

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

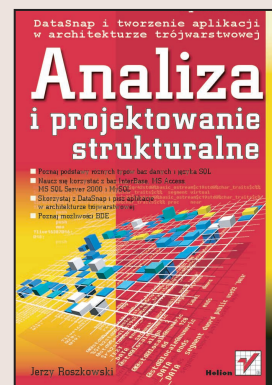
FRAGMENTY KSIĄŻEK ONLINE

Analiza i projektowanie strukturalne. Wydanie III

Autor: Jerzy Roszkowski

ISBN: 83-7361-397-8

Format: B5, stron: 256



Technologia CASE (Computer Aided System Engineering) jest obecnie od dziesięciu lat powszechnie stosowana w analizie i projektowaniu systemów informatycznych. Trudno sobie wyobrazić pracę bez niej (szczególnie przy dużych projektach) na etapach:

- analizy,
- tworzenia projektu systemu,
- a także samej implementacji.

Techniki CASE umożliwiają wspomaganie:

- analizy i projektowania bazy danych,
- projektowania aplikacji
- generacji kodu aplikacji
- automatycznego tworzenia dokumentacji analizy i projektu
- inżynierii odwrotnej (tworzenie modeli fizycznych i logicznych aplikacji na podstawie jej kodu i fizycznej bazy danych)

Techniki strukturalne są w dalszym ciągu kluczowymi w projektowaniu aplikacji bazodanowych. Niniejsza książka opisuje te techniki, stosując jako egzemplifikację klasyczną metodykę Yourdona (rozkład funkcjonalny), a także metodykę SSADM oraz (w zakresie modelowania danych) metodykę Martina. Autor na podstawie swojego dziesięcioletniego doświadczenia w stosowaniu technologii CASE, odwołując się do projektów którymi kierował, przedstawia możliwości i ograniczenia prezentowanej metodyki. Na konkretnych przykładach autor uczy jak budować aplikacje na etapie analizy i projektu posługując się technikami strukturalnymi. Uzupełnieniem są załączone przykłady w formie zadań z rozwiązaniami.

Zagadnienia omówione w książce obejmują zakres tematyczny:

- Budowy logicznych modeli danych i funkcjonalnego systemu
- Przekształcanie modeli logicznych w model fizyczny
- Przekształcanie modelu funkcjonalnego w model aplikacji
- Bilansowanie modeli
- Analizy systemów obiegu dokumentów
- Analizy systemów budowanych z gotowych komponentów
- Analizy cykli różnych wytwórczych oprogramowania
- Analizy i projektowania hurtowni danych



Spis treści

| | |
|--|-----------|
| Wprowadzenie | 7 |
| Rozdział 1. Ogólne metody analizy systemowej | 9 |
| Rozkład funkcjonalny | 10 |
| Model funkcjonalny — metoda przepływu danych | 11 |
| Modelowanie informacji (danych) | 11 |
| Podejście obiektowe | 12 |
| Rozdział 2. Diagramy modelowania metodyki strukturalnej | 13 |
| Charakterystyka narzędzi modelowania | 13 |
| Trzy modele systemu | 14 |
| Model funkcjonalny — diagramy przepływu danych (Data Flow Diagrams) | |
| — metodyka Yourdona — przykłady — typowe błędy | 14 |
| Elementy składowe DFD | 15 |
| Główne zalecenia przy projektowaniu DFD | 22 |
| Wielopoziomowe DFD | 24 |
| Rozszerzenia do DFD dla systemów czasu rzeczywistego | 28 |
| Model funkcjonalny — diagramy przepływu danych (Data Flow Diagrams) | |
| — metodyka SSADM — przykłady | 29 |
| Elementy składowe DFD w metodyce SSADM | 29 |
| Model danych — diagramy obiekt-relacja-atrybut | |
| (Entity Relationship Diagrams — ERD) — metodyka Martina | 30 |
| Elementy diagramu ERD | 33 |
| Projektowanie logiczne danych — model relacyjny | 39 |
| Projektowanie logiczne danych — normalizacja danych | 41 |
| Zależności atrybutów | 42 |
| Projektowanie logiczne danych — modelowanie tablic | 49 |
| Mapowanie w sytuacji interpretacji subtypów | |
| przez relację wzajemnego wykluczania się | 55 |
| Przekształcenie modelu funkcjonalnego w projekt strukturalny | |
| — diagramy strukturalne (STC Structured Charts) | 56 |
| Model dynamiki — diagramy przejść stanów (State Transition Diagrams) | 58 |
| Rozdział 3. Słownik danych (Data Dictionary) | 65 |
| Formalizm notacji słownika danych | 65 |
| Definicje | 66 |
| Rozdział 4. Specyfikacja procesów | 69 |

| | |
|---|-----------|
| Rozdział 5. Bilansowanie modelu | 75 |
| Bilansowanie diagramu DFD względem słownika danych (DD)..... | 76 |
| Bilansowanie diagramu DFD względem specyfikacji procesów..... | 76 |
| Bilansowanie specyfikacji procesów względem DFD i słownika danych..... | 76 |
| Bilansowanie słownika danych względem DFD i specyfikacji procesów..... | 77 |
| Bilansowanie ERD względem DFD i specyfikacji procesów..... | 77 |
| Bilansowanie DFD względem diagramu przejść stanów (STD) | 77 |
| Rozdział 6. Cykl projektowy | 81 |
| Etap I — Studium możliwości..... | 81 |
| Etap II — Analiza | 83 |
| Etap III — Projektowanie | 83 |
| Etap IV — Implementacja | 83 |
| Etap V — Przejście na nowy system | 84 |
| Cykle projektowe w technologiach niektórych kluczowych dostawców..... | 85 |
| Definicja potrzeb biznesowych..... | 86 |
| Analiza istniejących systemów | 87 |
| Opracowanie architektury technicznej..... | 87 |
| Projektowanie i budowa bazy danych..... | 87 |
| Projektowanie i budowa modułów..... | 87 |
| Konwersja danych..... | 88 |
| Opracowanie dokumentacji technicznej | 88 |
| Testowanie | 88 |
| Szkolenie..... | 89 |
| Przejście na nowy system | 89 |
| Obsługa serwisowa | 89 |
| CDM — podejście klasyczne..... | 89 |
| Definicja..... | 89 |
| Analiza | 90 |
| Projekt | 90 |
| Budowa | 90 |
| Przejście | 91 |
| Produkcja | 91 |
| CDM — podejście „szybkiej ścieżki” (Fast Track)..... | 91 |
| Modelowanie wymagań..... | 91 |
| Projektowanie i generowanie systemu..... | 91 |
| Przejście do produkcji..... | 92 |
| CDM — podejście „Lite”..... | 92 |
| Prototyp i budowa | 93 |
| Przejście do produkcji..... | 93 |
| Specyfikacja dostaw powstających w ramach przedsięwzięcia informatycznego (według metodyki CDM)..... | 94 |
| Dział I — Specyfikacja wymagań (Requirements Definition) | 94 |
| Dział II — Przegląd istniejącego systemu (Existing system examination) | 95 |
| Dział III — Architektura techniczna (Technical Architecture) | 95 |
| Dział IV — Projektowanie i wytworzenie bazy danych (Database Design and Build) | 96 |
| Dział V — Projektowanie i wytworzenie modułów (Module Design and Build)..... | 96 |
| Dział VI — Konwersja danych (Data Conversion) | 97 |
| Dział VII — Dokumentacja (Documentation)..... | 97 |
| Dział VIII — Testowanie (Testing)..... | 98 |
| Dział IX — Szkolenie (Training) | 98 |
| Dział X — Uruchomienie — przejście (Transition)..... | 99 |
| Dział XI — Wsparcie po uruchomieniu (Post-System Support)..... | 99 |

| | |
|--|------------|
| Rozdział 7. Studium możliwości (Feasibility Study) | 101 |
| Zapoczątkowanie projektu | 101 |
| Wybór przedsięwzięcia | 101 |
| Fazy realizacji | 103 |
| Sporządzanie analizy opłacalności | 105 |
| Rozdział 8. Proces analizy | 107 |
| Podejście klasyczne — cztery modele systemu | 107 |
| Model podstawowy systemu | 110 |
| Model otoczenia | 110 |
| Model zachowania się systemu | 112 |
| Zasady prowadzenia wywiadów | 115 |
| Formularz hierarchii operacji | 116 |
| Formularz wzorów dokumentów | 117 |
| Rozdział 9. Analiza systemu obiegu dokumentów | 119 |
| Formularz i semantyka opisu obiegu dokumentów | 119 |
| Model i jego konkretyzacja | 120 |
| Struktura modelu | 121 |
| Wizualizacja modelu | 130 |
| Rozdział 10. Analiza systemu budowanego z gotowych komponentów | 141 |
| Definicja istniejącej struktury organizacyjnej — (regulamin organizacyjny) | 141 |
| Definicja struktury organizacyjnej | 142 |
| Kluczowy personel jednostki | 142 |
| Grupy użytkowników wewnątrz organizacji | 142 |
| Obiekty (organizacje) zewnętrzne | 142 |
| Zakres analizy w układzie głównych procesów biznesowych — lista obszarów tematycznych (Context process model) | 142 |
| Prototypy podstawowych obiektów informacyjnych, w tym bazy normatywnej globalnej i lokalnej | 143 |
| Inwentaryzacja zasobów osobowych oraz technicznych (infrastruktury i oprogramowania) — istniejąca architektura techniczna | 143 |
| Przegląd architektury | 143 |
| Struktura sieci | 144 |
| Środowisko programowe (software) | 144 |
| Analiza procesów biznesowych istniejącego systemu informacyjnego | 144 |
| Ogólny model koncepcyjny rozwiązania docelowego | 145 |
| Model warstwowy systemu zarządzania | 145 |
| Model przypadków użycia docelowego systemu informatycznego | 145 |
| Model docelowy danych (model logiczny danych) | 145 |
| Bilansowanie obszarów tematycznych z gotowymi aplikacjami | 146 |
| Bilansowanie przypadków użycia obszaru tematycznego i aplikacji | 147 |
| Bilansowanie modelu logicznego danych z zakresem danych aplikacji | 149 |
| Rozdział 11. Analiza i projektowanie testów | 151 |
| Rodzaje i techniki testów | 153 |
| Testy regresyjne | 154 |
| Testy operacyjne | 154 |
| Testy pełnozakresowe (przy pełnym obciążeniu systemu) | 154 |
| Testy wydajnościowe | 155 |
| Testy negatywne | 155 |
| Testy ergonomiczne | 155 |
| Testy dokumentacji użytkownika końcowego | 155 |
| Testy akceptacyjne (α -testy i β -testy) | 156 |

| | | |
|------------------|---|------------|
| Dodatek A | Zastosowanie metod strukturalnych w projektowaniu hurtowni danych..... | 157 |
| | Niedostatki systemów wspomaganie decyzji oraz hurtownie danych jako usuwające je — koncepcje zmian | 157 |
| | Architektura i funkcje hurtowni danych..... | 160 |
| | Repozytorium metadanych | 162 |
| | Technologia bazy danych hurtowni danych | 163 |
| | Narzędzia zapytań, raportowania i analizy oraz narzędzia „data mining” | 163 |
| | Administracja i zarządzanie hurtownią danych | 164 |
| | Struktura hurtowni danych | 165 |
| | Warianty architektury technicznej hurtowni danych | 166 |
| | Wirtualna hurtownia danych..... | 166 |
| | Architektura wielu składnic danych..... | 168 |
| | Architektura hurtowni z dostępem tylko do składnic danych..... | 169 |
| | Architektura hurtowni z dostępem mieszanym..... | 171 |
| | Przykładowa specyfikacja tematycznych hurtowni danych..... | 173 |
| | Hurtownia danych w zakresie analizy i planu sprzedaży | 173 |
| | Hurtownia danych w zakresie analizy, planu i rozliczenia produkcji | 174 |
| | Hurtownia danych w zakresie analizy kosztów | 176 |
| | Przykładowe specyfikacje tematyczne systemów wspomaganie decyzji opartych na hurtowniach (aplikacje klienta w technologii klient-serwer)..... | 177 |
| | Aplikacje klienta obsługujące hurtownie danych | 177 |
| | Dedykowane systemy klasy DSS oparte na hurtowniach danych | 178 |
| | Specyfikacja cyklu projektowego dla hurtowni danych | 179 |
| | Określenie funkcji zarządzania wspieranych przez hurtownie | 180 |
| | Dokumentowanie istniejących w przedsiębiorstwie systemów transakcyjnych..... | 181 |
| | Doprowadzenie do spójności metadanych pomiędzy systemami transakcyjnymi przedsiębiorstwa | 181 |
| | Specyfikacja wymagań systemów DSS oraz aplikacji klienta obsługujących hurtownie danych | 181 |
| | Projektowanie hurtowni danych | 182 |
| | Specyfikacja mapowania i transformacji danych | 182 |
| | Narzędzia do analizy i projektowania..... | 182 |
| | Cykl realizacji | 183 |
| Dodatek B | Zadania..... | 187 |
| | Zadanie 1. — Diagramy przepływu danych i związków encji (ERD) | 187 |
| | Zadanie 2. — Diagramy przepływu danych i związków encji (ERD) | 189 |
| | Zadanie 3. — Diagramy związków encji (ERD) | 190 |
| | Zadanie 4. — Diagramy związków encji (ERD) | 190 |
| | Zadanie 5. — Diagramy związków encji (ERD) | 191 |
| | Zadanie 6. — Diagramy związków encji (ERD) | 192 |
| | Zadanie 7. — Studium możliwości | 193 |
| | Zadanie 8. — Zarządzanie marketingiem i kontrola procesu wytwórczego | 195 |
| | Zadanie 9. — Diagram obiegu dokumentów | 195 |
| | Zadanie 10. — Projekt modelu logicznego hurtowni danych w zakresie analizy sprzedaży | 197 |
| | Zadanie 11. — Projekt modeli logicznych kostek informacyjnych hurtowni danych w zakresie analiz finansowych i kosztów w przedsiębiorstwie | 198 |
| Dodatek C | Rozwiązania..... | 207 |
| | Literatura | 247 |
| | Skorowidz..... | 249 |

Rozdział 6.

Cykl projektowy

Wprowadzone w poprzednich rozdziałach narzędzia modelowania wykorzystuje się na różnych etapach cyklu projektowego.

Są trzy podstawowe cele wprowadzenia pojęcia cyklu projektowego:

- ◆ aby zdefiniować czynności w procesie budowy systemu,
- ◆ aby wprowadzić i utrzymać spójność pomiędzy wieloma projektami w tej samej organizacji,
- ◆ aby wprowadzić punkty kontrolne w zarządzaniu projektem na różnych etapach jego rozwoju.

Na rysunku 6.1 przedstawiono etapy klasycznego cyklu projektowego, najczęściej definiowane podczas budowy systemu.

Etap I — Studium możliwości

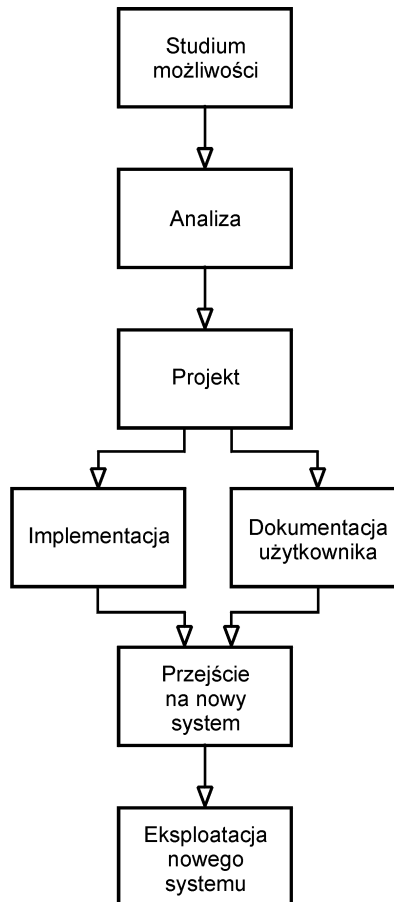
Zazwyczaj zaczyna się od zapytania użytkownika, czy można zautomatyzować jeden albo więcej elementów jego działalności.

Głównymi przyczynami wprowadzenia studium możliwości są:

- ◆ *identyfikacja ludzi odpowiedzialnych za określenie celu systemu*; prowadzi to do ustalenia szeregu interview, w wyniku których zostanie to sprecyzowane oraz zdefiniowany zostanie początkowy diagram kontekstu.
- ◆ *identyfikacja wad i niedostatków aktualnego systemu informatycznego-informacyjnego*; składa się z listy funkcji, których brakuje lub nie są wykonywane właściwie przez istniejący system.

Rysunek 6.1.

Klasyczny cykl projektowy stosowany w analizie strukturalnej



- ◆ *ustalenie celów i ograniczeń nowego systemu*; może to być także prosta lista istniejących funkcji systemu, które muszą być zaimplementowane ponownie, nowych funkcji, które muszą być dodane oraz lista kryteriów, które musi spełniać nowy system.
- ◆ *określenie wykonalności systemu z podaniem kilku scenariuszy*; powinien być określony harmonogram, koszt budowy nowego systemu oraz uzyskane korzyści. Zazwyczaj proponuje się kilka architektur dla wdrożenia systemu, na przykład przetwarzanie scentralizowane (*mainframe*), przetwarzanie rozproszone, architektura klient-serwer etc.
- ◆ *określenie lidera projektu (project manager)*; studium możliwości zajmuje na ogół od 5% do 10% całego czasu trwania projektu.

Etap II — Analiza

Głównym celem etapu analizy jest wprowadzenie strukturalnej specyfikacji opisu projektu za pomocą narzędzi modelowania wprowadzonych w poprzednich rozdziałach, tzn. diagramów przepływu danych — DFD, diagramów obiekt-relacja-atrybut — ERD, diagramów przejść stanów — STD.

Rezultatem analizy jest zbudowanie następujących modeli:

- ♦ model otoczenia,
- ♦ model zachowania systemu.

Modele te omówiono w rozdziale 7. Są one opisem formalnym systemu, niezależnym od technologii, jakiej użyje się do implementacji nowego systemu.

Na końcu etapu analizy określa się dokładniej niż w poprzednim etapie budżet projektu oraz kalkulację kosztów i zysków.

Etap III — Projektowanie

Etap ten przeznaczony jest też do budowy tzw. *modelu implementacji użytkownika*, która powinna zawierać:

- ♦ wyodrębnienie tych części *modelu zachowania systemu*, które będą implementowane w systemie informatycznym,
- ♦ przydzielenie poszczególnych części specyfikacji do odpowiednich procesorów lub serwerów (przetwarzanie rozproszone). Wycięte fragmenty DFD (te, które będą implementowane) są mapowane na zadania (*tasks*) — tu interfejsu użytkownika końcowego,
- ♦ zaprojektowanie struktury hierarchii modułów wewnątrz danego zadania, jak to opisano w podrozdziale „Przekształcenie modelu funkcjonalnego w projekt strukturalny — diagramy strukturalne (STC Structured Charts)”. Ponadto podczas etapu projektowania należy dokonać transformacji diagramów ERD na relacyjną bazę danych (projektowanie logiczne danych), tak jak to opisano w podrozdziale „Model danych — diagramy obiekt-relacja-atrybut (Entity Relationship Diagrams — ERD) — metodyka Martina”.

Etap IV — Implementacja

Etap ten składa się z kodowania i integracji modułów. Stosuje się techniki programowania strukturalnego oraz implementacji *top-down*.

Etap V — Przejście na nowy system

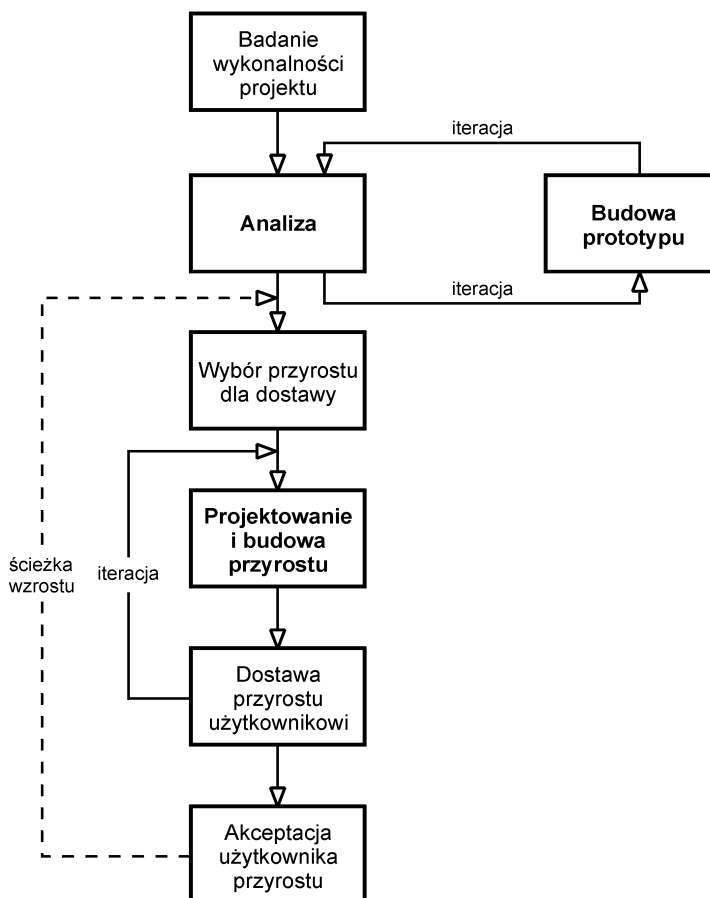
Podczas tego etapu wykonuje się następujące czynności:

- ◆ testy akceptacyjne systemu zapewniające jego właściwą jakość,
- ◆ konwersję bazy danych przy przejściu ze starego systemu na nowy,
- ◆ instalację.

W ostatnich latach rozpowszechnia się inny cykl budowy aplikacji spełniający wymagania *wielowarstwowej architektury klient-serwer*. Cykl ten schematycznie przedstawiono na rysunku 6.2. Ten cykl życia stosuje *iteracyjne prototypowanie aplikacji* w fazie analizy, a następnie przynosi szereg *przyrostowych dostaw*.

Rysunek 6.2.

Przyrostowy cykl życia projektu informatycznego, uwzględniający wymagania budowy aplikacji w technice obiektowej



Każdy przyrost odpowiada zaimplementowaniu pojedynczego *USE CASE* (*przypadku użycia*) lub zbioru przypadków użycia pojęć, wprowadzonych w metodyce obiektowej Jacobsona analizy i projektowania. [Jacobson-94].

Iteracyjne prototypowanie aplikacji jest także często stosowane z zastosowaniem technik strukturalnych.

Cykle projektowe w technologiach niektórych kluczowych dostawców

Kluczowi dostawcy oprogramowania opracowali własne cykle wytwórcze będące wewnętrznymi standardami tych firm. Przykładem może być tu Oracle Custom Development Method (CDM) [CDM-Oracle-1996]. Metodyka ta stosuje się do tych procesów biznesowych i funkcji, które nie mogą być obsługiwane za pomocą dostępnych aplikacji „z półki”.

CDM jest zbiorem zdefiniowanych procesów tworzenia oprogramowania, które zostały określone przy założeniu, że w procesie wytwórczym stosujemy metody i narzędzia CASE. Metodyka ta zakłada, że potrzeby biznesowe zostają wyraźnie zdefiniowane na samym początku cyklu projektowego oraz że ich zweryfikowanie jest możliwe podczas całego procesu wytwórczego.

Stosowanie CDM znacznie zwiększa prawdopodobieństwo pozytywnego zakończenia wdrożenia, ponieważ w wyniku stosowania tej metodyki otrzymujemy aplikacje zgodne z celami i potrzebami klienta.

CDM określa zdefiniowane *zadania* (*tasks*) i *dostawy* (*deliverables*), z których składa się pełen cykl projektowy. Każde zadanie wytwarza jedną zdefiniowaną dostawę (produkt). Zadania przyporządkowane są do procesów. Każdy z procesów składa się z wielu zadań. Każdy z procesów jest realizowany w określonej fazie wytwórczej (lub w wielu fazach). Zakończenie konkretnej fazy odpowiada osiągnięciu odpowiednich celów i tzw. „kamieni milowych”.

Każde ze zdefiniowanych zadań może się rozpocząć pod warunkiem dostarczenia wykonawcy dostaw wynikających z zadań poprzedzających. Dostawy te muszą być dostępne, zanim rozpocznie się praca nad przedmiotowym zadaniem. Zależności pomiędzy zadaniami oraz pomiędzy dostawami i zadaniami pozwalają kierownikom projektów podczas procesu planowania pracy określić skutki wykluczenia lub zmiany dostawy.

Struktura metodyki CDM opiera się zatem na metodologii budowy systemów opartej na zdefiniowanych *procesach*.

Proces w tej metodyce określa się jako spójny zbiór albo jako ciąg powiązanych ze sobą zadań, których *wykonanie* to określony wcześniej cel projektu. Rezultatem wykonania procesu jest jedna lub więcej *dostaw*. Można rozumieć proces jako podprojekt wewnątrz większego projektu. Pełny cykl projektowy składa się z wielu procesów. Większość procesów zachodzi na siebie czasowo i są one powiązane pomiędzy sobą poprzez wspólne dostawy.

W metodyce CDM wyróżnia się następujące *procesy*:

- ◆ definicja potrzeb biznesowych (studium możliwości),
- ◆ analiza istniejących systemów,
- ◆ opracowanie architektury technicznej,
- ◆ projektowanie i budowa bazy danych,
- ◆ projektowanie i budowa modułów,
- ◆ konwersja danych,
- ◆ opracowanie dokumentacji technicznej,
- ◆ testowanie,
- ◆ szkolenie,
- ◆ przejście na nowy system,
- ◆ obsługa serwisowa.

Definicja potrzeb biznesowych

Definicja potrzeb biznesowych określa wymagania biznesowe co do aplikacji końcowej. Zespół analityków buduje najpierw *model procesów biznesowych*, zawierający wszystkie zdarzenia biznesowe i reakcje na nie, które musi wspierać aplikacja. Następnie zespół ten buduje *model danych biznesowych* reprezentujący potrzeby informacyjne oraz *model funkcji biznesowych*, w którym podane są szczegóły funkcji biznesowych wskazanych przez model procesów.

Gdy tylko potrzeby biznesowe zostają zdefiniowane, ten sam zespół analityków dodaje do modeli wymagania technologiczne, takie jak interfejs użytkownika, czas odpowiedzi itd. W ten sposób zespół przekształca modele wymagań biznesowych na modele wymagań systemowych.

Analiza istniejących systemów

Istotnym wymaganiem w wielu budowanych projektach jest zastąpienie funkcjonalności istniejących systemów lub też praca nowego systemu w istniejącej architekturze technicznej. Proces *analizy istniejących systemów* spełnia te wymagania. Wiele z zadań w tym procesie może być usuniętych, jeśli projekt nie jest tylko funkcjonalnym zastąpieniem istniejącego systemu.

Proces *analizy istniejących systemów* można znacznie przyspieszyć, jeśli dysponujemy techniczną dokumentacją istniejącego systemu.

Opracowanie architektury technicznej

Proces ten określa elementy techniczne opracowywanego projektu. Przyjmuje się, że istnieje większość informacji zgromadzonych w fazie analizy. Są one podstawą do opracowania początkowej wersji architektury technicznej. Zespół analityków przekształca tę dostawę w dwa opracowania, „Definicja sprzętu i oprogramowania” (Hardware and Software Definition) oraz „Architektura rozproszona” (Distribution Architecture). W tym procesie powinny też zostać określone warunki dla bezpieczeństwa systemu oraz operacji *backup* (wykonywania kopii zapasowych) i *recovery* (odzyskiwania danych). Ostatnią dostawą w tym procesie jest opracowanie planu obciążenia (wydajności) przetwarzania i przesyłania oraz składowania danych przez system.

Projektowanie i budowa bazy danych

Proces *projektowania i budowy bazy danych* rozpoczyna się wykonaniem zadania „Projektowanie logiczne bazy danych” i kończy wygenerowaniem bazy produkcyjnej. Proces budowy relacyjnej bazy danych uwzględnia budowę schematu, projektowanie i budowę indeksów. Wygenerowana baza fizyczna używa *Planu wydajności* i *Planu przetwarzania* oraz *Architektury rozproszonej* jako podstawowych wejściowych dostaw.

Projektowanie i budowa modułów

Projektowanie i budowa modułów to główna część metodyki CDM. Projektanci używają *Modelu procesów systemowych*, *Modelu danych systemowych* oraz *Modelu funkcji systemowych* razem z architekturą techniczną do zaprojektowania pierwszej wersji architektury systemu oraz do opracowania procesu *Model modułów*. Następnie są specyfikowane funkcjonalne techniczne szczegóły każdego modułu. W dalszej kolejności programiści używają tej dokumentacji projektowej i prototypów modułów do budowy kodu aplikacji. Stosuje się często

podejście iteracyjne dla każdego obszaru funkcjonalnego. Bardzo złożone aplikacje mogą wymagać pełnej dokumentacji technicznej, zanim rozpocznie się programowanie.

Konwersja danych

Celem procesu *konwersji danych* jest opracowanie zasad migracji, konwersji i testowania danych z dotychczasowych systemów. Dane te są konieczne do testowania oraz do pracy nowej aplikacji. Pierwszym krokiem w tym procesie jest określenie, jakie dane powinny być skonwertowane, w podziale na odpowiednie źródła. Dane te mogą być potrzebne do testowania systemu, testów integracyjnych, szkolenia, testów akceptacyjnych, jak również do bieżącego przetwarzania. Dlatego też zespół projektowy musi określić ogólną strategię, aby spełnić te wymagania. Strategia ta powinna uwzględniać równocześnie metody automatyczne i ręczne konwersji. Proces konwersji danych zawiera w sobie projektowanie, kodowanie i testowanie każdego koniecznego modułu konwersji danych.

Opracowanie dokumentacji technicznej

Proces *opracowania dokumentacji technicznej* dotyczy produkcji wysokiej jakości tekstowych dostaw zawierających dokumentację użytkownika końcowego, dokumentację techniczną i dokumentację szkolenia dla przedmiotowego projektu. Dwoma podstawowymi dokumentami są tu „Podręcznik użytkownika” (User Guide) oraz „Podręcznik funkcjonalności systemu” (User Reference). Pierwszy z nich (User Guide) oparty jest na modelu procesów systemowych w ujęciu modułów (Module Process Model). Jest on pomocą dla użytkownika w użyciu aplikacji w zakresie wykonania procesów biznesowych. Drugi (User Reference) specyfikuje funkcjonalność każdego modułu aplikacji.

Testowanie

Proces *testowania* określa zintegrowane podejście do testowania jakości wszystkich elementów aplikacji. W jego zakres wchodzi: testowanie modułów, testowanie integracyjne modułów, testy integracyjne całego systemu i testy akceptacyjne. Istotne jest opracowanie wspólnego planu wszystkich testów. Zaleca się użycie iteracyjne skryptów do testowania na coraz większych porcjach systemu aplikacyjnego.

Szkolenie

Celem procesu *szkolenia* jest wykreowanie użytkowników i administratorów, którzy są odpowiednio szkoleni w celu sprawnego wykonywania zadań obsługi nowej aplikacji. Może być również szkolony personel, który w przyszłości będzie miał za zadanie utrzymanie systemu, oraz zespół obsługi testów akceptacyjnych.

Przejście na nowy system

Opracowanie zasad przejścia na nowy system zaczyna się dość wcześnie w fazach projektowania przez zdefiniowanie specyficznych wymagań dotyczących rozpoczęcia eksploatacji nowego systemu. Produkty tego procesu to „Plan instalacji” i „Określenie środowiska i otoczenia eksploatacji”.

Obsługa serwisowa

Są cztery cele obsługi serwisowej, którą prowadzi się po wdrożeniu systemu:

- ♦ monitorowanie i reakcja na problemy za pośrednictwem *help desk*,
- ♦ naprawa błędów oraz rozwiązywanie problemów dotyczących wydajności przetwarzania (*upgrade*)
- ♦ ocena systemu podczas przetwarzania użytkowego,
- ♦ planowanie zmian w systemie.