ich zastosowania, książki rzadko osiągały podobną klasę (chlubne wyjątki oczywiście się zdarzały: dość wymienić niektóre prace autorskie i pisane pod redakcją: T. Wierzbickiego, W. Radzikowskiego, W. Flakiewicza czy J. Kisielnickiego). Chociaż środowisko informatyków gospodarczych stale się powiększało (AE Katowice, Politechnika, potem Uniwersytet Szczeciński, AE w Poznaniu, Uniwersytet Gdański, SGH (wtedy SGPiS) i Uniwersytet Warszawski), to publikacje na ten temat tworzone były głównie pod aktualne potrzeby istniejących programów nauczania, rzadko wyznaczając ich standardy oraz w oparciu o aktualnie posiadane umiejętności i materiały osób, które je kreowały. Rola minimów programowych wyznaczanych przez urzędników ministerialnych była w tej mierze i jest głęboko kontrowersyjna. Płynące z nieznamości najnowszych osiągnięć w danej dziedzinie, czasem głęboko zachowawcze (niezmieniane przez wiele lat), czasem sprowadzające się do technologii, niespójne, a nawet nielogiczne zalecenia w tym zakresie z jednej strony konserwują społeczne odczucie o prymacie technicyzmu informatyki (czyli jak i czym, a nie dlaczego), z drugiej zaś – potrafią do dnia dzisiejszego wprowadzić chaos do, wydawałoby się uporządkowanej już w miarę, materii (np. dodając *Technologie informacyjne* jako odrębny przedmiot obowiązkowy na wielu kierunkach studiów). W tej skomplikowanej sytuacji środowisko akademickie radziło sobie bardzo dobrze – zob. np. cykl wydanych w latach: 1998–2003 książek pt. *Informatyka ekonomiczna* pod red. E. Niedzielskiej (1998), A. Nowickiego (2005; 2006) *Wstęp do systemów informacyjnych zarządzania w przedsiębiorstwie czy Komputerowe wspomaganie biznesu*. Wydania te konkurowały na rynku z takimi potentatami wydawniczymi, jak książka J. Kisielnickiego i H. Sroki pt. *Systemy informacyjne biznesu* (2005) czy wydana pod redakcją C. Olszak i H. Sroki

*Informatyka w zarządzaniu* (2003). Miały też szansę konkurować z książką pod red. C. Olszak i E. Ziemby pt. *Systemy gospodarki elektronicznej w erze informacji i wiedzy* (2007) i prawdopodobnie z książką K. Krupy pt. *Teoria zmian organizacyjnych przedsiębiorstw ery informacji* (2006). Natomiast w ostatnich latach na szczególną uwagę zasługują prace A. Januszewskiego pt. *Funkcjonalność informatycznych systemów zarządzania* (2008), J. Kisielnickiego pt. *MIS – Systemy informatyczne zarządzania* (2008) czy *Technologie informacyjne dla ekonomistów* pod redakcją A. Nowickiego, T. Turka (2010). Ostatnie prace były inspirowane najnowszymi dokonaniem, opublikowanymi w istniejących już od wielu lat na rynkach anglojęzycznych seriach wydawniczych, takich autorów, jak E. Turban z zespołem (2007; 2008), czy kolejne książki K. Laudon i J. Laudon (2010).

Niemal w każdej z przytoczonych wybitnych pozycji literaturowych możemy się spotkać z inną genealogią, klasyfikacją bądź typologią rozwoju systemów informatycznych wspomagających zarządzanie. Czasami wymieniane są systemy informatyczne w dowolnej postaci i porządku oraz szczegółowo charakteryzowane. Obecnie panuje kompletny chaos terminologiczny, pogłębiony zwyczajem powrotu w informatyce do terminów z przeszłości i nadawania im nowych nowoczesnych zgoła, a całkiem różnych znaczeń (np. systemy transakcyjne). Jeszcze inni autorzy w kolejnych wydaniach swoich książek przyjmują coraz to bardziej skomplikowany punkt widzenia na kwestie klasyfikacyjne, czasem niemal całkowicie różny niż w pozycjach poprzednich.

Dlatego też w niniejszym rozdziale podjęto próbę uporządkowania tej sfery z jednej strony opartą na tendencjach integracyjnych, z drugiej zaś – tendencjach konwergencyjnych (upodobniania się i przenikania implementacji) umożliwione ciągłym postępem technologicznym. Wyróżniono trzy rodzaje rozwoju systemów informatycznych opartych na różnych przesłankach koncepcyjnych, łączone obecnie na teoretycznych podstawach platform korporacyjnych.

Integracja – w sensie ideowym – polega na takim połączeniu elementów funkcjonalnych za pomocą relacji, by stanowiły składowe określonej strukturalnie całości. Integracja jest tu rozumiana jako proces scalania i zespalania się poszczególnych, różnej klasy postaci i form powiązanych wzajemnie elementów w celu tworzenia funkcjonalnej całości, o użyteczności i/lub efektywności większej niż posiadałaby każda z tych części działająca oddzielnie (efekt synergii). Konwergencja – to kształtowanie się w ewolucyjnym procesie rozwojowym podobnych cech budowy, funkcji i wyglądu zewnętrznego różnych grup systemów funkcjonujących w takich samych warunkach środowiskowych, niezależnie od przyjętych szczegółowych rozwiązań innowacyjnych oraz przenikanie i kojarzenie zjawisk znajdujących się na ich pograniczu. System informatyczny<sup>1</sup> traktowa-

<sup>1</sup> Niektórzy z przywołanych uprzednio autorów traktowali system informatyczny w zbliżony lub tożsamy sposób.

ny jest tu jako uporządkowany zbiór programów (modułów, podprogramów) odzwierciedlających w najbardziej użyteczny sposób funkcjonalne wymagania użytkownika za pomocą zapewniającej ten proces odpowiedniej infrastruktury technologicznej umiejscowionej zarówno w organizacji, jak i w jej otoczeniu. Zakłada się, że zarówno oprogramowanie, jak i technologie informatyczne podlegające opisywanym powyżej zjawiskom są wynikiem historycznego rozwoju, kształtującego się co najmniej na trzech ścieżkach rozwojowych:

- komplikacji architektury logicznej systemów,
- integracji funkcjonalnej systemów informatycznych, dostosowanej do aktualnych potrzeb organizacji i działającego w niej użytkownika,
- infrastrukturalnych rozszerzeń sieciowych (przestrzennych) ich działalności.

Wymienione ścieżki rozwojowe kształtowały się równolegle, ale nie obok siebie. Często się przeplatały i zawsze korzystały wzajemnie z własnych doświadczeń i narzędzi. Niekiedy w ramach danej ścieżki rozwojowej następowało sprzężenie zwrotne – powrót do przeszłości – w celu urzeczywistnienia i rozwinięcia koncepcji, wcześniej wymyślonej, a niemożliwej uprzednio do zrealizowania z powodu niedostatecznego rozwoju innowacji technologicznych.

Część teoretyków informatyki ekonomicznej (Silver, 1991) wiąże początek powstania systemów informacyjnych wspomagających zarządzanie z pojawieniem się pierwszego komputera sprzedanego w sposób komercyjny w 1951 r. do Biura Ewidencji Ludności w USA. Od tego czasu coraz więcej firm adaptowało oparte na komputerach systemy przetwarzania transakcyjnego (*Transactional Processing Data Systems – TPS lub inaczej Automation Data Processing – ADP – systemy automatyzacji przetwarzania danych*), dotyczące pojedynczych, prostych systemów ewidencyjnych typu: listy płac, wydruków rachunków, zestawień statystycznych. Nie wydaje się jednak, że fakt ten stanowił o ich późniejszej popularności. Nadal były to drogie, numeryczne, separowane systemy informacyjne przywiązane do klasy i firmy komputera, na który zostały stworzone, nieprzekładalne na inny typ sprzętu, a przez to bardzo drogie. I dodatkowo dla użytkownika – trudne do nauczenia siebie i innych. Rozwój techniki, który spowodował dekadę później wzrost mocy obliczeniowej komputerów, pociągnął za sobą również rozwój oprogramowania oraz nowych technik gromadzenia danych uzyskanych na wyjściu systemów transakcyjnych. A ponadto doprowadził w połowie lat sześćdziesiątych, na bazie zjawisk standaryzacji, do powstania systemów operacyjnych i języków programowania, takich, które można było wreszcie wykorzystać do tworzenia oprogramowania możliwego do zastosowania w wielu konfiguracjach komputerowych. Mniej lub więcej powielarnego, jak z obecnego punktu widzenia moglibyśmy zauważyć. Stwierdzono też, że zgromadzone w dotychczasowych systemach transakcyjnych informacje powinny być dalej wykorzystywane w procesie planowania i kontroli. Ta obserwacja spowodowała powstanie nowej klasy

oprogramowania użytkowego. Oprogramowanie to obejmowało głównie takie dziedziny działalności firmy, jak fakturowanie, rejestrację opłat i zakupów, ewidencję stanów i obrotów magazynowych, inwentaryzację, rachunkowość itp. Dla menedżerów wyniki działania takich systemów przedstawiały się jako wstępnie zdefiniowane, periodyczne, zestandaryzowane raporty otrzymywane w reżimie tygodniowym lub miesięcznym, będące wynikiem przetwarzania transakcyjnego. Systemy te nazwano systemami informacyjnymi zarządzania (*Management Information Systems – MIS*), a ich wykorzystanie dla celów zarządzania zaczęło opierać z czasem na bazach danych. Od nich właśnie rozpoczyna się nowoczesna era systemów informatycznych wspomagających zarządzanie. Nie był to rozwój ani prosty, ani łagodnie przechodzący w następne fazy. Problemem tym – jak napisano na wstępie – zajmowało się wielu teoretyków i praktyków tej dziedziny. Prezentacje ich dokonań wymienione na początku tej części opracowania do dziś budzą czasem niejako kontrowersje. Niniejszy rozdział jest więc też próbą włączenia się w tę dyskusję. Jego zasadniczym celem zaś jest prezentacja własnego podejścia, autorskiego do kierunków rozwoju systemów informatycznych wspomagających procesy zarządzania skupionego wokół wymienionych uprzednio trzech ścieżek rozwojowych, na tle tendencji integracyjnych i konwergencyjnych.

### **3.2. Rozwój systemów informatycznych drogą komplikacji architektury logicznej systemów informatycznych**

Pierwsza ścieżka rozwojowa – komplikacji architektury logicznej systemów informatycznych – wydaje się być najlepiej rozpoznana w literaturze przedmiotu. Zajmowało się nią najwięcej badaczy już od początków lat osiemdziesiątych ubiegłego stulecia, a rozwój jej traktowano jako bezpośrednią implikację postępu technologicznego. A był to rzeczywiście czas głębokich przemian i dynamicznego rozwoju technologicznego systemów informatycznych. Tak spektakularnego, że w końcu zaczął przesłaniać największe dokonania dla rozwoju zarządzania w tym zakresie, czyli tworzenie coraz bardziej wyrafinowanych systemów informatycznych, odzwierciedlających faktyczne potrzeby i wymagania użytkownika końcowego tych systemów. Poniżej przypomniano główne, a kolejne etapy rozwoju systemów informatycznych na tej drodze (Chmielarz, 2005).

#### **3.2.1. Transakcyjne systemy przetwarzania (*Transactional Processing Data Systems – TSP – od 1951 r.*)**

Dla porządku słów parę należy poświęcić systemom transakcyjnym (TSP), w ich ówczesnej postaci. Były to niewątpliwie pierwsze próby stworzenia narzędzia, które pośrednio mogłyby służyć wspomaganie prowadzenia bizne-



su. Podstawową zaletą takiego narzędzia była szybkość w wykonywaniu prostych, standardowych operacji masowych. Głównym problemem był wówczas niski rozwój technologiczny, który sprawiał, że zanim przetworzenie danych mogło dojść do skutku było obudowane szeregiem skomplikowanych ręcznych czynności i procedur administracyjno-organizacyjnych związanych z niedoskonałością istniejącego sprzętu i oprogramowania. Dodatkowo towarzyszyły temu procesowi, jak wspomniano uprzednio, znaczące koszty oraz zawodność i awaryjność sięgająca raptem do 80% czasu pracy takiej maszyny. Najbardziej skomplikowane było napisanie i przetworzenie programu, który operował na wprowadzanych do pamięci operacyjnej komputera danych. Ograniczeniem było właściwie wszystko: kłopoty z wprowadzeniem programu i danych, które miał przetwarzać, przetworzenie danych i zapamiętanie wyników, dystrybucja tych wyników pomiędzy zainteresowane jednostki (dochodziły do tego takie niedogodności, jak: szybkość przetwarzania, objętość pamięci, problemy z projektowaniem i budową oprogramowania itp.). Każdy z budowanych systemów był odrębny, co niekiedy prowadziło do wprowadzania tych samych danych w ramach organizacji w sposób wielokrotny, a jeszcze dodatkowo – często w różnych formatach, na różnych maszynach i przez różne zespoły programistyczne. Używanie zawodnych mediów wejściowych, o długotrwałej obróbce, i odseparowanie użytkownika od przetwarzania na komputerze danych, które mógł co najwyżej samodzielnie przygotować, dopełniło reszty. Systemy te sprawdzały się więc w masowych obliczeniach numerycznych (typu spis powszechny), interpretacji których dokonywano już w sposób „ręczny”. Ich przydatność do wspomagania zarządzania sprowadzała się głównie do przyspieszenia obliczeń numerycznych.

### 3.2.2. Systemy informacyjne zarządzania (*Management Information Systems – MIS – od 1964 r.*)

Systemy informacyjne zarządzania (MIS) od samego początku swojego powstania były desygnowane do prowadzenia rejestracji przeszłych i bieżących rutynowych informacji przeznaczonych dla planowania, organizowania oraz kontrolowania operacji w funkcjonalnych zakresach działania organizacji (Turban, Leidner, McLean i Wetherbe, 2008). Systemy informacyjne zarządzania są definiowane przez R.M. Staira (1992, s. 47) jako „...zorganizowany zbiór ludzi, procedur przetwarzania, baz danych oraz urządzeń używanych do dostarczania standardowej informacji w różnych przekrojach dla menedżerów i decydentów...”. Według E. Turbana (2008) „...system informacyjny zarządzania jest formalnym, komputerowym systemem, stworzonym dla rejestracji, selekcjonowania i integracji dostarczanej z różnych źródeł informacji w celu zapewnienia aktualnych danych niezbędnych dla podejmowania decyzji w zarządzaniu.

Najbardziej efektywne są one w systemach rutynowych, ustrukturalizowanych, w których podejmuje się przewidywalne typy decyzji...”. Systemy te wywarły – jak dotąd – bodajże największy wpływ na kształtowanie się systemów informatycznych wspomagających zarządzanie przez całe dekady od swojego powstania. Aby mogły to uczynić, przeszły przez cały cykl dogłębnych przemian. Wszak w połowie lat sześćdziesiątych, gdy powstawały, nikt nawet początkowo nie wiedział, jak takie systemy tworzyć<sup>2</sup>. Uniwersalne języki programowania dopiero powstawały (przełomowy był rok 1964 – wprowadzono wówczas Algol, Firtra, następnie Cobol i Basic), sprzęt – mimo szybkiego postępu – nadal tkwił w epoce poprzedniej lub był tak nowy i jakościowo inny, że należało się od nowa nauczyć posługiwania nim. Należało też znaleźć w strukturze przedsiębiorstwa miejsce i określić rolę jednostek organizacyjnych, w których dokonywano przetworzenia systemów. Głębokie przemiany początkowo tworzonych systemów informacyjnych zarządzania powodowane były, jak się powszechnie uważa, przede wszystkim zamianą systemu wsadowego na system bezpośredniego dostępu do komputera, ale chyba równie istotne były zmiany dotyczące metodyki i technologii gromadzenia, przechowywania i dostępu do danych. Nie tylko dla tworzenia pierwszych systemów dziedzinowych (jednodziedzinowych) obsługiwanych przez odrębną od innych systemów dziedzinowych bazę danych<sup>3</sup>, lecz także dla systemów wielodziedzinowych, które już w oparciu o wspólną bazę danych najczęściej są kojarzone z systemami informacyjnymi zarządzania. Najistotniejszym elementem w kształtowaniu tych, nadal najbardziej popularnych, systemów informatycznych była więc baza danych, zwłaszcza relacyjna baza danych z całym aparatem ją obsługującym, z czasem w podstawowej swej postaci zwanym systemem zarządzania bazą danych. Podstawowa struktura architektury logicznej składała się z:

- użytkownika końcowego z interfejsem, czyli zbiorem programów, przeważnie systemu operacyjnego, odpowiedzialnych za komunikację z użytkownikiem, który narzuca pewne standardy postrzegania i wykorzystania pozostałego oprogramowania;
- bazy danych wraz z systemem zarządzania bazą danych – zbiór danych zapisanych zgodnie z określonymi regułami porządkującymi, powiązanych określonymi zależnościami, zapisanych w ściśle zdefiniowany sposób w strukturach odpowiadających zakładanemu modelowi danych; oprogramowanie ułatwiające definiowanie, konstruowanie, manipulowanie i udostępnianie baz danych dla aplikacji i użytkowników nazywane jest systemem zarządzania bazą danych; dodatkowym elementem tego oprogramowania staje

<sup>2</sup> Zob. rozdział 1, przedstawiający m.in. brak metodyki projektowania systemów informatycznych w tym okresie.

<sup>3</sup> Każdy z systemów dziedzinowych był wzorem TSP obsługiwany przez własną bazę danych – przyp. autora.

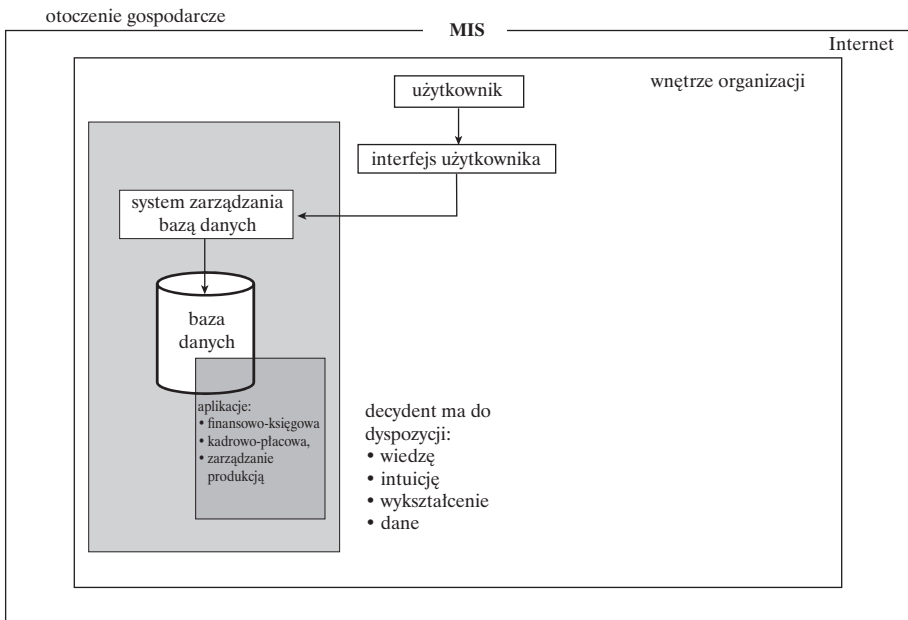
się w późniejszym okresie język zapytań ułatwiający komunikację z bazą danych użytkownikowi (w swej idei nawet niewykwalifikowanemu) w sensie przyjęcia zapytania, jego formalizacji i udostępnienia wyniku decydentowi;

- aplikacje – np. podsystemy, oprogramowanie użytkowe składające się ze zbioru instrukcji, których zadaniem jest dostarczenie użytkownikowi określonej funkcjonalności (podsystem finansowo-księgowy, podsystem magazynowy, podsystem kadrowo-płacowy, podsystem sterowania produkcją itd.).

Ta prosta konstrukcja architektury logicznej znalazła swoje zastosowanie w dziesiątkach tysięcy działających do dziś na rynku systemów, jak również stała się podstawą pod budowę bardziej skomplikowanych systemów, zarówno od strony jej rozbudowy o dodatkowe elementy, jak i obsługi szeregu nowych funkcjonalności. Użytkownik podejmujący decyzje – menedżer – podczas wspomagania swojej działalności tego typu systemami ma do dyspozycji swoją wiedzę zawodową, kwalifikacje, umiejętności, intuicję działacza gospodarczego oraz dostęp do zgromadzonych, ustrukturyzowanych, pewnych danych zarejestrowanych z dokumentów pojawiających się w trakcie działalności gospodarczej organizacji. Sposób dotarcia, posługiwania się i dystrybucji informacji wynikowej jest jeszcze dość prosty, ale do jej wykorzystania potrzebna jest wiedza informatyczna:

- informacja wykorzystywana w procesie decyzyjnym uzyskiwana jest w postaci raportów (dokumentów przetworzonych);
- sposób prezentacji i głębokość jej selekcji i przetworzenia wstępnego zależy od możliwości języka programowania oraz systemu zarządzania bazą danych;
- dla uzyskania przez decydenta informacji w określonym przekroju i o określonym zakresie niezbędna jest co najmniej podstawowa wiedza o strukturze bazy danych;
- występuje względna nadmiarowość informacji bazowej, którą możemy pozyskać z bazy danych w stosunku do możliwości podjęcia na jej podstawie decyzji zarządczej;
- brak jest bezpośrednich mechanizmów przetworzenia informacji pobranych z bazy danych we wzorce mogące stanowić podstawę podjęcia decyzji.

Strukturę logiczną systemu informacyjnego zarządzania przedstawia rysunek 18.



**Rysunek 18.** Struktura logiczna systemu informacyjnego zarządzania

Źródło: opracowanie własne.

### 3.2.3. Systemy wspomagające podejmowanie decyzji (*Decision Support Systems – DSS – od 1978 r.*)

Podstawowa definicja systemów wspomagających podejmowanie decyzji (Silver, 1991, s. 13–23) określa je jako „...oparte o infrastrukturę komputerowo-komunikacyjną systemy informacyjne wspomagające działalność ludzi zaangażowanych w procesie podejmowania decyzji...” (Turban i in., 2008, s. 245). Jako wspomaganie rozumie się pomoc decydentowi w wypracowaniu podejmowanej decyzji, a nie podejmowanie tej decyzji za niego i zastępowanie go w procesie decyzyjnym. Najistotniejsza różnica w stosunku do systemów informacyjnych zarządzania zawiera się w fakcie, że systemy klasy DSS dają menedżerowi oprócz intuicji, wiedzy, umiejętności i informacji, narzędzia do wypracowywania decyzji. Narzędzia przybierają na ogół postać programów (pakietów programowych) składających się z modeli matematycznych, statystycznych, ekonometrycznych lub ich kombinacji, specjalizowanych na zagadnienia związane z zarządzaniem przedsiębiorstwem. Oznacza to, że oprócz deterministycznych warunków, w których podejmowano decyzje na podstawie sprawdzonych danych źródłowych

z bazy danych lub ich selekcji/kombinacji, za pomocą tych systemów możliwe jest podejmowanie decyzji w sytuacjach probabilistycznych, charakteryzujących się posiadaniem przez decydenta niepełnych, wyrwykowych, a czasem nawet częściowo błędnych lub sprzecznych ze sobą danych.

Do struktury wykorzystywanej przez systemy informacyjne zarządzania dodawane są tu nowe elementy:

- baza modeli – zawierająca rutynowe, standardowe oraz wyspecjalizowane modele służące podejmowaniu decyzji w przedsiębiorstwie; z matematycznego punktu widzenia mogą to być modele oparte na zależnościach liniowych bądź nieliniowych, symulacyjne, optymalizacyjne, oparte na teorii gier lub wynikające z dobrych praktyk zarządzania; z organizacyjnego punktu widzenia dzielące się często na wbudowane (z góry narzucone użytkownikowi) oraz konstruowane przez użytkownika, z gotowych bloków (systemów) i procedur (reguł) lub w sposób dowolny ograniczony jedynie nomenklaturą narzędzi wspomagających ten proces;
- system zarządzania bazą modeli – oprogramowania zawierające wszelkie narzędzia niezbędne do obsługi lub manipulowania gotowymi modelami, utrzymania bazy modeli oraz ich wersji modyfikowanych, tworzenia nowych modeli z gotowych elementów lub konstruowania ich na określonych możliwościach systemu zasadach, integracji modeli w jedną całość lub nowych modeli ze starymi, mechanizmów koordynujących zapotrzebowanie na dane przetwarzanych modeli z możliwością bazy danych systemu oraz parametrów i danych zewnętrznych w stosunku do zawartych w bazie danych, koordynowania i współdziałania z urządzeniami dodatkowo rozszerzającymi możliwości wykorzystania bazy modeli; komunikacja z bazą modeli odbywa się za pomocą interakcyjnego języka bazy modeli (uszczegóławiającego zapytania i żądania użytkownika finalnego), a modele utrzymywane są w odpowiednich katalogach;
- baza procedur (*solver*) – program lub pakiet programów służących rozwiązywaniu szczególnie skomplikowanych zagadnień matematycznych, wynikających z konstruowanych modeli (programowania liniowego, nieliniowego, wielokryterialnego, systemów rozmytych itp.); sposób rozwiązania zagadnienia jest albo na stałe przypisany do przetestowanego, standardowo używanego modelu, albo – po ewentualnych konsultacjach (na ogół jest tu możliwość wyboru) z użytkownikiem – dopasowywany do tworzonego od nowa modelu za pomocą mechanizmów systemu zarządzania bazą modeli;
- baza danych i parametrów modeli – podręczna baza danych, która może zawierać dane niezbędne do uruchomienia i użytkowania modelu niepochozące z historycznych i aktualnych danych rejestrowanych w bazie danych (dane modelowe: parametry i współczynniki), dane zewnętrzne pobrane i wprowadzone „ręcznie” z otoczenia gospodarczego (niekiedy wraz z kon-

werterem na format danych używanych w modelu), standardy normalizacyjne itp.).

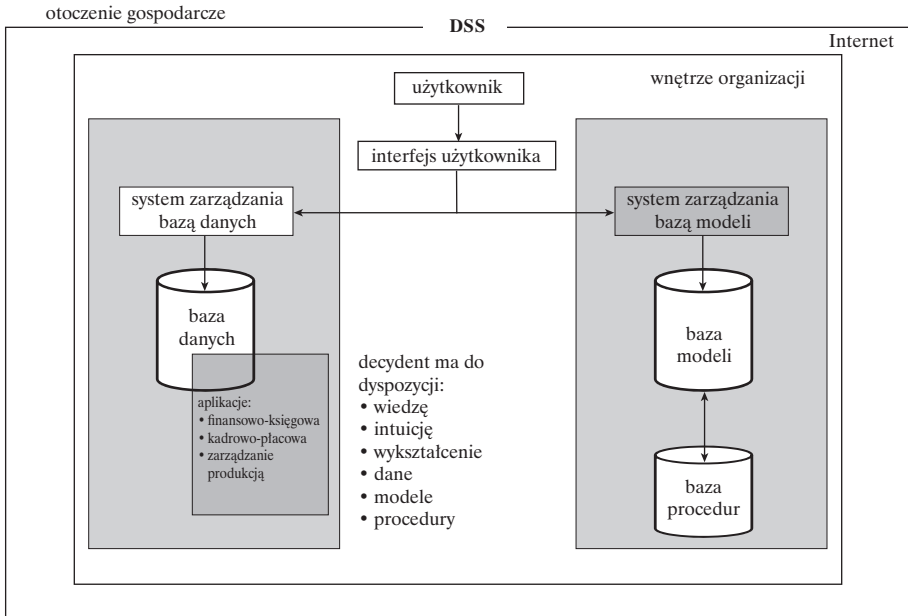
Jak wspomniano, taka architektura po raz pierwszy umożliwiła wypracowanie – na podstawie modelowo sformułowanego procesu podejmowania decyzji – podpowiedzi decyzji najlepszej z punktu widzenia przyjętego kryterium lub zbioru możliwych decyzji do wyboru przez użytkownika. I mniej istotne jest tu epatowanie osiągniętymi na początku tego okresu rozwiązaniami technologicznymi, a istotniejsze jest stworzenie alternatywy dla decydenta – decyzja wypracowana na podstawie dostępnych (ewentualnie wyselekcjonowanych) danych versus decyzja, którą podpowiada komputer na podstawie przyjętego modelowego rozwiązania problemu. Istotne są tu też pojawiające się możliwości badania skutków podejmowania różnych decyzji, jak również projekcji (prognozowanie) w przyszłość czy w układzie przestrzennym.

W rozwoju bazy modeli i systemu zarządzania nią wyróżnia się co najmniej cztery etapy:

- umieszczanie w systemie pojedynczych modeli matematycznych, do których w razie potrzeby można się odwołać jako alternatywy dla podejmowania decyzji intuicyjnej lub opartej na danych z bazy danych;
- dołączanie do struktury systemu gotowych modeli matematycznych opisujących wyróżnione wydarzenia gospodarcze do wyboru przez użytkownika;
- włączanie w strukturę systemu pakietów modeli matematycznych, wraz z odpowiadającymi im procedurami, metodami ich rozwiązania oraz parametrami niezbędnymi do ich uruchomienia;
- umieszczenie w systemie zarządzania bazą modeli mechanizmów pozwalających na stworzenie własnego modelu z bloków i komponentów modeli, integracji ich w całość i automatyczne generowanie podczepianie zbioru danych niezbędnych do jego rozwiązania.

Oprócz niewątpliwie większych możliwości wykorzystania systemów wspomaganie decyzji pojawia się jednak jeden zasadniczy problem, i to mimo znacznie wyższych kosztów tworzenia takich systemów. Jest nim znaczące zwiększenie trudności w posługiwaniu się narzędziami zawartymi w systemie. Do trudności zawartych w efektywnym posługiwaniu się bazą modeli dochodzą więc kłopoty związane z umiejętnościami posługiwania się matematycznymi modelami gospodarczymi i – w razie tworzenia modelu – problemy z budową modelu jako prawidłowego (czy przynajmniej wystarczającego) odzwierciedlenia rzeczywistości. Kolejne etapy rozwoju systemów wspomagających podejmowanie decyzji miały ułatwić posługiwanie się nimi przez użytkownika końcowego, ale wydaje się, że do momentu powstania mechanizmów *Business Analytics* w systemach *Business Intelligence* problem ten do końca nie został rozwiązany.

Strukturę logiczną systemu klasy DSS przedstawia rysunek 19.



**Rysunek 19.** Architektura logiczna informatycznego systemu wspomaganie podejmowania decyzji

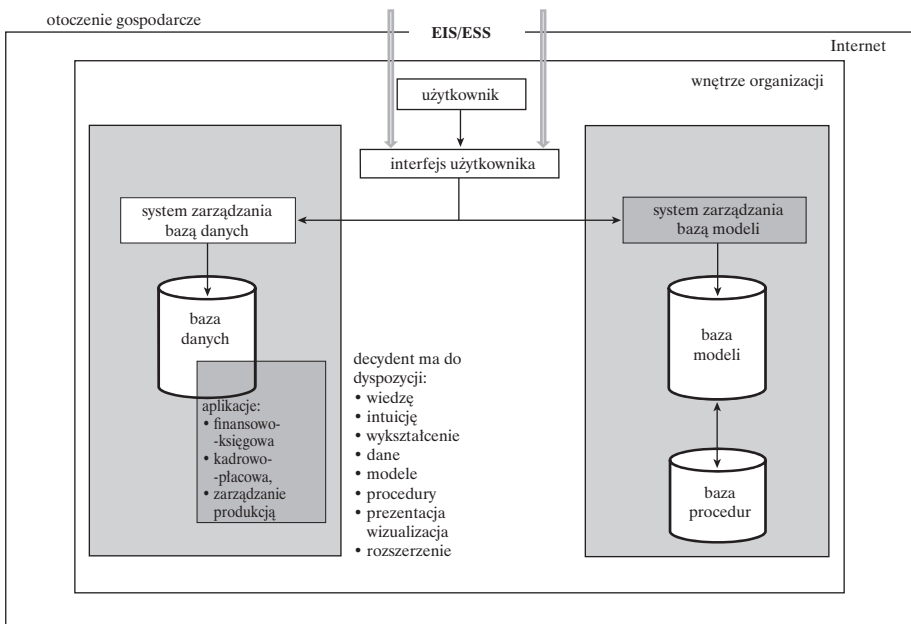
Źródło: opracowanie własne.

### 3.2.4. Systemy informowania kierownictwa (systemy informacyjne kierownictwa – *Executive Information Systems* – EIS i systemy wspomaganie kierownictwa – *Executive Support Systems* – ESS – od 1980 r.)

Odpowiedzią na pierwszy z wyróżnionych problemów, czyli kłopoty komunikacyjne, miały się stać systemy informowania kierownictwa. Idea ich powstania była niezwykle prosta – miały zapewnić bezpośredni dostęp do możliwości systemu najwyższej kadrze kierowniczej, nieprofesjonalnej pod względem posługiwania się systemami informatycznymi. Początkowo idea ta była realizowana poprzez zwiększenie możliwości prezentacyjnych danych z bazy danych oraz zrozumiałej prezentacji wyników przetwarzania modeli.

W zasadzie do nowych elementów wprowadzonych przez systemy EIS należy więc zaliczyć jedynie rozszerzenie interfejsu użytkownika lub systemu zarządzania bazą danych o zwiększenie możliwości porządkowania i selekcji danych

(najlepiej bez znajomości struktury bazy danych) oraz wizualizację graficzną uzyskanych wyników. Raporty uzupełnione danymi z bazy danych można było albo z góry zorganizować wg najczęściej przez użytkownika stosowanych przekrojów, albo dać mu samemu możliwość ich generowania. Ta druga możliwość wymagała jednak od niego znajomości przynajmniej struktury zbiorów w bazie danych i ich zawartości bądź należało mu stworzyć możliwość korzystania z tzw. języków komunikacji z bazą danych. Wizualizacja graficzna np. w postaci wykresu strukturalnego czy dynamicznego powodowała, że decydent już na pierwszy rzut oka potrafił ocenić strukturę analizowanego zjawiska w danym momencie czasowym lub jego rozwój w czasie. Dodatkowo starano się – chyba po raz pierwszy w historii rozwoju systemów informatycznych – o zapewnienie dopływu danych zewnętrznych w celu umożliwienia porównań z sytuacją innych firm w danej branży w kraju bądź za granicą. Dane te co prawda na początku były ręcznie wprowadzone do systemu, ale był to prawidłowy krok w kierunku rozszerzenia systemu poza organizację.



**Rysunek 20.** Architektura logiczna systemu informatycznego klasy EIS/ESS

Źródło: opracowanie własne.

Z kolei systemy ESS – stanowiące niejako lustrzane odbicie systemów EIS, a przez niektórych badaczy uważane za następny etap rozwoju systemów infor-



mowania kierownictwa, pozwalały na łatwiejszą manipulację wynikami uzyskiwanymi poprzez przetworzenie modeli, co niekiedy sprowadzało się jedynie do możliwości transferu wyników przetworzenia do arkusza kalkulacyjnego. Niekiedy jednak budowano własne oprogramowanie umiejscowione głównie w interfejsie użytkownika.

Strukturę logiczną systemów EIS/ESS przedstawia rysunek 20.

### 3.2.5. Systemy eksperckie (*Expert Systems* – ES – I generacja – od 1975 r., II generacja – od 1985 r.)

W.A. Freyfeld (1984, za: Chmielarz, 1996, s. 126) określił system ekspercki jako „... system zawierający w sobie specjalizowaną wiedzę na temat określonego obszaru ludzkiej działalności zorganizowaną w sposób umożliwiający wejście z użytkownikiem w dialog dotyczący tego obszaru, na podstawie którego system może oferować rady lub propozycje, oraz objaśniać sposób rozumowania, leżący u ich podstaw...”. Ta stara definicja wyjaśnia chyba w najlepszy sposób działanie systemów eksperckich.

Pierwsze z nich, konstruowane jeszcze w latach siedemdziesiątych, architektonicznie nie zawierały nic nowego – opierały się na istniejącej w wielu językach programowania konstrukcji skoku warunkowego (jeżeli... – to...) lub bezwarunkowego (przejdź do...). Niemniej jednak pierwsze wcale wyrafinowane systemy dotyczące rozwiązań problemów zdrowotnych (np. MYCIN) powstały niemal wyłącznie na podstawie tej zasady. Miały wówczas charakter ściśle branżowy lub problemowy i ich zastosowanie ze względu na tak ukształtowaną funkcjonalność było mocno ograniczone. Początkowo systemy klasy ES mylono z systemami klasy DSS. Jako podstawowy wyróżnik przyjęto szerszy zakres systemu wspomagającego podejmowanie decyzji (dotyczący np. całości organizacji), opierały się też na bardziej wyrafinowanych, a czasem nawet abstrakcyjnych modelach ekonomicznych i ekonometrycznych. Systemy eksperckie dotyczyły wybranego, wąskiego problemu o skomplikowanych relacjach pomiędzy składowymi oraz złożonej algorytmice.

Dopiero druga generacja systemów eksperckich, znajdująca swoje oparcie w ideach tzw. systemów sztucznej inteligencji, twórczo rozwinęła logiczną konstrukcję architektoniczną poprzednich klas systemów. Oprócz nieco sztucznie, bo zewnętrznie wobec struktur przedsiębiorstwa, wyodrębnionych modeli ekonometrycznych, statystycznych, prognostycznych itd. wzięto pod uwagę możliwość skorzystania z modeli opartych na najnowszych wówczas zdobyczach metodycznych zarządzania (*Business Process Reengineering* – BPR): modelach najlepszych praktyk zarządzania organizacją; analizie i w potocznym sensie optymalizacji funkcji i procesów zachodzących w przedsiębiorstwie, w istniejącej lub modyfikowanej strukturze organizacyjnej przedsiębiorstwa. Pojawiły się

wówczas w systemach informatycznych nowe elementy strukturalne, z których najważniejsze wydają się:

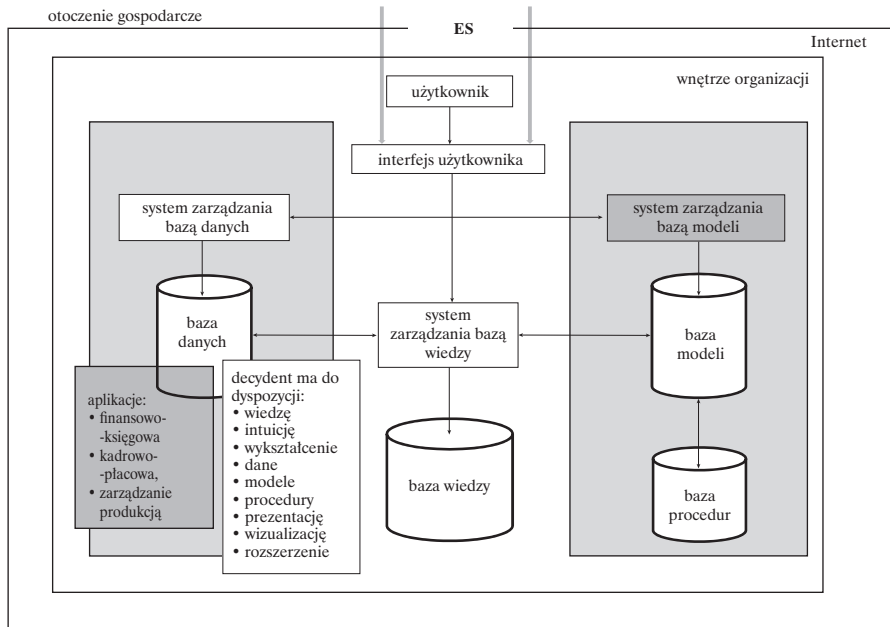
- baza wiedzy – zasób wiedzy niezbędny do podejmowania decyzji, przechowywane, udostępniane i modyfikowane modele najlepszych praktyk zarządzania organizacją, składające się z wzorców, w których znajdują się powiązane za pomocą reguł działania i fakty dotyczące zdefiniowanej sytuacji gospodarczej wymagającej podjęcia decyzji wraz z przykładowymi sytuacjami, w jakich je podjęto;
- system zarządzania bazą wiedzy – składający się z podsystemu pozyskiwania wiedzy, podsystemu wnioskowania, podsystemu interpretującego skutki podjętych decyzji oraz podsystemu udoskonalającego przechowywaną wiedzę; podsystem pozyskiwania wiedzy służy do gromadzenia, przesyłania i przetwarzania wiedzy dotyczącej rozwiązania danego problemu, pochodzącej z różnych źródeł: eksperta „ludzkiego”, podręczników, obcych baz danych, raportów specjalnych, badań, Internetu itp. oraz posiada mechanizmy wspomagające proces uczenia się systemu; podsystem wnioskowania składa się z trzech głównych elementów: tłumacza – interpretera reguł (wykonującego określone działania skojarzone z odpowiednimi regułami bazy wiedzy); planisty – utrzymującego kontrolę porządku wykonywanych działań i szacującego skutki zastosowania reguł wnioskowania z punktu widzenia zastosowanych priorytetów oraz mechanizmu wymuszania spójności pojawiających się rozwiązań.

Podsystem interpretujący skutki podjętych decyzji wyjaśnia, w jaki sposób należy używać systemu eksperckiego, jak osiągnięto poszczególne konkluzje, czy istnieją alternatywne ścieżki osiągnięcia zakładanego rozwiązania, jeżeli tak – dlaczego niektóre z nich odrzucono. Podsystem udoskonalania wiedzy – gromadzi najlepsze wśród aktualnie w danej sytuacji wypracowanych rozwiązań z punktu widzenia zakładanych celów lub kryteriów oceny.

Zakłada się również doskonalenie interfejsu użytkownika w kierunku ułatwienia komunikacji za pomocą grafiki i docelowo – języka naturalnego.

Z punktu widzenia decydenta systemy eksperckie dostarczają mu nowego narzędzia do podejmowania decyzji – oprócz strukturalizowanych danych z bazy danych, rozwiązań modelowych na podstawie bazy modeli otrzymuje on trzecią możliwość – sugerowane rozwiązania budowane na podstawie najlepszych praktyk zarządzania. W każdym z tych trzech przypadków dysponuje oczywiście jeszcze swoją wiedzą fachową, umiejętnościami oraz wypracowaną w ciągu lat praktyki intuicją w podejmowaniu decyzji gospodarczych. To daje mu szansę w łatwiejszy sposób podjąć taką decyzję, która zapewni organizacji największe korzyści lub uchroni ją od strat.

Strukturę logiczną takiego rozwiązania przedstawia rysunek 21.



**Rysunek 21.** Architektura logiczna systemu eksperckiego

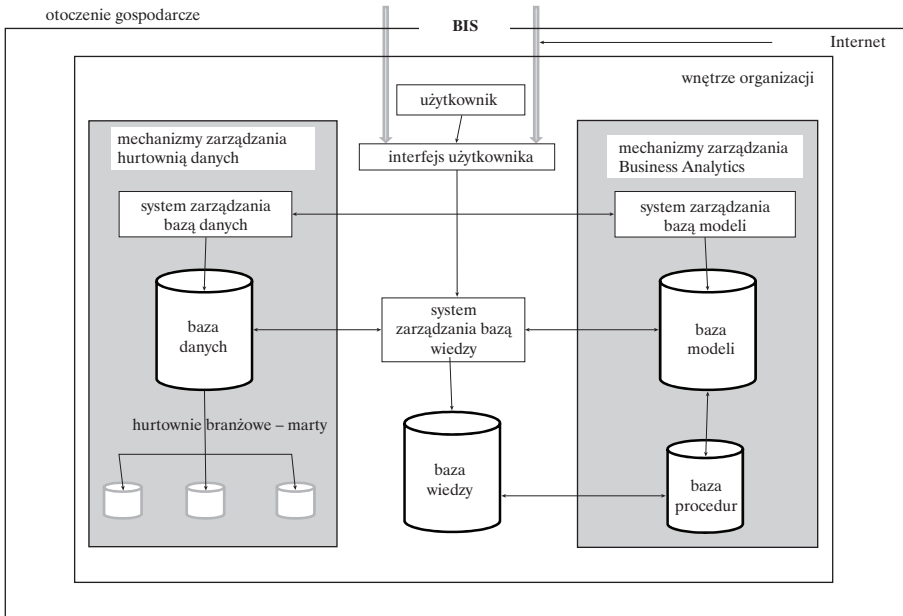
Źródło: opracowanie własne.

### 3.2.6. Systemy *Business Intelligence* (*Business Intelligence Systems*) – od 1990 r.

Wydawać by się mogło, że już bardzo niewiele można dodać do tak skomplikowanej struktury, jaka została zaproponowana przez systemy eksperckie. A jednak opierając się na rozwiązaniach wypracowanych z jednej strony przez masowe przetwarzanie danych, a drugiej – przez powstające w tym czasie systemy sztucznej inteligencji (*Artificial Intelligence Systems* – AIS), zaczęto budować nowe konstrukty modelowe przez udoskonalenie struktury lub funkcjonalności poprzednich. Taka sytuacja – nie tylko w informatyce – cyklicznie się powtarza, wykorzystano ją i tym razem. Z definicji BI to „...system informacyjno-analityczny zbudowany w oparciu o hurtownię danych wraz z mechanizmami zbierania danych oraz wykorzystujący różne narzędzia analityczne, w szczególności narzędzia służące do analizy wielowymiarowej oraz eksploracji danych...”. Definicja ta wskazuje kierunki przemian, które nastąpiły od czasu powstania systemów DSS czy ES.

Po pierwsze – to wyraźne rozbudowanie bazy danych, związane z masowością i zróżnicowaniem danych przetwarzanych w systemach informatycznych

w kierunku stworzenia hurtowni danych. Hurtownia danych to w zasadzie – rozszerzona korporacyjna baza danych wraz z mechanizmami ekstrakcji danych z heterogenicznych (w tym zewnętrznych) źródeł danych oraz procesami ich przetwarzania do postaci wspólnej i odpowiedniej dla analityków i użytkowników podejmujących decyzje biznesowe, wspomagana przez dziedzinowe lub branżowe bazy danych (*mart*) i obudowana mechanizmami współpracy z narzędziami analitycznymi. Do podstawowych zadań bazy danych (danych bieżących) należy więc oprócz standardowego raportowania i definiowania raportów i zapytań *ad-hoc przez użytkownika*, analiza statystyczna, interaktywne przetwarzanie analityczne, eksploracja danych (*data mining*) oraz – w ograniczonym zakresie – modelowanie biznesu, czyli – jak wynika z powyższego – nastąpiła zmiana jakościowa na poziomie głównego źródła informacji w systemie.



**Rysunek 22.** Architektura logiczna systemu *Business Intelligence*

Źródło: opracowanie własne.

Po drugie, nastąpiła również zmiana jakościowa wspomaganego modelowego w stosunku do poprzednich klas systemów. Tzw. *Business Analytics* to wszelkiego rodzaju narzędzia i aplikacje analityczne służące do szeroko rozumianego zarządzania wydajnością przedsiębiorstwa (*corporate performance management*). Wśród narzędzi i aplikacji do zarządzania wydajnością wyróżnia się uniwersal-

ne narzędzia analityczne, narzędzia służące do analizy danych przestrzennych zapisanych w bazach danych systemów informacji przestrzennej oraz aplikacje analityczne przeznaczone dla konkretnych obszarów zarządzania przedsiębiorstwem, takich jak: zarządzanie finansami i strategią, zarządzanie relacjami z klientem, zarządzanie pracownikami, zarządzanie łańcuchem dostaw itp. (zob. Gołuchowski, 2007; Olszak, 2007).

W sumie użytkownik końcowy dla swych celów otrzymuje wielowymiarowe, niesłychanie skomplikowane narzędzie, którego nadmiarowość funkcji może mu wręcz przeszkadzać w podejmowaniu decyzji, nawet jeśli sięgnie po nowe media komunikacyjne (w rodzaju kokpitu menedżerskiego).

### 3.2.7. Integracja „wewnętrzna” systemów informacyjnych zarządzania

Oczywiście systemy informacyjne zarządzania nie rozwijały się w sposób odosobniony. Ich praktyczna użyteczność rzeczywiście rozpoczęła się wraz z postępowaniem nad integracją na poziomie danych i wizualizacji funkcjonalnej, niemniej równie powszechnym zjawiskiem było łączenie różnych klas systemów ze sobą, co powodowało, czasami tymczasowe, istotne przemiany jakościowe.

Przykładowo, połączenie systemów informacyjnych zarządzania oraz systemów eksperckich miało dwustronne implikacje. Systemy informacyjne zarządzania dostarczają w tym tandemie informacji dla systemów eksperckich oraz ułatwiają manipulację danymi. Program aplikacyjny może wykorzystywać dane bezpośrednio z bazy danych, jak również może korzystać z danych pobranych i wstępnie zinterpretowanych przez mechanizmy systemu eksperckiego. Systemy eksperckie służyły w tym połączeniu jako rozszerzenie funkcji systemu zarządzania bazą danych (poprzez nadzór proces zapisu, odzyskiwania i uaktualniania informacji), ułatwiają operatorom poprawne zarządzanie bazami, szczególnie w przypadku, rozproszonych baz danych, optymalizują zapytania i ścieżki poszukiwań oraz ilość przesyłanych danych, czy występują jako inteligentne interfejsy w złożeniach komercyjnych, strukturalnych baz danych.

Łączenie architektoniczne systemów wspomagania decyzji i systemów eksperckich dawało następujące efekty (Adamczyk i Chmielarz, 2005):

- możliwości logicznego wyjaśnienia podejmowanych akcji i uzyskanych wyników – w przypadku dołączenia systemu eksperckiego do elementów systemu wspomagania decyzji;
- szybszego wykonywania operacji, gdy rezultaty działania systemu wspomagania są danymi wejściowymi do systemu eksperckiego;
- poprawnej identyfikacji problemu w sytuacji odwrotnej;
- zwiększenia możliwości wyboru użytkownika – korzystania z systemów dwóch typów sprzężonych bazą danych w zależności od bieżących potrzeb logicznego procesu decyzyjnego;

- generowania rozwiązań wariantowych (systemy wspomaganie decyzji) i dołączanie do nich alternatywnych działań, które powinny być podjęte dla ich uzyskania.

Podstawową metodą integracji systemów informowania kierownictwa i systemów wspomaganie decyzji (w rezultacie powstaje forma systemu klasy ESS) jest wykorzystywanie przez systemy DSS informacji generowanych przez system informowania kierownictwa jako informacji wejściowych. W bardziej skomplikowanych przypadkach przewiduje się istnienie pętli sprzężenia zwrotnego uruchamianej przez specjalny inteligentny interfejs, który ze strony systemu informowania kierownictwa będzie pozwalał na tworzenie zapytań do systemu wspomaganie, a w odwrotnym kierunku będzie przysyłał interpretacje i rekomendacje uzyskane od systemu wspomaganie. W skrócie można ten proces zawrzeć w dwóch stwierdzeniach:

- dane po wstępnej obróbce w systemie informowania kierownictwa będą użyte jako dane wejściowe do systemów wspomaganie decyzji,
- system informowania kierownictwa służy do dalszych interpretacji rozwiązań uzyskanych za pomocą systemu wspomaganie decyzji.

Połączenie systemów eksperckich z systemami informowania kierownictwa w praktyce jest stosowane bardzo rzadko. Systemy informowania kierownictwa mogą się zwracać z zapytaniem do systemów eksperckich w kwestii rozwiązania szczególnych, specjalistycznych problemów, którymi te ostatnie się zajmują i w zamian uzyskiwać interpretacje dotyczące rozwiązań. Systemy informowania kierownictwa mogą też odwoływać się do bazy wiedzy eksperckiej w sytuacjach, gdy ich interfejs użytkowy wyposażony jest w mechanizmy komunikacyjne, które mu na takie działanie pozwalają. Niekiedy system ekspercki występuje w roli regularnego dostawcy sprawozdań (wraz z ich automatyczną oceną merytoryczną) generowanych na podstawie danych przesyłanych z systemu informowania kierownictwa.

Tego typu powiązania pomimo swych walorów praktycznych, z teoretycznego punktu widzenia zaciemniały obraz klasyfikacji systemów informacyjnych zarządzania. Zwłaszcza, że jednocześnie rozwijały się one w drodze integracji funkcjonalnej.

### **3.3. Rozwój w drodze integracji funkcjonalnej systemów informacyjnych zarządzania**

Integracja funkcjonalna oznacza, że różne funkcje systemu informatycznego są realizowane w taki sposób, jak byłyby wykonywane w jednym, pojedynczym

systemie. W ten sposób z jednego stanowiska roboczego mamy teoretycznie dostęp do wszystkich możliwych form działalności istniejących w systemie poprzez jeden spójny interfejs i możliwość przełączania się pomiędzy różnymi zadaniami. W systemach zintegrowanych oznacza to dostęp do dowolnego systemu funkcyjnego oraz wszelkie możliwości współdziałania z poziomu tego systemu z innymi narzędziami zawartymi w systemie. Integracja funkcjonalna nie jest pomysłem nowym – od samego początku istnienia systemów informatycznych próbowano doprowadzić do jej urzeczywistnienia, najpierw w ramach określonej dziedziny działalności, a potem całej już organizacji. Dla zbudowania zintegrowanego systemu informatycznego trzeba więc uzyskać wspólny dla całej organizacji zasób informacji, jednolity standard gromadzenia, przetwarzania i przesyłania informacji, jednolitą postać mediów gromadzenia i obróbki informacji, wspólne narzędzia i procedury rozwoju systemu oraz jednolitą procedurę prowadzenia dialogu z użytkownikiem. Niemniej jednak tworzenie zintegrowanych systemów zarządzania przebiegało stopniowo, początkowo przez dodawanie w ramach systemu obsługi nowych funkcji do już istniejących, w sposób coraz bardziej dostosowany do aktualnego stopnia rozwoju technologii informatycznych. Nie można też pominąć faktu, że budowa systemów zintegrowanych „wewnętrznie” była odpowiedzią na problemy związane z istnieniem w jednej firmie kilkudziesięciu nieraz jednodziedzinowych systemów informacyjnych zarządzania o zróżnicowanej postaci, formie, strukturze i przeznaczeniu oraz problemy związane z próbami ich integracji „zewnętrznej”.

### 3.3.1. Zintegrowane systemy planowania zasobów materiałowych (*Material Requirements Planning – MRP – od 1959 r.*)

Pomysł na stworzenie systemów zintegrowanych był następujący – projektujemy system odzwierciedlający w najwyższym możliwym stopniu procesy zachodzące w przedsiębiorstwie oparte na bilansie materiałowym (z jednej strony surowce, materiały, półprodukty, komponenty, z drugiej – cały asortyment produktów gotowych), znając zależności pomiędzy nimi wyznaczone obowiązującymi przepisami i normatywami. Wychodzono z założenia, że wyznaczając zapotrzebowanie na zasoby materiałowe i harmonogram ich dostaw, można zaplanować optymalną produkcję, pod względem kosztów, czasu wykonania, opłacalności itp.

Pierwszy system spełniający (choć w małym stopniu) przyjęte założenia – zarządzanie zapasami magazynowymi (*Inventory Control IC*) – powstał w 1964 r., dotyczył jednak tylko obsługi jednej z najłatwiejszych dziedzin działalności przedsiębiorstwa, jaką jest gospodarka magazynowa (Adamczyk i Chmielarz, 2005). Następne były systemy sterowania produkcją. Początkowo systemy te tworzone były głównie na potrzeby seryjnej produkcji przemysłowej, w szcze-

gólności elektromaszynowej. Liczba branż, dla których opracowywano wspomaganie komputerowe systematycznie rosła. Do ich rozwoju przyczyniło się powstanie i rozwój norm produkcji przemysłowej i norm tworzenia systemów informatycznych zarządzania produkcją.

W Stanach Zjednoczonych, w 1957 roku, powstało Amerykańskie Stowarzyszenie Sterowania Produkcją i Zapasami (American Production & Inventory Control Society), w skrócie APICS. Postawiło ono sobie za cel opracowanie metod wykorzystania komputerów w zarządzaniu organizacjami produkcyjnymi. Pod koniec lat pięćdziesiątych APICS opracował założenia standardu Planowania Potrzeb Materiałowych, czyli *Material Requirements Planning* (MRP). MRP pozwalała obliczyć dokładną ilość materiałów i ustalić terminarz ich dostaw tak, aby sprostać ciągle zmieniającemu się popytowi na poszczególne produkty. Celami MRP były: redukcja zapasów magazynowych i międzyoperacyjnych, dokładne określanie czasów dostaw surowców i półproduktów, dokładne wyznaczanie kosztów produkcji, lepsze wykorzystanie posiadanej infrastruktury wytwórczej, szybsze reagowanie na zmiany zachodzące w otoczeniu, kontrola poszczególnych etapów produkcji. Rozszerzeniem specyfikacji MRP było uwzględnienie działania zamkniętej pętli sprzężenia zwrotnego (*Closed Loop MRP*), czyli planowania potrzeb materiałowych i zdolności produkcyjnych w zamkniętej pętli procesu produkcyjnego. Dzięki sprzężeniu zwrotnemu można było na bieżąco reagować na zmieniające się parametry produkcji.

### 3.3.2. Zintegrowane systemy planowania zasobów produkcyjnych (*Manufacturing Resource Planning – MRP II – od 1989 r.*)

Blisko trzydzieści lat trwało opracowanie nowego standardu systemów zintegrowanych, który powstał dopiero pod koniec lat osiemdziesiątych. W 1989 roku APICS opracowało standard MRP II (*Manufacturing Resource Planning – Planowanie Zasobów Produkcyjnych*), który przyjął się jako powszechnie stosowany we wszystkich większych zintegrowanych systemach informacyjnych zarządzania (Lech, 2003). Standard ten w stosunku do poprzedniego został rozbudowany o elementy związane z procesem sprzedaży i wspierające podejmowanie decyzji na szczeblach strategicznego zarządzania produkcją. W miarę rozwoju MRP obejmowało kolejne obszary działalności przedsiębiorstwa, stając się stopniowo systemem zawierającym wszystkie podstawowe procesy zachodzące w przedsiębiorstwie. W modelu MRP II bierze się pod uwagę wszystkie sfery zarządzania przedsiębiorstwem związane z przygotowaniem produkcji, jej planowaniem i kontrolą oraz sprzedażą i dystrybucją wyprodukowanych dóbr. Poza materiałami bezpośrednio produkcyjnymi, MRP II uwzględnia także materiały pomocnicze, zasoby ludzkie, pieniądze, czas, środki trwałe itp. Aby możliwe było współdziałanie modułów produkcyjnych z pozostałymi, zaczęto opierać



integrację wszystkich podsystemów w systemie zintegrowanym na bilansie finansowym, zamiast na bilansie produkcyjnym. To z kolei umożliwiło przeniesienie idei integracji systemów w obszarze organizacji poza sektor produkcyjny do sektora handlu, usług i finansów. Pogłębiło to ekstensywny rozwój tych systemów i opanowanie przez nie nowych rynków.

Pod względem architektury logicznej systemu zarówno MRP, jak i MRP II nie wyszły jednak poza obręb systemów informatycznych zarządzania (MIS). W kolejnych ich wersjach i mutacjach integracja następowała głównie na płaszczyźnie funkcjonalnej, nie zaś komplikacji elementów logicznych. Komplikacja zaś następowała w sferze łączenia ze sobą kolejnych współpracujących modułów, ale de facto nadal były to systemy aplikacyjne oparte na bazie danych i systemie zarządzania bazą danych. Ich duża popularność (do 80% rynków) wywodziła się głównie z łatwości i bezproblemowości obsługi odzwierciedlającej procesy zachodzące w organizacji. Natomiast wspomaganie procesów zarządzania sprowadzało się do dostarczania raportów o ograniczonej wartości analitycznej.

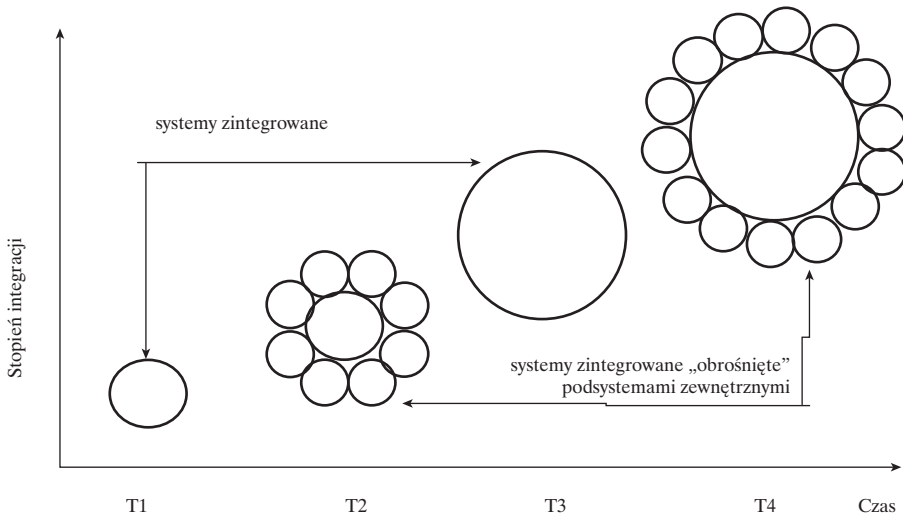
### **3.3.3. Zintegrowane systemy planowania zasobów przedsiębiorstwa lub planowanie zasobów na potrzeby przedsięwzięć (Enterprise Resource Planning – ERP – od 1995 r.; ERP II – od 1998 r.)**

Kolejnym krokiem w rozwoju systemów zintegrowanych – stymulowanym ciągle przez dodawanie nowych podsystemów i funkcji do już istniejących – było przyjęcie w połowie lat dziewięćdziesiątych zakresu systemów odpowiadającego planowaniu zasobów na potrzeby przedsięwzięć (*Enterprise Resource Planning* – ERP). Głównym celem tych systemów stała się możliwie najpełniejsza integracja wszystkich funkcjonalności na wszystkich szczeblach zarządzania przedsiębiorstwem. Nowoczesny ERP szybko stał się systemem obejmującym całość procesów produkcji i dystrybucji, który integruje różne obszary działania przedsiębiorstwa, usprawnia przepływ krytycznych dla jego funkcjonowania informacji i pozwala szybko odpowiadać na zmiany popytu zgłoszonego z zewnątrz. Informacje te są uaktualniane w czasie rzeczywistym i dostępne w momencie podejmowania decyzji. Jednymi z najistotniejszych wyróżników tego typu systemów jest zastosowanie, dwukierunkowych mechanizmów optymalizujących planowanie oraz wbudowana w system możliwość integracji z zewnętrznymi podmiotami w ramach łańcucha dostaw i sprzedaży. Ponadto, w ERP stosowane są mechanizmy umożliwiające symulowanie różnorodnych działań i analizę ich skutków, w tym finansowych. ERP obejmuje następujące obszary (Januszewski, 2008):

- komunikacja z klientami i obsługa klientów – baza danych o klientach, przetwarzanie zamówień, obsługa zamówień, elektroniczny transfer dokumentów (EDI), łączność z systemami zewnętrznymi w tym z Internetem;

- sfera produkcji – zarządzanie magazynami, wyznaczanie kosztów produkcji, zakupy surowców i półproduktów, ustalanie harmonogramu produkcji, zarządzanie zmianami produktów, sterowanie produkcją, (MRPI/II), prognozowanie zdolności produkcyjnych, wyznaczanie krytycznego poziomu zasobów/zapasów, kontrola procesu produkcji itp.;
- obszar finansów – prowadzenie pełnej księgowości, kontrola przepływu dokumentów finansowo-księgowych – pozwala przygotowywać raporty finansowe zgodnie z oczekiwaniami poszczególnych grup odbiorców;
- integracja w ramach całego łańcucha logistycznego – jeden z kierunków rozwoju systemów ERP (zamówienia – dostawa – produkcja – dystrybucja-płatności).

W podstawowej architekturze w zasadzie zmiany nie nastąpiły, pomimo przenikania (konwergencji) idei *Business Intelligence Systems* do systemów zintegrowanych (np. produkty SAS Institute). W szczególności dotyczyło to możliwości systemów związanych z zastosowaniem hurtowni danych. Zmieniło się podejście do kierunków integracji w ramach systemów zintegrowanych. Po pierwsze, integracja oparta na systemach wewnątrz organizacji zaczęła ustępować idei tzw. systemów rozszerzonych – otwartych na otoczenie gospodarcze (podobnie jak – zapoczątkowane to zostało w systemach EIS i ESS). To rozszerzenie miało obejmować głównie informację z zewnątrz, przede wszystkim ze sfery Internetu i ich konwersję na taki format, który można wspólnie przetwarzać w ramach podejmowania decyzji. To umożliwiło objęcie przetwarzaniem całego łańcucha logistycznego i analizy porównawcze. Po drugie, kolejne implementacje uprzytomniły nie tylko klientom, lecz także twórcom tego oprogramowania, że integracja poprzez dodawanie kolejnych podsystemów, modułów i funkcji na etapie rozwoju osiągniętym w latach dziewięćdziesiątych jest drogą donikąd. W setkach tysięcy przeplatających się funkcji i klienci, i nawet producenci czy dystrybutorzy przestali się sprawnie poruszać. Duże i ciężkie systemy zintegrowane typu SAP R3 były tak wielkie, że nie istnieli specjaliści znający się na całości systemu, a znający się jedynie na poszczególnych modułach. Systemy były coraz droższe i coraz wolniejsze. Potęgował to tzw. efekt słonecznika. Zintegrowane początkowo systemy, wraz ze zmianami w otoczeniu systemu obrastały masą zintegrowanych z nimi zewnętrznie podsystemów, otaczających wianuszkami system zintegrowany. Powodowało to spowolnienie działania systemu zintegrowanego i konieczność integracji wszystkich tych podsystemów z uprzednim poziomem zintegrowanym. Po pewnym czasie proces się powtarzał dla kolejnego, większego od poprzedniego systemu. Kształt odwzorowań pośrednich faz rozwoju systemów zintegrowanych w czasie T2 i T4 przypomina kwiat – stąd nazwa – „efekt słonecznika” (zob. rys. 23).

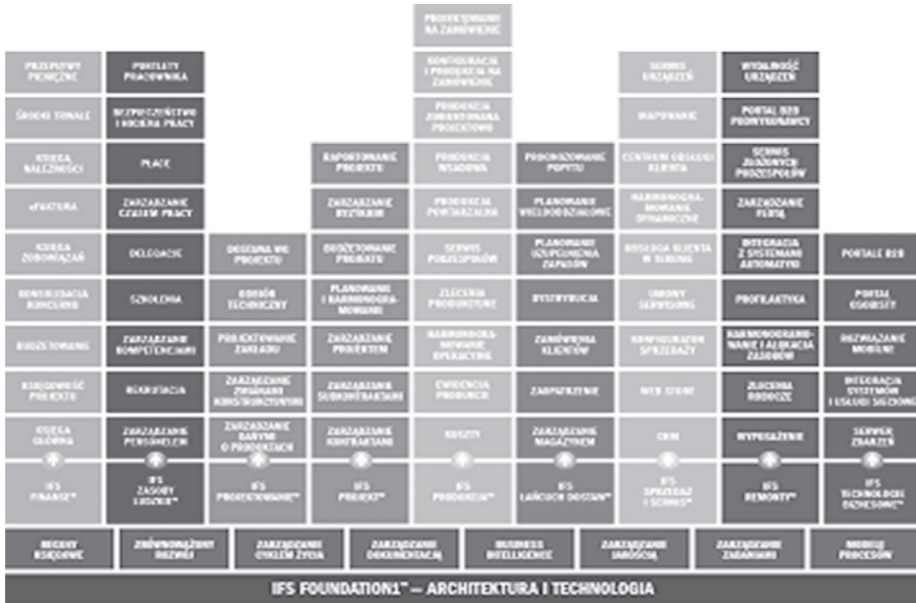


**Rysunek 23.** „Efekt słonecznika” w rozwoju zintegrowanych systemów zarządzania

Źródło: opracowanie własne.

Systemy stają się więc anormalnie olbrzymie i trudne do okiełzania przez użytkownika. A nietypowe funkcje systemu trudno było w pełni zintegrować z całym systemem. Przeszły działały takie mechanizmy, które w sposób automatyczny integrowały modyfikacje systemu (np. *Dynamic Enterprise Modeler* (DEM) systemu BaaN IV). Zaczęto powoli powracać do pochodzących z lat siedemdziesiątych ubiegłego wieku systemów branżowych (usunięcie funkcji „zbędnych”, okrojenie systemu do potrzeb użytkownika, tworzenie systemu pod potrzeby użytkownika – pełna klasteryzacja itp.). Powrócono także do idei systemów klockowo-modułowych, czyli takich, które w miarę zgłaszania potrzeb przez użytkownika dają się składać z poszczególnych elementów w jedną funkcjonalną postać w ramach wytyczonych przez ramę systemu (oczywiście w ramach rozsądnych ograniczeń – bilansu finansowego lub materialnego). W tej sytuacji stworzenie jednego standardowego systemu klasy ERP nie wydawało się możliwe. Tak więc naokoło znormalizowanego zakresu systemu ERP zaczęły powstawać jego liczne odmiany i mutacje.

Maksymalna struktura funkcjonalna standardowego systemu zintegrowanego przedstawiona została na przykładzie zaczerpniętym z materiałów firmy IFS (rys. 24). Jak widać obejmuje on – jak każdy system klasy ERP – wszystkie możliwe komponenty działalności większości organizacji na rynku. Budowa komponentowa umożliwia wybór zakresu systemu w zależności od branży, w której organizacja operuje.



**Rysunek 24.** Struktura systemu zintegrowanego na przykładzie systemu IFS Application

Źródło: <http://www.ifsworld.com/pl-pl/solutions/ifs-applications-8/>

Najważniejszym wyróżnikiem zróżnicowania systemów ERP i powstania następnej generacji systemów – ERP II, stały się ich relacje do systemów sieciowych. Początki nastąpiły w trakcie rozwijania podsystemów/modułów łączności z kontrahentami. Naturalną niejako konsekwencją rozwoju „ręcznych” i za pośrednictwem laptopa kontaktów z klientami stała się łączność za pomocą sieci Internet. W momencie utworzenia serwisów internetowych z zewnętrznym dostępem do systemu kontrahenci mogą przeglądać aktualną ofertę, składać na nią zamówienia, dowiadywać się o bieżącym statusie realizacji zamówienia, stanach półproduktów itp. Podobnie pracownicy organizacji, nawet przebywając poza firmą, mogą uzyskiwać informacje z interesujących ich zakresów. Wyszczególniono też klasę systemów, w których dominującą sferą była przestrzeń internetowa i do niej zostały przeniesione podstawowe funkcje systemu – czyli działające w tej sferze ERP (*Electronic Enterprise Resource Planning* – e-ERP). Ich współdziałanie z gospodarką elektroniczną obejmuje obsługę różnych form transakcji elektronicznych (np. B2B, B2C, B2P itp.), budowę interfejsów do istniejących systemów e-biznesowych (typu e-commerce, c-commerce), jak również wiązanie z systemami mobilnymi (m-commerce) oraz

dzielenie się zasobami i funkcjonalnościami ERP poprzez portale korporacyjne w ramach sieci intranetu gospodarczego, czy ekstranetu. Niekiedy wyróżnia się jeszcze inne rodzaje systemów zintegrowanych klasy ERP II, takie jak (Korczak i Dyczkowski, 2008, s. 59–60):

- EERP (*Extended Enterprise Resource Planning*), czyli rozszerzone poza obszar organizacji systemy ERP, posiadające dodatkowe funkcjonalności dotyczące podmiotów znajdujących się w otoczeniu gospodarczym organizacji, który cechuje się zwiększeniem zakresu wspomaganych procesów gospodarczych;
- @ERP (*active Enterprise Resource Planning*), czyli aktywne systemy ERP, służące do wspomaganie procesów modyfikacji podczas projektowania lub przeprojektowywania systemu ERP (przebudowa struktur organizacyjnych, BPR, praktyk biznesowych) w całym cyklu życia systemu, wspierany – narzędziami typu CASE (*Computer Aided System Engineering*), oparty na technologii komponentowej i/lub na nowym podejściu do integracji systemów (platforma korporacyjna);
- IERP (*Intelligent Enterprise Resource Planning*), czyli system ERP posiadający elementy sztucznej inteligencji, pozwalające na lepsze wykorzystanie zasobów informacyjnych poprzez mechanizmy hurtowni danych oraz wykorzystanie warstwy analitycznej (*Business Analytics*) zwiększający dzięki temu możliwości wspomaganie decyzji, szczególnie w obszarze zarządzania wartością firmy.

Wśród systemów współpracujących z ERP (lub nawet przejmujących jego rolę) największą karierę zrobił system zarządzania relacjami z klientem (*Customer Relationship Management* – CRM) – idea podporządkowana potrzebom odbiorcy (a nie producenta, czy dystrybutora). Na podstawie realizacji tej idei zaczęto konstruować systemy, rozpatrując wymagania najpierw klienta zewnętrznego (indywidualnego, instytucjonalnego), a następnie potrzebom tym podporządkowywać wymagania użytkownika wewnętrznego w organizacji. Innym takim systemem były systemy zarządzania relacjami z dostawcami (*Vendor Relationship Management* – VRM), czy obejmujący cały proces logistyczny (*Supply Chain Management*). Zwiększyło to możliwość wyboru w dużej grupie systemów zintegrowanych, zmniejszyło ich średni koszt implementacji, ale podważyło jednocześnie wiarę w jedynie słuszną do tej pory integrację na platformie systemów zintegrowanych. No i uprzytomniło istnienie przez dłuższy czas lekceważonej ścieżki rozwoju drogą rozwoju infrastruktury sieciowej.

Jednakże, jak wspomniano wcześniej, podstawą architektoniczną tych systemów było wykorzystanie idei systemów informacyjnych zarządzania – aplikacji skupionych wokół bazy, a następnie hurtowni danych. De facto były to więc nadal proste systemy o zmoltiplikowanej do granic możliwości ilości funkcji, oparte na systemie, jeśli tak to można uprościć, zązębiających się wzajemnie bilansów. I tak dla systemów IC był to bilans magazynowy, dla MRP – materiałowy, dla MRPII – finansowy, w przypadku ERP – bilans okołoprodukcyjny,

następnie dla ERP II – logistyczny i pełny bilans komunikacyjny dla eERP, w sumie stanowiąc bilans całościowy organizacji. Prostota działania tych systemów na szczeblu stanowiska pracy nadal jest zachęcającą i atrakcyjną cechą dla użytkowników tych systemów na poszczególnych stanowiskach pracy.

Ilustracją tych zależności jest rysunek 25.

**bilans całościowy organizacji**

bilans produkcyjny	bilans finansowy	bilans okołoprodukcyjny	b k i o l m a u n n s i k a c y j n y
bilans magazynowy IC	MRP	MRP II	ERP
bilans logistyczny, specjalizacje i mutacje			eERP
			ERP II

**Rysunek 25.** Bilanse w systemach zintegrowanych

Źródło: opracowanie własne.

### 3.4. Ścieżka rozwoju drogą infrastrukturalnych rozszerzeń sieciowych – przestrzennych

Jednocześnie, a równolegle w sposób niemal niezauważalny, rozpoczął się rozwój systemów sieciowych (Chmielarz, 2007). Dotyczył on głównie takich działalności organizacyjnych, w których informacje zbierane są w sposób rozproszony, a przetwarzane następnie w sposób scentralizowany, po czym ponownie dystrybuowane przestrzennie. Budowa takich systemów w wyspecjalizowanych branżach (bilety lotnicze, bankowość, turystyka, służba zdrowia itd.) rozpoczęła się już w latach sześćdziesiątych (Chmielarz, 2005a). Niemniej jednak dwie cechy charakterystyczne odstręczały potencjalnych użytkowników: wysoka awaryjność

i wysoka cena prywatnych, jednostkowych sieci tworzonych dla dużych, bogatych odbiorców. Jednak cel takich przedsięwzięć był oczywisty – z jednej strony było to zdobycie silnej przewagi konkurencyjnej za pomocą zupełnie nowej technologii, z drugiej zaś – wygoda użytkownika i to zarówno wewnętrznego, jak i zewnętrznego klienta firmy, który odpowiada na ofertę blisko miejsca zamieszkania i uzyskuje błyskawicznie potwierdzenie. Na bazie tej filozofii, na początku lat siedemdziesiątych, zaczęły się wykształcać standardy organizacyjne i technologiczne, na podstawie których tworzone były sieci prywatne głównie na potrzeby sektorowe lub wąskobranżowe.

### 3.4.1. Systemy oparte na sieciach prywatnych (1970–1990)

Podstawą realizacji większości systemów opierających się na rozwoju sieci prywatnych były – początkowo teoretyczne – koncepcje tzw. elektronicznej wymiany danych (*Electronic Data Interchange* – EDI). Opierały się one na wymianie w określony standardami sposób, ustrukturyzowanych danych handlowych pomiędzy systemami informatycznymi dwóch lub więcej organizacji przeprowadzających transakcje (Layland, 1995), a jej celem jest usprawnienie i automatyzacja tego procesu (Abt, 1998). EDI bazuje na wykorzystaniu standardów związanych z zestawieniem łączy fizycznego oraz standardów dokumentacyjnych EDI. Określają one protokoły transmisji informacji oraz postaci dokumentu, w ramach którego jest ona rejestrowana, przetwarzana i prezentowana. W zakresie formułowania dokumentu z czasem wykształciły się dwie podstawowe drogi jego tworzenia: oparte na dokumencie jednostkowym, złożonym z poszczególnych pól dostosowanych do sytuacji, którą dokument obsługuje (wykorzystywane np. w systemie SWIFT) oraz oparte na dokumencie uniwersalnym, w którym w zależności od sytuacji, w jakiej ma być użyty, wypełnia się jedynie te pola, które są niezbędne do jej obsługi (np. EDIFACT).

Elektroniczna wymiana dokumentów zdecydowanie przyspiesza proces realizacji transakcji w handlu elektronicznym, ale głównie w kontaktach między firmami. W literaturze wyróżnia się trzy rodzaje relacji tworzonych dla elektronicznej wymiany dokumentów:

- korporacyjne (lokalne – koncentryczne), najstarsze połączenia, w których tworzy się platformy sieciowe dla dużych i bardzo dużych korporacji, skupiających partnerów biznesowych (dostawców, kooperantów, dystrybutorów itp.);
- branżowe (pionowe – wertykalne) – typowe niejako dla tej sfery, które obsługują pojedyncze rynki pozwalające rozwijać specyficzne cechy charakterystyczne wyspecyfikowanej branży;
- wielobranżowe (przekrojowe – horyzontalne) – dotyczą funkcjonalnego obszaru działalności, skupiające handel produktami z wielu, powiązanych ze sobą branż; pozwalają na bardziej kompleksowe potraktowanie klienta.

Relacje te znalazły dwadzieścia lat później swoje odzwierciedlenie w konstrukcjach opartych na Internecie.

Niemniej jednak pierwsza generacja (1970–1990) całkowicie jednorodnych sieci prywatnych, własnych i jednostkowych, stworzona została na potrzeby określonej sieci standardów (dokumentacyjnych, komunikacyjnych, transmisyjnych). Łączność w nich odbywała się poprzez napisane specjalnie do tego celu własne oprogramowanie zarówno sieciowe, jak i aplikacyjne, a sam proces transmisyjny odbywał się przez bardzo kosztowne sieci prywatne. W związku z tym mogła ona dotyczyć tylko wielkich, bardzo bogatych i innowacyjnych użytkowników, na ogół w sektorach o wysokiej koncentracji produkcji i kapitału (ponieważ tylko takie mogły sobie na to pozwolić). Rozwiązania takie były bardzo bezpieczne, ze względu na ograniczony dostęp do sieci.

### 3.4.2. Systemy oparte na sieciach komercyjnych (1991–1999)

Generacja druga – sieci komercyjnych (1991–1999) była oparta na rozwoju technologii informacyjnych, postępującej standaryzacji (powstanie nowych standardów branżowych i regionalnych) komunikujących się na platformie komercyjnych sieci VAN (*Value-Added Network*) – medium umożliwiające oprócz transmisji (Niedźwiedziński, 2004), konwersję między różnymi systemami. Charakteryzuje się ona lepszą (mniej awaryjną, bezbłędną) i tańszą transmisją danych. Zmniejszanie kosztów powoduje rozszerzenie zasięgu zastosowań systemów elektronicznego przesyłania dokumentów na przedsiębiorstwa średnie. Sieci są nadal relatywnie bezpieczne w porównaniu z późniejszymi rozwiązaniami internetowymi. Takie sieci stanowią alternatywne rozwiązanie dla firm chętnych do korzystania z gotowej infrastruktury telekomunikacyjnej stworzonej przez zewnętrzne firmy, wnosząc opłatę subskrypcyjną oraz płacąc za przesyłanie siecią danych (Laudon i Laudon, 2010). Tego typu podejście, istniejące równoległe z sieciami prywatnymi, stało się najpowszechniejsze wśród dużych organizacji.

Wady rozwiązań opartych na transferze dokumentów elektronicznych, takie jak (OECD, 1998, s. 12):

- zakres ograniczony do dużych, bogatych firm,
- ograniczenie do transakcji między firmami,
- bardzo wysokie koszty,
- wynikająca ze standardów mała elastyczność,

spowodowały w sumie niewielkie rozpowszechnienie systemów opartych na sieciach komercyjnych.

Jedynym pozytywnym wyjątkiem był utworzony w 1984 r. we Francji system wideo tekstowy Minitel (Wielki, 2000, s. 55). Umożliwił on konsumentom dokonywanie zakupów czy też rezerwację biletów kolejowych, lotniczych, hoteli, elektroniczne płatności, dostęp do baz danych itp. (Benjamin, Malone i Yates,



1989). Przez wiele gremiów był on uważany za najważniejszego prekursora Internetu.

### 3.4.3. Systemy oparte na sieci Internet (od 1995 r.)

Pomimo że historia powstania i kształtowania się Internetu sięga daleko głębiej w przeszłość w jego gospodarczych zastosowaniach, wyróżnia się na ogół trzy zasadnicze fazy: pierwotną (przedkryzysową – do 2001 r.), przejściową (kryzys – 2001–2003) oraz wtórną (postkryzysową, społecznościową od 2004 r.).

W pierwszej, pierwotnej fazie, po ukształtowaniu technicznej infrastruktury Internetu oraz zapewnieniu w nim możliwości prowadzenia biznesu (1991 r.), nastąpiło z jednej strony przekształcenie i przystosowanie systemów sieciowych, tworzonych dotąd na zasadach prywatnych i komercyjnych, do uwarunkowań Internetu i samego Internetu do obsługi zarówno w modelu przedsiębiorstwo–przedsiębiorstwo (Business2Business – B2B), jak i relacji pomiędzy organizacjami w modelu przedsiębiorstwo–klient indywidualny (Business2Customer, B2C). Mechanizmy tych przystosowań kształtowały się do połowy lat dziewięćdziesiątych. Opierały się one na kilku podstawowych narzędziach: poczcie elektronicznej; oprogramowaniu zapewniającym dostęp do zasobów sieci WWW oraz komunikatorach sieciowych.

Wydaje się, że z tej grupy instrumentów najistotniejszym i najpowszechniej używanym narzędziem komunikacyjnym stała się poczta elektroniczna (Wielki, 2012). Powszechność, uniwersalność i, jak się wkrótce okazało, niezbędność w każdej organizacji, niezależnie od branży oraz dla każdego użytkownika indywidualnego nie podlegają dziś dyskusji. Na bazie doświadczeń z pocztą zaczęto budować grupy i listy dyskusyjne; magazyny (biuletyny) elektroniczne oraz tzw. autorespondery.

Po drugie, istotnym elementem wczesnych systemów internetowych stały się narzędzia dostępu do jego zasobów. Dostęp do informacji i wiedzy był jednym z podstawowych filarów ideologii tworzenia tej sieci. Natomiast póki nie pojawiły się przeglądarki internetowe, umożliwiające łatwe i proste przeglądanie zawartości stron WWW, dostępność tej usługi dla niespecjalistów była problematyczna. Wiązało się to z koniecznością uporządkowania zasobów Internetu, co miało ułatwić dostęp do przechowywanych w nim informacji. Dlatego musiał nastąpić i nastąpił w tym czasie rozwój katalogów (*directories*) oraz wyszukiwarek internetowych (*search engines*). Pierwsze polegało na tworzeniu umiejscowionych w sieci, hierarchicznych baz danych (np. Yahoo!Directory), które (podobnie jak katalogi w sferze niesieciowej) porządkowały zawartość stron WWW. Drugie – poprzez zastosowanie najpierw elektronicznych, a później inteligentnych agentów (*intelligent e-agents*), występujących w różnej postaci indeksującej zawartość multimedialnej części Internetu (*crawler, spiders*) – pozwalały na bardziej

wyrafinowane wyszukiwanie w handlu elektronicznym. Ostatnią grupą narzędzi charakterystycznych dla tej fazy rozwoju były komunikatory internetowe (*instant messaging* – IM), umożliwiające rozmowę między ich użytkownikami w czasie rzeczywistym, ułatwiające procesy komunikowania się dla konsumentów indywidualnych i organizacji.

W fazie przejściowej – będącej reakcją na nadto aż optymistyczny wydzwięk lat poprzednich, w wyniku pęknięcia dot.com-owej „bańki internetowej”, nastąpiło znormalnienie relacji do sfery elektronicznej. Sfera biznesu elektronicznego zaczęła podlegać zaś normalnym i sprawdzonym zasadom ekonomii. Liczne upadki firm internetowych zarówno w Stanach Zjednoczonych, jak i w krajach europejskich spowodowane były złymi relacjami inwestycji i kosztów realizacji zamówień w stosunku do zysków, nielogiczną konkurencją cenową w ramach danej branży, złą logistyką dostaw, niskim poziomem organizacyjnym pracy, złym rozeznaniem rynków, niechęcią do działań związanych z rozpoznaniem rynku, wciąż słabą infrastrukturą techniczną (szeroki front inwestycji z lat 1997–2000 dopiero zaczynał przynosić powoli efekty) oraz czynnikami społeczno-kulturowymi – brak tradycji sprzedaży zdalnej, opór przed zmianami i tradycje. Już od drugiej połowy 2001 r. do połowy 2003 r. nastąpiło szybkie wyjście z kryzysu – upadły firmy najsłabsze, pozostałe zmieniały asortyment na inny lub różnorodny, łączyły się z przedsięwzięciami tradycyjnymi lub łączyły się ze sobą. Zaczęto dbać o własną logistykę, powstawały alternatywne systemy dystrybucji oraz rozwinęły się systemy elektronicznych płatności. Pomimo kryzysu nastąpił relatywny, a później i rzeczywisty wartościowy, a potem i ilościowy wzrost zjawiska biznesu elektronicznego, powodowany rosnącymi zakupami w sieci i zacieśniającymi się relacjami między przedsiębiorstwami.

Wtórna, bo opierająca się na wcześniejszych doświadczeniach, faza ewolucji sieci Internetu zaczyna się w zasadzie od roku 2004. Charakteryzują ją szereg zjawisk i w obrębie samego Internetu i w jego otoczeniu, które były całkiem nowe w stosunku do wcześniej rozwijanych. Przede wszystkim trzeba zauważyć rosnącą, a potem pod względem kontaktów dominującą rolę multimedialnej części Internetu. To właśnie wówczas zaczęła się rozwijać technologia Web 2.0, bazująca nie tylko na nowych narzędziach, lecz także na narastającym zaangażowaniu użytkowników. W rezultacie następować zaczęła bardzo wyraźna zmiana jakościowa, jeśli chodzi o całościowe funkcjonowanie tej globalnej sieci oraz sposoby jej wykorzystania przez wszystkich jej użytkowników (Fox i Madden, 2006). Najważniejszymi narzędziami technologii Web 2.0 są: szybsze i wydajniejsze wyszukiwarki nowej generacji, oprogramowanie opierające się na aktywności użytkownika – z mechanizmami typu Wiki, szeroka gama blogów, podcasty i wideocasty, światy wirtualne oraz portale społecznościowe. Niekiedy do tej grupy zalicza się również kanały RSS i sieci *peer-to-peer*. Najistotniejsze wydają się tu wyszukiwarki nowego typu np. Google, traktujące Internet

całościowo jako „społeczność”, nastawione na dostosowywanie mechanizmów wyszukiwania i porządkowania stron do popularności stron wśród użytkowników. Ta popularność ma decydujący wpływ na pozycję określonej strony w kolejności wyszukiwania. Wyszukiwarki te zarówno ogólne, jak i specjalistyczne stały się najważniejszym narzędziem wykorzystywanych przez użytkowników systemów informatycznych sieci. Drugą grupą narzędzi technologii Web 2.0 są rozwiązania oparte na mechanizmach *wiki*, będącymi specyficznymi stronami internetowymi wykorzystującymi przygotowane dla użytkownika oprogramowanie, dające mu możliwość wspólnej pracy przy tworzeniu określonych treści witryn. Pierwszą taką witryną była założona w 2001 r. Wikipedia. Tego typu serwisy są traktowane obecnie przez wiele firm jako narzędzie gromadzenia, powstawania i dystrybucji wiedzy. Kolejną kategorią instrumentów Web 2.0, są blogi – strony internetowe, na których użytkownicy dokonują wpisów, na interesujące ich tematy. Tworzone indywidualnie i instytucjonalnie (ogólnokorporacyjna platforma blogowa) wykorzystywane są do ulepszania współpracy wewnętrznej oraz wymiany informacji i wiedzy pomiędzy pracownikami oraz stają się znaczącym narzędziem prezentacji innowacyjności własnej firmy. Następne narzędzia należące do technologii Web 2.0 to podcasty (*podcasts*) oraz wideocasty (*videocasts*). Są to dźwiękowe lub dźwiękowo-obrazowe pliki, dostępne do ściągnięcia poprzez Internet (np. w postaci pliku mp3), odtwarzane następnie w komputerze użytkownika. Mogą być wykorzystane w celu rozszerzenia i stymulowania współpracy oraz wymiany wiedzy pomiędzy pracownikami. Modelowanie rzeczywistości najsilniej jest uwzględnione w tzw. światach wirtualnych (np. SecondLife), w którym użytkownicy, pod postacią sztucznych osobowości o określonych cechach – awatarów (*avatars*), mają możliwość „przeżywania” alternatywnego życia i prowadzenia realnej działalności gospodarczej. Jednym z najważniejszych narzędzi Web 2.0 są portale społecznościowe (*social networks, social network services*). Odgrywają one rolę społeczności internetowych skupiających użytkowników o wspólnych cechach, zainteresowaniach, hobby itp. występujących w ich życiu prywatnym i zawodowym. Podobnie jak poprzednie narzędzia, dają one internautom nowe możliwości komunikowania się, dzielenia się różnymi informacjami czy też materiałami zarówno tekstowymi, jak i zdjęciami czy materiałami wideo.

Z technicznego punktu widzenia ważnym elementem Web 2.0, są kanały RSS (*Really Simple Syndication*). Służą one do śledzenia określonego typu informacji w Internecie, a następnie dostarczania użytkownikowi nagłówków nowych wiadomości pojawiających się w określonych sekcjach witryn internetowych czy też blogach. Ułatwia to zbiorcze śledzenie istotnych dla użytkownika informacji, bez przeglądania tych witryn po raz kolejny. Zwłaszcza, że sam może on sobie stworzyć wzór interesujących go kanałów RSS. Ostatnim istotnym technologicznym narzędziem z grupy technologii Web 2.0 są sieci *peer-to-peer*, popularne już pod koniec ubiegłego wieku (*Napster*). Umożliwiają one wymianę różnego

typu plików – muzycznych, filmów, programów komputerowych itp. Nie są to jednak instrumenty istotne z punktu widzenia zarządzania organizacją.

Systemy oparte na rozwoju sieci opanowują więc nie tylko sferę ściśle ekonomiczną, lecz także społeczną, spełniając niejako postulaty tworzenia tzw. społeczeństwa informacyjnego i społeczeństwa wiedzy. Przenikanie się zaś tych sfer jest dziś już tak duże, że niemal niezauważalne, czemu sprzyja postęp w zastosowaniach mobilnych i komunikacyjnych.

Rozwój systemów opartych na sieci przedstawia rysunek 26.

<p>rozwiązania komercyjne dla dużych i średnich firm, sieciowość organizacyjna</p> <p>sieci komercyjne</p>	<p>rozwiązania całościowe i globalne dla wszystkich (organizacje, klient, społeczeństwo), sieciowość społeczna</p>
<p>rozwiązania korporacyjne oparte na standardach EDI, wielkie organizacje, sieciowość korporacyjna</p> <p>sieci prywatne</p>	<p>sieć Internet</p>

**Rysunek 26.** Etapy rozwoju systemów opartych na sieci

Źródło: opracowanie własne.

### 3.5. Całościowa integracja systemów na płaszczyźnie portali korporacyjnych

W ostatniej fazie rozwoju systemów zintegrowanych ERP (około 2008 r.) nastąpiło rozproszenie wysiłków zmierzających do pełnej, uniwersalnej integracji funkcjonalnej, ukierunkowanej wyłącznie na wewnętrzne procesy organizacji. Wraz z pojawieniem się równoległych niejako systemów internetowych o zupełnie innej technologii, a podobnych funkcjonalnościach z punktu widzenia użytkownika (konwergencja), tradycyjne systemy punktowe ERP musiały

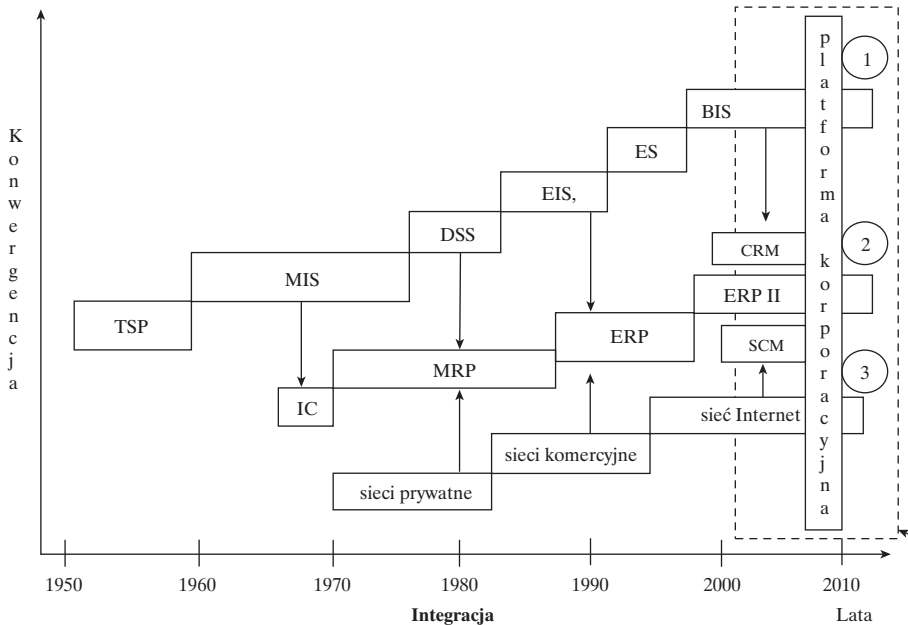
odpowiedzieć bądź rozszerzeniem do granic absurdu wielkości systemu w celu skupienia w nim obsługi wszystkich potencjalnie możliwych zadań, bądź otwarciem go na zewnątrz i dążeniem do stworzenia nowej płaszczyzny w przestrzeni zewnętrznej. Ich rozwój umożliwił swobodną wymianę danych i informacji między dostawcami i odbiorcami dysponującymi systemami tej klasy w oparciu o wykorzystanie np. przeglądarki internetowej. Co więcej, w systemach B2B zaczęto coraz częściej i chętniej korzystać zarówno z mechanizmów wypracowanych i przetestowanych w Internecie, tworząc wewnętrzne intranety gospodarcze, jak i ekstranetów – rozwiązań sieciowych polegających na połączeniu intranetów za pomocą protokołów sieciowych. Celem intranetu jest udostępnienie zasobów wewnątrz organizacji. Celem ekstranetów jest udostępnienie własnych zasobów między organizacjami lub między nimi i ich klientami, przy braku możliwości powszechnego dostępu z globalnej sieci Internet.

I tu widać przejawy upodabniania się funkcji aplikacji oprogramowania sieciowych systemów prywatnych i komercyjnych (interorganizacyjnych) do systemów internetowych. Oczywiście nie bez znaczenia jest fakt, że budowane w ten sposób systemy są znacznie tańsze od budowanych w sposób tradycyjny.

Rozwiązaniem, które w warunkach rozwoju systemów internetowych zaczęto stosować w miejsce integracji wewnątrzorganizacyjnej (dokonywanej poprzez dane, przez interfejs użytkownika oraz wewnętrzny interfejs aplikacji) było zastosowanie integracji zewnętrznej poprzez portale korporacyjne. Portal korporacyjny to „...platforma informatyczna, która integruje systemy i technologie informatyczne, dane, informację i wiedzę funkcjonujące w organizacji oraz jej otoczeniu, w celu umożliwienia użytkownikom spersonalizowanego i wygodnego dostępu do danych, informacji i wiedzy (oraz ich źródeł), stosownie do wynikających z ich zadań potrzeb, w dowolnym czasie i miejscu, w bezpieczny sposób i poprzez zunifikowany (jednolity) interfejs WWW...” (Ziomba, 2007, s. 344–356). Zasadniczą cechą platform korporacyjnych jest więc integracja danych pochodzących ze źródeł wewnętrznych, z danymi zewnętrznymi, ich konwersja na wspólne i dające się wspólnie przetwarzać formaty; integracja niejednorodnych aplikacji; integracja komunikacji pomiędzy poszczególnymi użytkownikami i zapewnienie im spersonalizowanej informacji i wiedzy (Ziomba, 2009).

Pojawienie się portali korporacyjnych związane jest z rozwojem technologii sieci internetowych, a funkcjonują one na ogół w środowisku intranetowym organizacji. Poprzez to środowisko – interfejs www – są dystrybuowane użytkownikom jako niezbędna mu informacje i wiedza.

Odnosi się wrażenie, że platforma korporacyjna jako instrument integracji pełni jednocześnie funkcję narzędzia konwergencji – na płaszczyźnie tej możliwe jest współdziałanie nie tylko uzupełniających się systemów, lecz także takich, które można wykorzystywać równolegle.



Legenda:

- 1) ścieżka rozwoju poprzez komplikację architektury logicznej,
- 2) ścieżka rozwoju poprzez integrację funkcjonalną,
- 3) ścieżka rozwoju poprzez rozszerzenia sieciowe.

Rodzaje systemów: TPS – systemy przetwarzania danych, MIS – systemy informacyjne zarządzania, DSS – systemy wspomagające zarządzanie, EIS – systemy informacyjne kierownictwa, ESS – systemy wspomaganie kierownictwa, ES – systemy eksperckie, BIS – systemy inteligencji biznesowej, IC – system zarządzania magazynem, MRP – systemy planowania zapotrzebowania materiałowego, MRP II – systemy planowania zasobów produkcyjnych, ERP – systemy planowania zasobów na potrzeby przedsiębiorstwa, ERP II – elektroniczne systemy planowania zasobów na potrzeby przedsiębiorstwa, CRM – zarządzanie relacjami z klientem, SCM – zarządzanie łańcuchem dostaw.

**Rysunek 27.** Ścieżki rozwoju systemów informatycznych wspomagających zarządzanie

Źródło: opracowanie własne.

Odwierciedlenie opisanych procesów zawiera rysunek 27. Dla jaśniejszego przedstawienia analizowanego zagadnienia ograniczono się w nim do ilustracji zasadniczych wśród omówionych tendencji występujących w rozwoju systemów informatycznych zarządzania. Nie zostały na przykład odzwierciedlone naturalne niemal skłonności do wewnętrznego w ramach pierwszej ścieżki rozwoju scalania i łączenia się systemów. Uznano, że tendencja ta była niejako spotęgowaniem, uprzednio szczegółowo rozpatrzonej komplikacji struktury logicznej architektury w poszczególnych typach systemów i nie wymaga dalszego pogłębiania tematu.

Nie odzwierciedlono też szczegółowo rozwoju poszczególnych narzędzi internetowych zakładając, że są one nadal w fazie intensywnego rozwoju. Niemniej jednak wyraźnie widoczna jest – możliwa dzięki platformie korporacyjnej – tendencja łączenia wszystkiego ze wszystkim (integracja wielopłaszczyznowa) w warunkach przenikania się idei współdziałania różnych systemów informatycznych na wszystkich wyszczególnionych ścieżkach ich rozwoju.

## Zakończenie

---

Obecnie zakłada się, że punkt ciężkości w rozumieniu pojęcia systemów informatycznych wspomagających zarządzanie przesuwa się od prezentacji tego, co zapewniają (i czego nie da się przeważnie zmienić, czego konsekwencją jest konieczność dostosowania użytkownika do systemu) do idei zbudowania systemu rozwiązującego realnie istniejące problemy i podpowiadającego użytkownikowi, jak ma w określonej sytuacji postąpić.

Sprowadza się to do wyartykułowania, z jakich komponentów system ma się składać, jaki wzorzec rozwiązania problemu (najlepszych praktyk zarządzania) będzie w danej sytuacji najbardziej użyteczny, a w jaki sposób rozwiązanie problemu w danym przypadku będzie w ogóle możliwe. Proces ewolucji takiego myślenia, wynikający z jednej strony z rozwoju metod zarządzania projektami wpływających na systemy informatyczne, a z drugiej – z rozwoju systemów informatycznych rzutujących na kierunki unowocześnienia metod projektowania został przedstawiony w punktach poniżej.

1. We wczesnych latach siedemdziesiątych systemy informatyczne wspomagające zarządzanie były prezentowane jako oparte na komputerze systemy wspomagające pośrednio podejmowanie decyzji na podstawie raportów, doświadczeń i wyszkolenia decydentów. Koncentrowały się na rozwiązywaniu problemów pojedynczego decydenta. Punktem startowym do ich tworzenia stało się zastosowanie technologii informacyjnej dla lepszego (szybszego, dokładniejszego) uzyskiwania zarejestrowanych w systemie informacji na potrzeby użytkownika. Ze względu na stosunkowo prostą logikę większości tworzonych systemów nie występowała potrzeba stosowania metod wychodzących poza dotychczasowe doświadczenia wynikające z zarządzania organizacją (na tej bazie powstały metody strukturalne zarządzania).
2. W latach późniejszych określano systemy informatyczne wspomagające zarządzanie jako interakcyjne systemy komputerowe, które pozwalają decydentowi stosować bazy danych i bazy modeli do rozwiązywania tzw. problemów nie tylko deterministycznych, lecz także słabo ustrukturalizowanych. Wspomaganie dotyczyło nie samego procesu decyzyjnego, a pomocy w indywidualnym przetwarzaniu danych za pomocą narzędzi programistycznych pozwalających w szybki sposób określić kierunki rozwoju lub np. ustalić plany finansowe



- organizacji. Zmiana paradygmatu wykorzystania systemów informatycznych w kierunku możliwości modelowania decyzji i podpowiadania jej użytkownikowi spowodowała poszukiwania odpowiadających jej metod projektowania grup metodyk systemów: operacyjnej, obiektowej czy społecznej.
3. W drugiej połowie lat osiemdziesiątych systemy informacyjne wspomagające zarządzanie traktowano w kategoriach zwiększenia efektywności zarówno operacyjnej, jak i związanej z podejmowaniem decyzji zarządczych (taktycznej i strategicznej). Nowe wyzwania łączące się z powstaniem idei systemów informowania i wspomaganie kierownictwa oraz systemów eksperckich spowodowały dalszy rozwój koncepcji zarządzania projektami informatycznymi w kierunku metod nazywanych potem zwinnymi (agile), szybkimi, nowoczesnymi.
  4. Początek lat dziewięćdziesiątych stworzył nowe technologiczne możliwości informatyzacji organizacji. Z jednej strony dominująca na rynku pozycja powielarnych systemów zintegrowanych powodowała poszukiwanie udoskonalonych metod ich budowy i wdrażania. Z drugiej zaś – rozwój systemów ukierunkowany został na tworzenie inteligentnego środowiska pracy na wszystkich szczeblach zarządzania. Coraz częściej pod etykietą systemu informatycznego rozumie się system zdolny wykorzystywać odzwierciedlone w nim zdolności intelektualne (wiedzę) dla doskonalenia kreatywności w podejmowaniu decyzji. Dotychczasowe doświadczenia dotyczące zarówno tendencji rozwojowych systemów, jak i metod projektowania zaczynają przynosić rezultaty nie tylko w postaci nowych klas metod projektowania systemów, lecz także komercyjnych pakietów łączących doświadczenia organizacyjne, projektowe, informatyczne itp.
  5. Wraz z początkiem nowego wieku punktowo (w sensie lokalizacji wewnątrz organizacji) traktowana inteligencja systemu zaczęła się rozprzestrzeniać w kierunku rozszerzenia relacji z kontrahentami poprzez kanały tradycyjne i elektroniczne. Jak następuje pełna integracja systemów różnych typów, gospodarki tradycyjnej i elektronicznej, tak konieczna staje się integracja metodyk projektowania wszystkich tych sfer.

Na problem metod powstania i ewolucji metod, jak i tworzenia systemów można patrzeć z punktu widzenia rozwoju technologii informacyjnych (zwłaszcza komponentów systemów) oraz rozwoju technik modelowania wykorzystywanych w zarządzaniu procesami i projektami. Powstanie systemu informatycznego jest w tym kontekście naturalną niejako konsekwencją spotkania się ze sobą wymogów określonych charakterystyk tworzonych systemów oraz zmian sposobów ich uzyskania. Trzecim łączącym te tendencje czynnikiem jest powstanie bliskiej i szybkiej technologii interakcji pomiędzy twórcami a odbiorcami projektów (w sensie punktowym i przestrzennym – Internet).

W tym kontekście fazy ewolucji technologii informacyjnych można ująć w sposób następujący<sup>1</sup>:

- *faza pierwotnego przetwarzania danych* – charakteryzowała się wykorzystaniem komputera do rozwiązywania pojedynczych zadań, głównie w sposób transakcyjny, gdzie każdy program przetwarzał podporządkowane sobie własne zbiory; operowanie na danych było ograniczone do bardzo prostych operacji typu: sortowanie, klasyfikowanie, sumowanie itp.; modelowanie pojawiało się jedynie w postaci pojedynczych równań bądź grupy równań pośród wierszy programu; rezultatem przetworzenia danych były sumarycznie ujęte wyniki, najczęściej w postaci tabelarycznej;
- *faza systemów zarządzania zbiorami i modeli symbolicznych* – polegała na wykorzystaniu komputera do przetwarzania zadań zintegrowanych z wykonywanymi funkcjami, niekiedy udostępniającymi te same zbiory różnym programom; na tym etapie wysiłki projektantów i twórców systemów koncentrowały się na rozwoju wspólnego oprogramowania dla tworzonych zbiorów i zapewnieniu elementarnych procedur zabezpieczenia oraz integracji danych; towarzyszyło temu proste modelowanie charakteryzujące się rozwiązywaniem, niekiedy w dużej ilości, zbiorów równań liniowych i nieliniowych; wynikiem przetworzenia danych były tu raporty otrzymywane po każdorazowej sesji pracy komputera;
- *faza przetwarzanie baz danych* – szczególną cechą tego etapu było oddzielenie danych od programów, które dane te przetwarzały; tworzone systemy gwarantowały redukcję oprogramowania niezbędnego dla modyfikacji danych, a także wykorzystywały modele danych odwzorowujących relacje pomiędzy danymi; z matematycznego punktu widzenia modelowanie nadal jest proste – dostosowane jednak do nowych warunków stworzonych przez funkcjonowanie baz danych, szczególnie w relacyjnym modelu danych; rezultatem działania systemu były raporty o z góry zagwarantowanej strukturze;
- *faza systemów języków zapytań* – była następnym krokiem w rozwoju metod przetwarzania danych w technologiach informacyjnych; podczas ich konstruowania główny punkt ciężkości został położony na zagadnienie kontaktu z użytkownikiem, zwłaszcza nieprofesjonalnym, zapewniając mu bezpośredni dostęp do zbiorów danych poprzez dialog lub specjalnie konstruowany język użytkownika; wynikiem przetworzenia danych były tu raporty generowane na żądanie użytkownika, o kształcie i strukturze, która najbardziej mu odpowiadała;
- *faza systemów baz danych hybrydowych* – faza ta charakteryzowała się, oprócz obsługi bazy danych, obsługą bazy modeli, gdzie występowały duże ilości

<sup>1</sup> O potrzebie wspólnej ewolucji systemów przetwarzania danych i modeli pisał już: Spraque, 1987.

parametrów i współczynników modelowych o charakterze, postaci i strukturze różnej od danych źródłowych przechowywanych w bazie danych; w związku z tym obsługa ich przechodziła przez szereg etapów związanych z różnymi fazami modelowania matematycznego:

- *modele obliczeniowe* – komputery na tym etapie stały się istotne dla modelowania ze względu na potencjalne możliwości redukcji wielkich ilości danych podczas estymacji współczynników równań lub rozwiązywania skomplikowanych układów równań;
  - *modele komputerowe* – etap ten obejmuje okres, od którego program zamiast służyć głównie do rozwiązania problemu staje się sam modelem; zmienne komputerowe stają się symbolami, którymi manipuluje się poprzez program (a nie poprzez równania połączone operacjami matematycznymi); to podejście prowadzi do sytuacji, w której problem nie jest w zasadzie rozwiązywany, lecz program jest uruchamiany po to, by obserwować zachowanie modelu w celu rozpoznania modelowanej sytuacji (symulacja, wariantowanie);
  - *systemy baz modeli* – dla rozwiązania szczególnych, a w pewnym sensie zalgorytmizowanych problemów zaczęto tworzyć takie systemy informatyczne, które ze swej istoty mogły być traktowane jako system modeli; oprogramowanie zapewniało jedynie wspólny format danych wejściowych, podobne formaty generowanych raportów i zintegrowaną dokumentację, natomiast system taki mógł być wykorzystywany przez każdego użytkownika indywidualnie lub był specjalizowany na poszczególne stanowiska robocze w organizacji;
  - *modele interakcyjne* – wykorzystanie ich stało się możliwe, gdy pojawiły się m.in. i mikrokomputery wyposażone w biblioteki modeli, które mogą być zastosowane do różnego rodzaju analiz; niestety wspólną cechą tych modeli jest uruchamianie pojedynczych programów, dla których są wymagane na ogół różne pod względem merytoryki i struktury dane wejściowe, ale rzadko kiedy możliwy jest jakikolwiek związek logiczny czy formalny pomiędzy realizowanymi programami;
- *faza systemów ekstrakcji danych (data mining) z hurtowni danych* – to jeden z etapów procesu odkrywania wiedzy, polegający na wykorzystaniu oprogramowania do znajdowania ukrytych dla człowieka prawidłowości i zależności pomiędzy danymi w celu pozyskania na tej podstawie wiedzy mogącej służyć za podstawę podejmowania decyzji; wspierane przez *systemy interakcyjnych baz modeli (Business Analytics)* – umożliwiające dowolne analizy matematyczne, statystyczne, ekonometryczne na danych z hurtowni danych i uzupełniających danych wewnątrzpakietowych, w tym modele zarządzania wiedzą.

Generalnie, można powiedzieć, że podejście do technologii informacyjnych ewoluowało od tradycyjnego przetwarzania danych i prostych modeli matematycznych w kierunku charakteryzującym się:

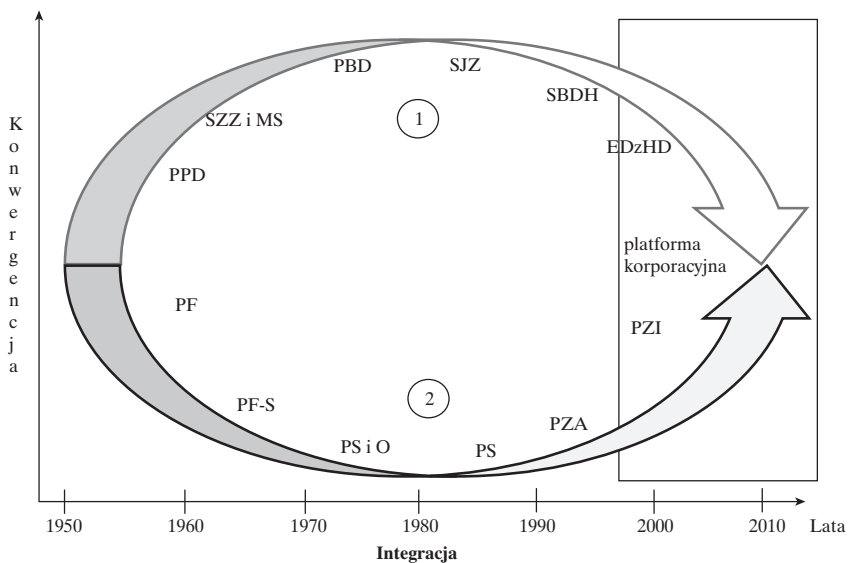
- efektywnym zarządzaniem dużą liczbą danych,
- wprowadzeniem elastycznego systemu łatwego dostępu do danych dla osób niebędących programistami,
- komplikacją i rosnącym wyrafinowaniem modeli matematycznych od modeli symbolicznych, poprzez bazy modeli i interaktywne bazy modeli, przy malejącym stopniu komplikacji posługiwania się nimi,
- wykrywaniem zależności pomiędzy danymi w celu transformacji ich w wiedzę i pozyskania tej wiedzy dla użytkownika.

Aby być zgodnym z postawioną tezą, tak pojętym fazom ewolucji technologii informacyjnych powinny odpowiadać fazy rozwoju metodyk zarządzania projektami w ogóle, a zarządzania projektami informatycznymi w szczególności.

Fazy rozwoju metodyk zarządzania projektami informatycznymi kształtowały się zaś następująco:

- *faza projektowania funkcjonalnego* – opracowywano wrywkowo traktowane projekty odseparowanych, prostych programów i systemów informatycznych obejmujących wyróżnione pod kątem uciążliwości masowych, rutynowych obliczeń lub trudności algorytmicznych;
- *faza projektowania funkcjonalno-strukturalnego i strukturalnego* – zdeterminowana najpierw technologią sekwencyjnego przetwarzania danych, a następnie baz danych, głównie hierarchicznych i relacyjnych; umożliwiona tym postępująca na płaszczyźnie technologicznej, informacyjnej i funkcjonalnej integracja do dnia dzisiejszego wykorzystuje dorobek tej szkoły w implementacjach powielarnych systemów zintegrowanych;
- *faza projektowania obiektowego* – nastąpiła wraz z erą rozwoju metod modelowania matematycznego na potrzeby podejmowania decyzji – trzeba było zmienić podejście do metodyk projektowania, co związane było z jednej strony, z ogromną ilością danych do przetworzenia i relacji pomiędzy nimi, a z drugiej – zastosowaniem coraz bardziej wyrafinowanych modeli matematycznych używanych w procesie przetwarzania decyzji;
- *faza projektowania społecznego* – gdy okazało się, że tworzone i wdrażane systemy informatyczne napotykały głównie trudności niezwiązane z przeszkodami technologicznymi i organizacyjnymi, a z zarządzaniem potencjałem społeczny firmy;
- *faza projektowania zwinnego (agile)* – gdy pojawiła się konieczność konstruowania systemów szybko, taniej, bardziej związanych z wymaganiami użytkownika, w sferze już nie tylko tradycyjnie wewnątrz przedsiębiorstwa, lecz także na styku przedsiębiorstwo-otoczenie ekonomiczne, a także w sfe-

- rze elektronicznej, opartego na szeroko pojętym modelowaniu procesów i rozwiązań z tym związanych;
- *faza projektowania zintegrowanego* – gdy pojawiają się naukowe i komercyjne pakiety metodyk zarządzania projektami informatycznymi stanowiące konglomerat metod zarządzania projektami i procesami, najlepszych wzorców praktyk zarządzania, metod organizacyjnych, socjologii i psychologii organizacji itp.; ich masowe zastosowanie umożliwi lepszą obsługę procesów integracyjnych współczesnego oprogramowania systemów informatycznych.



#### Legenda

##### 1. Fazy rozwoju technologii informacyjnych:

- PPD – pierwotne przetwarzania danych,
- SZZiMS – systemy zarządzania zbiorami i modele symboliczne,
- PBD – przetwarzanie baz danych,
- SJZ – systemy języków zapytań,
- SBDH – systemy baz danych hybrydowych,
- EDzHD – ekstrakcja danych z hurtowni danych.

##### 2. Fazy projektowania:

- PF – projektowanie funkcjonalne,
- PF-S – projektowanie funkcjonalno-strukturalne i strukturalne,
- PS i O – projektowanie obiektowe,
- PS – projektowanie społeczne,
- PZA – projektowania zwinnego (agile),
- PZI – projektowania zintegrowanego.

**Rysunek 28.** Ewolucja technologii informacyjnych ze względu na zmiany metod zarządzania projektami systemów informatycznych

Źródło: opracowanie własne.

Zarówno ostatni etap ewolucji technologii informacyjnych, jak i ostatni etap ewolucji zarządzania projektami mają jedną cechę wspólną – następuje konwergencja w punkcie zbliżenia do końcowego, nieprofesjonalnego użytkownika poprzez rozbudowany system komunikacyjny na platformie korporacyjnej. Jest to prawdopodobnie podstawowa płaszczyzna, na której mogą i powinny rozwijać się obecnie systemy informatyczne.

Można mieć jednak wątpliwości czy istnieje świadomość powiązania zmian metod zarządzania projektami informatycznymi z rozwojem systemów informatycznych wspomagających zarządzanie. Z obserwacji dotychczasowych tendencji rozwojowych w tym zakresie nasuwają się następujące wnioski:

- początkowo, metodyki projektowania umożliwiały przyspieszenie rozwoju systemów informatycznych i były dostosowane do aktualnych potrzeb tworzenia systemów informatycznych (funkcjonalne i funkcjonalno-strukturalne); analizy dotyczyły wyszczególnionych funkcji organizacji, projekt – jednostkowych, przeważnie algorytmicznie prostych operacji pomiędzy danymi, opartych na modelach symbolicznych;
- z chwilą rozwoju technologii informacyjnych (pamięci o dostępie bezpośrednim i bazy danych) musiała nastąpić zmiana sposobu analizy i projektowania na metody uwzględniające te zmiany; najpierw powstały i zaczęły się rozwijać metody strukturalne, będące rozwinięciem metod analizy systemowej organizacji oraz badań operacyjnych, naturalnie łączących się z wiedzą, doświadczeniami i umiejętnościami projektantów; następnie – wraz z rozwojem potrzeb wynikających z rozwoju języków programowania – powstała grupa metod obiektowych zaspokajających te wymagania; rywalizacja pomiędzy tymi szkołami przyspieszyła rozwój systemów bardziej wyrafinowanych niż systemy oparte tylko na bazach danych – pierwszych systemów wspomagania podejmowania decyzji i systemów eksperckich;
- problemy człowieka z komunikacją z systemami wspomagania decyzji oraz systemami eksperckimi spowodowały rozwinięcie systemów opartych na czynnikach społecznych – pierwsze metody nastawione na potrzeby klienta – przyspieszyło to też prace nad systemami informowania i wspomagania kierownictwa, posiadające znacznie bardziej rozwinięte mechanizmy komunikacji kierownictwa wyższych szczebli z systemem, metody wizualizacji oraz łączności ze światem zewnętrznym;
- usprawnienie komunikacji umożliwiło dalszy rozwój systemów opartych nie tylko na bazach danych, lecz także na bazach modeli i bazach wiedzy; po prywatnych, rozwijały się sieci komercyjne, powstawał Internet i zaistniała potrzeba łączności z nimi, ale również budowy systemów spełniających funkcje komercyjne w samych sieciach; do tego nie wystarczyły stare, tradycyjne, ociężałe (bo długo, choć na ogół skutecznie) działające metodyki projektowania; zostały więc zastąpione metodykami lżejszego kalibru (zwinnymi),

- pozwalającymi na projektowanie szybsze, dokładniej uwzględniające wymagania użytkownika, o mniejszym ryzyku realizacji;
- postęp technologiczny i rosnące w tempie wykładniczym potrzeby informacyjne systemów spowodowały, że konieczne stało się tworzenie oprogramowania wielopłaszczyznowego – z jednej strony zapewniającego obsługę hurtowni danych, w sensie ekstrakcji danych w pożądanym układzie, z drugiej – łączenie danych pochodzących z wnętrza organizacji z danymi zewnętrznymi z Internetu i konwersji na format, umożliwiający ich wspólne użytkowanie i wykorzystanie w procesie podejmowania decyzji; niezbędne do tego stało się zastosowanie komercyjnych metodyk projektowania uwzględniających zróżnicowane lecz zintegrowane wewnętrznie metody i techniki zarządzania projektami.

Przedstawione powyżej wnioski z dotychczasowych analiz mają charakter wstępny i w miarę postępu badań będą rozwijane. Autor wyraża nadzieję, że cel pracy, zarysowany we wstępie, został chociaż częściowo zrealizowany, a zaprezentowane zestawienie jego przemyśleń i opinii poszerza w pewien sposób literaturę tematu. Należy jednak również zauważyć, że przedstawione opracowanie porusza wiele istotnych problemów badawczych oraz proponuje sposoby ich rozwiązania, które – można żywić przynajmniej taką nadzieję – rozwijane będą przez następców i doczekają się wnikliwej analizy na następnym szczeblu rozwoju społeczeństwa informacji i wiedzy w Polsce.

# Literatura

---

- Abt, S. (1998). *Zarządzanie logistyczne w przedsiębiorstwie*. Warszawa: PWE.
- Adamczyk, A. i Chmielarz, W. (2005). *Zintegrowane systemy informatycznego wspomagania zarządzania*. Warszawa: Wydawnictwo Naukowe WSEI.
- Awad, M.A. (2005). *A Comparison between Agile and Traditional Software Development Methodologies*. The University of Western Australia. Pozyskano z: <http://www.scribd.com/doc/55475190/A-ion-Between-Agile-and-Traditional-SW-Development-Methodologies> (styczeń 2012).
- Barney, M. i McCarty, T. (2005). *Nowa Six Sigma*. Gliwice: Wydawnictwo Helion.
- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Boston, MA: Addison-Wesley.
- Beck, K. i Ander C. (2004). *Extreme Programming Explained: Embrace Change*. Addison-Wesley, Second Edition.
- Beck, K. i in. (2001). *Manifesto for Agile Software Development*. Agile Alliance. Pozyskano z: <http://www.pmbriefcase.com/methodologies/50-software-development/55-agile-software> (luty 2013).
- Boehm, B. (1981). *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice Hall.
- Booch, G. (1994). *Object-Oriented Analysis and Design with Applications*. Redwood City, CA: Benjamin/Cummings.
- Bralczyk, J. (red.) (2008). *Słownik 100 tysięcy potrzebnych słów*. Warszawa: Wydawnictwo Naukowe PWN.
- Bukłaha, E. (2010). Zarządzanie projektami informatycznymi. W: J. Zawila-Niedźwiecki, K. Rostek i A. Gąsioriewicz (red.), *Informatyka gospodarcza*, t. II. Warszawa: Wydawnictwo C.H. Beck.
- Burlton, R. (2001). *Business Process Management: Profiting From Process*. Indianapolis: SAMS Publishing.
- Chang, L.J. i Powell, P. (1998). Towards a Framework for Business Process Re-engineering in Small and Medium-sized Enterprises. *Information Systems Journal*, 3.
- Checkland, P.B. (2000). Soft Systems Methodology: A Thirty Year Retrospective. *Systems Research and Behavioral Science*, 17, 11–58.
- Chmielarz, W. (2000). *Zagadnienia analizy i projektowania systemów informatycznych wspomagających zarządzanie*. Warszawa: Wydawnictwo Naukowe WZ UW.
- Chmielarz, W. (2005). *Selected Problems of IT Development*. Warszawa: Wydawnictwo Naukowe WZ UW.
- Chmielarz, W. (2005a). *Systemy elektronicznej bankowości*. Warszawa: Difin.
- Chmielarz, W. (2007). *Systemy elektronicznego biznesu*. Warszawa: Difin.
- Chmielarz, W. (2010). Projektowanie systemów informatycznych. W: J. Zawila-Niedźwiedzki, K. Rostek i A. Gąsioriewicz (red.), *Informatyka gospodarcza*, t. 1, s. 359–402, Warszawa: Wydawnictwo C.H. Beck.



- Chmielarz, W. i Klineciewicz, K. (2010). Zarządzanie projektami. W: J. Bogdanienko (red.), *Organizacja i zarządzanie w zarysie*, s. 238–252. Warszawa: Wydawnictwo Naukowe WZ UW, Dom Wydawniczy Elipsa.
- Coad, P. i Yourdon, E. (1994). *Analiza obiektowa*. Warszawa: Oficyna Wydawnicza Read Me.
- Coad, P. i Yourdon, E. (1994a). *Projektowanie obiektowe*. Warszawa: Oficyna Wydawnicza Read Me.
- Cohen, I., Mandelbaum, A. i Shub A. (2004). Multi-project Scheduling and Control: A Process-based Comparative Study of the Critical Chain Methodology and Some alternatives. *Project Management Journal*, 35(2).
- Coleman, D. i in. (red.) (1995). *Object-Oriented Development: The Fusion Method in the Real World*. Englewood Cliffs, N.J.: Prentice Hall.
- Czarnacka-Chrobot, B. (2009). *Wymiarowanie funkcjonalne przedsięwzięć rozwoju systemów oprogramowania wspomagającego zarządzanie*. Warszawa: Oficyna Wydawnicza SGH.
- DeMarco, T. (1978). *Structured Analysis and Design*. New York: Yourdon Inc.
- DeMarco, T. (2002). *Zdążyć przed terminem – opowieść o zarządzaniu projektami*. Warszawa: Studio Emka.
- DMAIC (2012). *The Six Sigma Toolkit*. Pozyskano z: [http://www.dmaictools.com/?utm\\_expid=18088366-3](http://www.dmaictools.com/?utm_expid=18088366-3) (grudzień 2012).
- Dumnicki, R., Kasprzyk A. i Kozłowski A. (1998): *Analiza i projektowanie obiektowe*. Gliwice: Helion.
- Flakiewicz, W. (2002). *Systemy informacyjne w zarządzaniu*. Warszawa: Wydawnictwo C.H. Beck.
- Flasiński, M. (2006). *Zarządzanie projektami informatycznymi*. Warszawa: Wydawnictwo Naukowe PWN.
- Flasiński, M. (2009). *Zarządzanie projektami informatycznymi*. Warszawa: Wydawnictwo Naukowe PWN.
- Fox, S. i Madden M. (2006). *Riding the Waves of “Web 2.0”*. W: Pew Internet & American Life Project, 5 października. Pozyskano z: [http://www.pewinternet.org/pdfs/PIP\\_Web\\_2.0.pdf](http://www.pewinternet.org/pdfs/PIP_Web_2.0.pdf). (luty 2013).
- Freyenfeld, W.A. (1984). *Decision Support Systems*, Manchester: NCC Publications.
- Fuglewicz, P., Stapor, K. i Trojnar A. (1995). *CASE dla ludzi*, seria *Inżynieria oprogramowania*. Warszawa: Wydawnictwo Lupus.
- Galant-Pater, M. (2009). *Przyczyny porażek i sukcesów informatyzacji biznesu w świetle badań empirycznych*. Pozyskano z: [http://www.ptzp.org.pl/files/konferencje/kzz/artyk\\_pdf\\_2009/035\\_Galant-Pater.pdf](http://www.ptzp.org.pl/files/konferencje/kzz/artyk_pdf_2009/035_Galant-Pater.pdf) (kwiecień 2012).
- Gibson, M., Hughes, G. (1994). *Systems Analysis and Design. A Comprehensive Methodology with CASE*. Boyd&Fraser.
- Goldratt, E.M. (2000). *Łańcuch krytyczny*. Warszawa: Wydawnictwo Werbel.
- Goldratt, E.M. i Cox, J. (2000). *Cel*. Warszawa: Wydawnictwo Werbel.
- Gołuchowski, J. (2007). *Technologie informatyczne w zarządzaniu wiedzą w organizacji*, wyd. II. Katowice: Wydawnictwo Akademii Ekonomicznej w Katowicach.
- Hammer, M. i Herschmann L. (2010). *Faster Cheaper Better: The 9 Levers for Transforming How Work Gets Done*. New York: Crown Business.
- Hanna, A., Windebank, J., Adams, S., Sowerby, J., Rance, S. i Cartlidge, A. (2009). *ITIL® Foundation Handbook. ITILV3. TSO (The Stationery Office)*.

- Harmon, P. (2007). *Business Process Change, Second Edition: A Guide for Business Managers and BPM and Six Sigma Professionals*. Morgan Kaufmann Publishers.
- Harmon, P. (2011). *Business Process Trends Advisor*, 9(16).
- Hellwig, Z. (red.) (1971). *Automatyczne przetwarzanie Informacji*. Warszawa: PWE.
- Hellwig, Z. (red.) (1975). *Maszyny cyfrowe i ich zastosowanie*. Warszawa: PWE.
- Highsmith, J. (1999). *Adaptive Software Development*. New York: Dorset House Publishing.
- Highsmith, J. (2004). *Agile Project Management: Creating Innovative Products*. Addison-Wesley.
- <http://itilservicestrategy.blogspot.com/> (grudzień 2012).
- <http://itsm.itlife.pl/content/view/10005/50> (styczeń 2013).
- <http://itsm.itlife.pl/content/view/10037/82/> (luty 2013).
- [http://mfiles.pl/pl/index.php/Og%C3%B3lna\\_charakterystyka\\_metodyki\\_Prince\\_2](http://mfiles.pl/pl/index.php/Og%C3%B3lna_charakterystyka_metodyki_Prince_2) (maj 2012).
- <http://office.microsoft.com/pl-pl/project-help/krotka-historia-zarzadzania-projektami-HA001135342.aspx> (maj 2012).
- [http://pl.wikipedia.org/wiki/Information\\_Technology\\_Infrastructure\\_Library](http://pl.wikipedia.org/wiki/Information_Technology_Infrastructure_Library) (styczeń 2013).
- [http://pl.wikipedia.org/wiki/Information\\_Technology\\_Infrastructure\\_Library](http://pl.wikipedia.org/wiki/Information_Technology_Infrastructure_Library) (grudzień 2012).
- <http://pubs.opengroup.org/architecture/togaf9-doc/arch/> (luty 2013).
- <http://www.15000.net/>; <http://www.itlife.pl/> (listopad 2012).
- <http://www.15000.net/>; <http://www.itlife.pl/> (luty 2013).
- [http://www.4pm.pl/artykul/czym\\_jest\\_six\\_sigma-26-1462.html](http://www.4pm.pl/artykul/czym_jest_six_sigma-26-1462.html) (luty 2013).
- [http://www.4pm.pl/artykul/six\\_sigma\\_czesc\\_4-26-27-2.html](http://www.4pm.pl/artykul/six_sigma_czesc_4-26-27-2.html) (styczeń 2013).
- [http://www.4pm.pl/artykul/six\\_sigma\\_czesc\\_5-26-56-4.html](http://www.4pm.pl/artykul/six_sigma_czesc_5-26-56-4.html) (listopad 2012).
- [http://www.best-management-practice.com/gempdf/PRINCE2\\_2009\\_Overview\\_Brochure\\_June2011.pdf](http://www.best-management-practice.com/gempdf/PRINCE2_2009_Overview_Brochure_June2011.pdf) (styczeń 2013).
- <http://www.best-management-practice.com/Knowledge-Centre/An-Interview-with/?DI=571463> (luty 2013).
- <http://www.best-management-practice.com/Knowledge-Centre/Publication-Reviews/PRINCE2/?DI=622360> (luty 2013).
- <http://www.best-management-practice.com/Publications-Library/IT-Service-Management-ITIL/ITIL-2011-Edition/> (styczeń 2012).
- <http://www.controlchaos.com/storage/S3D%20First%20Chapter.pdf> (listopad, 2012).
- <http://www.controlchaos.com/storage/S3D%20First%20Chapter.pdf> (listopad, 2012).
- <http://www.controlchaos.com/storage/S3D%20First%20Chapter.pdf> (luty 2013), za: Schwaber, K. i Sutherland, J. (2012). *Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust*. NY: Wiley&Sons.
- <http://www.controlchaos.com/storage/S3D%20First%20Chapter.pdf> (luty 2013).
- <http://www.itil.org/en/vomkennen/itil/index.php> (luty 2013).
- <http://www.itil.org/en/vomkennen/itil/index.php> (wrzesień 2012).
- <http://www.itil-officialsite.com/AboutITIL/WhatisITIL.asp> (listopad 2012).
- [http://www.itlibrary.org/index.php?page=Capacity\\_Management](http://www.itlibrary.org/index.php?page=Capacity_Management) (luty 2013).
- <http://www.pmi.org/About-Us.aspx> (luty 2013).
- <http://www.prince-officialsite.com/> (styczeń 2013).

- <http://www.silencefails.com/downloads/SilenceFailsFullReport.pdf> (luty 2013).
- <http://www.statsoft.pl/sixsigma.html> (styczeń, 2013).
- <http://www.zarządzanie-projektami-it.pl/sukces-projektu/> (marzec, 2012).
- <http://zarządzanie-projektami-it.pl/sukces-projektu/> (marzec, 2012).
- Jackson, M. (1983). *System Development*. Engelwood Cliffs: Prentice Hall.
- Jacobson, I., Christerson, M., Jonsson, P. i Overgaard G. (1996). *Object-Oriented Software Engineering*. Harlow: Addison-Wesley.
- Januszewski, A. (2008). *Funkcjonalność informatycznych systemów zarządzania*, t. I i II. Warszawa: Wydawnictwo Naukowe PWN.
- Jaszkiewicz, A. (1997). *Inżynieria oprogramowania*. Gliwice: Helion.
- Johnson, J. (2005). *CHAOS Rising, Standish Group*. Referat wygłoszony na: II Krajowej Konferencji Jakości Systemów Informatycznych, Computerworld, czerwiec.
- Kasperek, M. (2011). *Zarządzanie projektem*. Katowice: Wydawnictwo Uniwersytetu Ekonomicznego w Katowicach.
- Kerzner, H. (2005). *Advanced Project Management*. Edycja polska. Wydawnictwo One Press.
- Kerzner, H. (2005a). *Zarządzanie projektami. Studium przypadków*. Gliwice: Onepress Exclusive, Helion.
- Kerzner, H. (2009). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. John Wiley and Sons.
- Kisielnicki, J. (2008). *MIS – systemy informatyczne Zarządzania*. Warszawa: Placet.
- Kisielnicki, J. (2011). *Zarządzanie projektami. Ludzie – procedury – wyniki*. Warszawa: Oficyna Wolters Kluwer Business.
- Kisielnicki, J. i Sroka H. (1999). *Systemy informacyjne biznesu*. Warszawa: Placet.
- Kloze, M. (2009). *Kluczowe aspekty zarządzania projektami*. Prezentacja szkoleniowa Akademii Konsultingu, Deloitte.
- Korczak, J. i Dyczkowski, M. (red.) (2008). *Informatyka Ekonomiczna, część I, Propedeutyka informatyki, technologie informacyjne*. Wrocław: Wydawnictwo Uniwersytetu Ekonomicznego we Wrocławiu.
- Koszłajda, A. (2010). *Zarządzanie projektami IT, Przewodnik po metodykach*. Gliwice: Helion.
- Kruger, W. (1993). Projektmanagement. W: *Handwörterbuch der Betriebswirtschaft*. Stuttgart: Schaffer-Poeschel Verlag.
- Krupa, K. (2006). *Teoria zmian organizacyjnych przedsiębiorstw ery informacji (wybrane aspekty i narzędzia)*. Rzeszów: Wydawnictwo Uniwersytetu Rzeszowskiego.
- Laudon, K. i Laudon J. (2010). *Management Information Systems*. Prentice Hall.
- Layland, V. (1995). *EDI. Elektroniczna wymiana dokumentacji*. Warszawa: Wydawnictwo Naukowo-Techniczne.
- Lech, P. (2003). *Zintegrowane systemy zarządzania ERP/ERP II – Wykorzystanie w biznesie, wdrażanie*. Warszawa: Difin.
- Managing and Directing Successful Project with Prince2*. Pozyskano z: [http://www.best-management-practice.com/gempdf/PRINCE2-2009\\_Overview\\_Brochure\\_June2011.pdf](http://www.best-management-practice.com/gempdf/PRINCE2-2009_Overview_Brochure_June2011.pdf) (styczeń 2013).
- Martin, J. i Odell, J. (1994). *Podstawy metod obiektowych*. Warszawa: WNT.
- Mingus, N. (2009). *Zarządzanie projektami*. Gliwice: Helion.
- Modliński, W. (2005). Firma na poziomie 6, czyli mity i prawdy o metodzie Six Sigma, *Personel i Zarządzanie*, 11. Pozyskano z: [http://www.4pm.pl/artukul/firma\\_na\\_pozio- mie\\_6\\_czyli\\_mity\\_i-prawdy\\_o\\_metodzie\\_six\\_sigma-26-95.html](http://www.4pm.pl/artukul/firma_na_pozio- mie_6_czyli_mity_i-prawdy_o_metodzie_six_sigma-26-95.html) (styczeń 2013).

- Mumford, E. (1995). *Effective Requirements Analysis and System Design: The ETHICS Method*. McMillan: Baringstoke.
- Niedzielska, E. (red.) (1998). *Informatyka ekonomiczna*. Warszawa: PWE.
- Niedźwiedziński, M. (2004). *Globalny handel elektroniczny*. Warszawa: Wydawnictwo Naukowe PWN.
- Niemiec, A. i Grocholski, L. (2005). Inżynieria oprogramowania w świetle wytycznych norm międzynarodowych. W: M. Goliński, J.K. Grabara i J.S. Nowak (red.), *Informatyka i efektywność systemów*, s. 250–262. Katowice: PTI Oddział Górnośląski.
- Nowicki, A. (red.) (2005). *Wstęp do systemów informacyjnych zarządzania w przedsiębiorstwie*. Częstochowa: Wydawnictwo Politechniki Częstochowskiej.
- Nowicki, A. (red.) (2006). *Komputerowe wspomaganie biznesu*. Warszawa: Placet.
- Nowicki, A. i Turek, T. (red.) (2010). *Technologie informacyjne dla ekonomistów*. Wrocław: Wydawnictwo Uniwersytetu Ekonomicznego we Wrocławiu.
- OECD, (1998). *The Economic and Social Impacts of Electronic Commerce: Preliminary Findings and Research Agenda*. Pozyskano z: <http://www.oecd.org/dataoecd/3/12/1944883.pdf> (luty 2013).
- OGC, (2005). *PRINCE2. Skuteczne zarządzanie projektami*. Londyn: TSO.
- Olszak, C. (2007). *Tworzenie i wykorzystanie systemów Business Intelligence na potrzeby współczesnej organizacji*. Katowice: Wydawnictwo Naukowe Akademii Ekonomicznej w Katowicach.
- Olszak, C. i Sroka, H. (red.) (2003). *Informatyka w zarządzaniu*. Katowice: Wydawnictwo AE w Katowicach.
- Olszak, C. i Ziemia, E. (red.) (2007). *Systemy gospodarki elektronicznej w erze informacji i wiedzy*. Warszawa: Wydawnictwo Naukowe PWN.
- Pańkowska, M. (2010). Środowiska projektowe przedsięwzięć informatycznych. W: J. Sobieskiej-Karpińskiej (red.), *Informatyka ekonomiczna, Informatyka w zarządzaniu*, s. 238-252. Seria Prace naukowe UE we Wrocławiu nr 15, Wrocław: Wydawnictwo Uniwersytetu Ekonomicznego we Wrocławiu.
- Pawlak, M. (2006). *Zarządzanie projektami*. Warszawa: Wydawnictwo naukowe PWN.
- Pieter, J. (1967). *Ogólna metodologia pracy naukowej*. Wrocław: Ossolineum.
- PMBOK Manual*. Pozyskano z: <http://www.pmi.org> (luty 2012).
- PMI, (2003). *Kompendium wiedzy o zarządzaniu projektami*. Warszawa: MT&DC.
- PMI, (2004). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. Third Edition. Newtown Square, Pennsylvania, USA: Inc., Four Campus Boulevard.
- Pritchard, C.L. (2002). *Zarządzanie ryzykiem w projektach. Teoria i praktyka*. Warszawa: IG-Press.
- Rising, L. i Janoff, N.S. (2002). The Scrum Software Development Process for Small Teams, *IEEE Software*, 17(4), 26–32.
- Roszkowski, J. (1998). *Analiza i projektowanie strukturalne*. Gliwice: Helion.
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. i Lorensen, W. (1991). *Object-Oriented Modeling and Design*. Englewood Cliffs, NJ: Prentice-Hall.
- Rummler, G., Ramias, A. i Rummler, R. (2010). *White Space Revisited: Creating Value through Process*. San Francisco: Jossey Bass.
- Saaty, T.L. (1999). *Fundamentals of the Analytic network proces*. ISAHP, Kobe, 8.
- Sacha, K. (2010). *Inżynieria oprogramowania*. Warszawa: PWN.
- Schmidt, P. (2010). *Dlaczego warto zarządzać projektami?*. Pozyskano z: [http://grupapm.pl/files/NiS\\_07-8\\_2010.pdf](http://grupapm.pl/files/NiS_07-8_2010.pdf) (styczeń 2013).

- Schwaber, K. (2005). *Sprawne zarządzanie projektami metodą Scrum*. Warszawa: Promise.
- Schwaber, K. i Beedle, M. (2001). *Agile Software Development with Scrum*. NY: Prentice Hall.
- Schwaber, K. i Sutherland, J. (2012). *Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust*. NY: Wiley&Sons.
- Silver, M.S. (1991). *Systems That Support Decisions Makers. Description and Analysis*. NY: John Wiley & Sons.
- Snedaker, S. (2007). *Zarządzanie projektami IT w małym palcu*. Gliwice: Helion.
- Sobczak, A. (2012). *Dlaczego modele architektoniczne to za mało*. Pozyskano z: [http://architekturakorporacyjna.pl/wp-content/uploads/downloads/2012/01/Dlaczego\\_modele\\_architektoniczne\\_to\\_za\\_malo\\_Wprowadzenie\\_do\\_ladu\\_architektury\\_korporacyjnej.pdf](http://architekturakorporacyjna.pl/wp-content/uploads/downloads/2012/01/Dlaczego_modele_architektoniczne_to_za_malo_Wprowadzenie_do_ladu_architektury_korporacyjnej.pdf) (luty 2013).
- Sobczak, A. (2012a). *Koncepcja metodycznego podejścia do podnoszenia interoperacyjności w organizacjach publicznych z zastosowaniem architektury korporacyjnej*. Pozyskano z: [http://www.architekturakorporacyjna.pl/wp-content/uploads/downloads/2012/01/Koncepcja\\_metodycznego\\_podejscia\\_do\\_podnoszenia\\_poziomu\\_interoperacyjnosci\\_w\\_organizacjach\\_publicznych\\_z\\_zastosowaniem\\_architektury\\_korporacyjnej.pdf](http://www.architekturakorporacyjna.pl/wp-content/uploads/downloads/2012/01/Koncepcja_metodycznego_podejscia_do_podnoszenia_poziomu_interoperacyjnosci_w_organizacjach_publicznych_z_zastosowaniem_architektury_korporacyjnej.pdf) (styczeń 2013).
- Software Engineering Institute (2006). *CMMI Product Team, CMMI for Development, Version 1.2*. Pitsburg: Carnegie Mellon University, sierpień.
- Sprague, R.H. (1987). DSS in Context. *Decision Support Systems*, 3(3).
- Stabryła, A. (2006). *Zarządzanie projektami ekonomicznymi i organizacyjnymi*. Warszawa: Wydawnictwo Naukowe PWN.
- Stair, R.M. (1992). *Principles of Information Systems. A Managerial Approach*. Boston: Boyd&Fraser Publishing Company.
- Standish Group International, (2007). *The Standish Group Report 2007*. Massachusetts: West Yarmouth.
- StatSoft Polska, (2012). *Statistica a „Metodyka Sześć Sigma”*. Pozyskano z: <http://www.statsoft.pl/sixsigma.html> (grudzień, 2012).
- Stępień, P. (2012). *Wprowadzenie do zarządzania projektami*, część I. Pozyskano z: <http://www.skutecznyprojekt.pl/arttykul.htm?AID=65> (luty 2012).
- Szyjewski, Z. (2004) *Metodyki zarządzania projektami informatycznymi*. Warszawa: Placet.
- Tartanus, Ł. (2010). *Dojrzałość procesowa polskich organizacji*. Pozyskano z: [www.procesowcy.pl](http://www.procesowcy.pl) (styczeń 2013).
- The Standish Group, (2009). *Chaos Summary*. Massachusetts: West Yarmoth. Pozyskano z: <http://www.controlchaos.com/storage/S3D%20First%20Chapter.pdf> (listopad, 2012).
- The Standish Group International, (2009a). Incorporated, *CHAOS Report 2009*. Pozyskano z: <http://blog.standishgroup.com/news> (wrzesień 2012).
- Trocki M. (red.) (2012). *Nowoczesne zarządzanie projektami*. Warszawa: PWE.
- Trocki, M., Gruzca, B. i Ogonek, K. (2003). *Zarządzanie projektami*. Warszawa: PWE.
- Turban, E., Leidner, D., McLean, E. i Wetherbe J. (2008). *Information Technology for Management. Transforming Organizations in the Digital Economy*. NY: Wiley and Sons.
- VitalSmarts, (2006). *Executive Summary., The Five Crucial Conversations for Flawless Execution*. Pozyskano z: <http://www.silencefails.com/downloads/media/SilenceFailsExecutiveSummary.pdf>. (luty 2013).

- VitalSmarts, (2006). *Silence Fails: The Five Crucial Conversations for Flawless Execution*. Pozyskano z: <http://www.silencefails.com/downloads/media/SilenceFailsExecutiveSummary.pdf> (styczeń 2013).
- VitalSmarts, (2006a). *Silence Fails: The Five Crucial Conversations for Flawless Execution*. Pozyskano z: <http://www.silencefails.com/downloads/SilenceFailsFullReport.pdf> (luty 2013).
- Waszczuk, P. (2008). *Warto rozmawiać*. Pozyskano z: <http://www.computerworld.pl/artykuly/323361/Warto.rozmawiac.html> (luty 2013).
- Wielki, J. (2000). *Elektroniczny marketing poprzez Internet*. Warszawa-Wrocław: Wydawnictwo Naukowe PWN.
- Wielki, J. (2012). *Modele wpływu przestrzeni elektronicznej na organizacje gospodarcze*. Wrocław: Wydawnictwo UE Wrocław.
- Wilson, B. (1990). *Systems: Concepts, Methodologies and Applications*. NY: John Wiley & Sons Ltd.
- Wirfs-Brock, R., Wilkerson, B. i Wiener, L. (1990). *Designing object-oriented software*. Englewood Cliffs, NJ: Prentice Hall.
- Wojtkiewicz, K. (2012). *Metodyka definiowania i wdrożenia strategii informatyzacji w przedsiębiorstwie zarządzanym procesowo*. Niepublikowana praca doktorska pod kierunkiem W. Chmielarza, Wydział Zarządzania UW, Warszawa.
- Wrycza, S. (2010). *Informatyka ekonomiczna*. Warszawa: Wydawnictwo Naukowe PWN.
- Wysocki, R. i McGary, K. (2005). *Efektywne zarządzanie projektami*. Gliwice: OnePress.
- Wysocki, R. i McGary, R. (2003). *Efektywne zarządzanie projektami*. Gliwice: Helion.
- Yourdon, E. (2006). *Death March*. 2-nd Edition. NY: Yourdon Press.
- Ziomba, E. (2007). Portale korporacyjne. W: C. Olszak i E. Ziomba (red.), *Strategie i modele gospodarki elektronicznej*, s. 344–356. Warszawa: Wydawnictwo Naukowe PWN.
- Ziomba, E. (2009). *Projektowanie portali korporacyjnych dla organizacji opartych na wiedzy*. Katowice: Wydawnictwo Akademii Ekonomicznej im. Karola Adamickiego.

## Spis tabel

---

Tabela 1. Podobieństwa i różnice pomiędzy projektami i procesami .....	15
Tabela 2. Średnia skala realizacji projektów analizowanych przez The Standish Group w wybranych latach z okresu 1994–2010 .....	17
Tabela 3. Czynniki utrudniające realizację projektu .....	17
Tabela 4. Pokrycie etapów cyklu życia systemu informatycznego przez wybrane metodyki zwinne .....	55
Tabela 5. Elementy metodyki Prince2 (2009 r.) .....	76

## Spis rysunków

---

Rysunek 1. Relacje pomiędzy projektami, procesami a przedsięwzięciem i programem .....	14
Rysunek 2. Obszar dopuszczalnych kombinacji podstawowych parametrów projektu i jego rozszerzenie .....	20
Rysunek 3. Podstawowe fazy cyklu życia projektu w modelu kaskadowym ..	34
Rysunek 4. Schemat ideowy ewolucyjnego modelu życia systemu .....	35
Rysunek 5. Schemat ideowy przyrostowego cyklu życia systemu .....	36
Rysunek 6. Schemat ideowy modelu tworzenia struktury baz danych .....	37
Rysunek 7. Schemat ideowy modelu z tworzeniem prototypu .....	38
Rysunek 8. Schemat postępowania w spiralnym modelu cyklu życia projektu .....	40
Rysunek 9. Podstawowe fazy cyklu życia projektu w modelu XP .....	46
Rysunek 10. Cykl życia systemu informatycznego w metodyce Scrum .....	48
Rysunek 11. Cykl życia systemu informatycznego w metodyce FDD .....	50
Rysunek 12. Cykl życia projektu informatycznego w metodyce DSDM .....	52
Rysunek 13. Cykl życia projektu informatycznego w metodyce ASD .....	54
Rysunek 14. Porównanie sukcesu projektów realizowanych za pomocą metod tradycyjnych i nowoczesnych .....	58
Rysunek 15. Kierunki rozwoju metodyk projektowania systemów informatycznych .....	60
Rysunek 16. Nakładanie się grup procesów w ramach danego etapu projektu .....	66
Rysunek 17. Skutki wdrożenia systemu informatycznego wspomagającego zarządzanie, generacja strat i zysków w okresie powdrożeniowym .....	98
Rysunek 18. Struktura logiczna systemu informacyjnego zarządzania .....	107
Rysunek 19. Architektura logiczna informatycznego systemu wspomaganie podejmowania decyzji .....	110
Rysunek 20. Architektura logiczna systemu informatycznego klasy EIS/ESS .....	111
Rysunek 21. Architektura logiczna systemu eksperckiego .....	114
Rysunek 22. Architektura logiczna systemu Business Intelligence .....	115
Rysunek 23. „Efekt słonecznika” w rozwoju zintegrowanych systemów zarządzania .....	122
Rysunek 24. Struktura systemu zintegrowanego na przykładzie systemu IFS Applications .....	123



---

Rysunek 25. Bilanse w systemach zintegrowanych .....	125
Rysunek 26. Etapy rozwoju systemów opartych na sieci .....	131
Rysunek 27. Ścieżki rozwoju systemów informatycznych wspomagających zarządzanie .....	133
Rysunek 28. Ewolucja technologii informacyjnych ze względu na zmiany metod zarządzania projektami systemów informatycznych .....	140



Prof. dr hab. Witold Chmielarz – kierownik Katedry Systemów Informatycznych Zarządzania Wydziału Zarządzania UW, pasjonat oceny użyteczności, efektywności i jakości systemów informatycznych z punktu widzenia ich użytkownika. Od kilku lat prowadzi intensywne badania stanu i dynamiki gospodarki elektronicznej, szczególnie e-bankowości. Autor i współautor ponad dwustu publikacji naukowych, w tym kilku książek poświęconych problemom tworzenia i zastosowania systemów informatycznych, oraz kilkunastu rozdziałów w książkach, wydanych w kraju i za granicą. Odbýwał staże naukowe i prowadził wykłady m.in. w USA i Indiach oraz kilku krajach europejskich. Przez wiele lat pracował jako analityk, projektant, wdrożeniowiec i kierownik projektów informatycznych w firmach komputerowych w Polsce. Rozpoznawalny, nie tylko w kraju, ekspert w zakresie e-biznesu, w tym e-bankowości; metod analizy i projektowania systemów informatycznych, w tym metodyki dostosowania systemów informatycznych do wymagań klienta oraz wdrożeń systemów zintegrowanych.

*Książka jest wartościowym opracowaniem ze względu na zawartość merytoryczną, zwłaszcza rozważania teoretyczne nad rozwojem metodologii SIZ oraz wskazania praktyczne. [...] z pewnością przyczyni się do upowszechnienia i rozwoju usystematyzowanej wiedzy nie tylko wśród nauczycieli akademickich i studentów informatyki ekonomicznej, lecz także wśród specjalistów z praktyki gospodarczej, zaangażowanych w zarządzanie organizacjami oraz zarządzanie projektami.*

Prof. dr hab. Jerzy Gołuchowski

*Książka jest monografią o oryginalnym autorskim ujęciu problematyki zarządzania projektami informatycznymi i rozwoju systemów informatycznych zarządzania. Autor podejmuje w niej ambitne zadanie odpowiedzi na pytanie: czy i w jaki sposób rozwój metod zarządzania projektami wpływa na rozwój systemów informatycznych, a jednocześnie docieka czy jest to relacja zwrotna, czy rozwój technologii informacyjnych stymuluje rozwój metod projektowania. Na rynku polskim brakuje pracy podejmującej problematykę zarządzania projektami informatycznymi i rozwoju systemów informatycznych postawioną w tak istotnym ujęciu dla badań postępu w zakresie zastosowań informatyki. [...] stanowi może ciekawy materiał dla ekspertów i praktyków zarządzania – zarówno do przemysłów naukowych i jako inspiracja nowych koncepcji, jak i do nauki podstaw zarządzania projektami, w kontekście systemów informatycznych. [...] Dodatkową zaletą jest tu oparta na praktyce postawa Autora do możliwości wykorzystania prezentowanych metod w rzeczywistości.*

Prof. dr hab. Mirosława Lasek