

Inżynieria oprogramowania

Inżynieria oprogramowania jest wiedzą techniczną dotyczącą wszystkich faz cyklu życia oprogramowania. Traktuje oprogramowanie jako produkt, który ma spełniać potrzeby techniczne, ekonomiczne lub społeczne.

Dobre oprogramowanie powinno być:

- zgodne z wymaganiami użytkownika,
- niezawodne, poprawne
- efektywne,
- łatwe w konserwacji,
- ergonomiczne.

Zagadnienia inżynierii oprogramowania:

- Sposoby prowadzenia przedsięwzięć informatycznych.
- Techniki planowania, szacowania kosztów, harmonogramowania i monitorowania przedsięwzięć informatycznych.
- Metody analizy i projektowania systemów.
- Techniki zwiększania niezawodności oprogramowania.
- Sposoby testowania systemów i szacowania niezawodności.
- Sposoby przygotowania dokumentacji technicznej i użytkowej.
- Procedury kontroli jakości.
- Metody redukcji kosztów konserwacji (usuwania błędów, modyfikacji i rozszerzeń)
- Techniki pracy zespołowej i czynniki psychologiczne wpływające na efektywność pracy.

Kryzys oprogramowania:

- duża złożoność tworzonych SI
- niepowtarzalność i wielod dziedzinowość projektów
- „niewidzialność” SI i procesów ich wytwarzania
- długi czas wytwarzania SI w warunkach stałych zmian w otoczeniu
- pozorna łatwość wprowadzania poprawek

Złożoność SI:

- duża liczba elementów, złożone związki
- środki techniczne i programowe – złożoność i rozwój
- użytkownicy – różnorodność, zmienność wymagań, skłonność do błędów, tajność
- zespół wykonawczy – umiejętności, czas, środki, komunikacja

Walka za złożonością

Zasada dekompozycji i hierarchiczności:

rozdzielenie złożonego problemu na podproblemy, które można rozpatrywać i rozwiązywać niezależnie od siebie i całości.

Zasada abstrakcji i strukturyzacji:

ukrycie lub pominięcie mniej istotnych szczegółów danego przedmiotu lub mniej istotnej informacji; wyodrębnianie cech wspólnych i niezmiennych dla pewnego zbioru bytów

Zasada ponownego użycia:

wykorzystanie wcześniej wytworzonych schematów, metod,

Zasada sprzyjania naturalnym ludzkim własnościom:

dopasowanie modeli pojęciowych i modeli realizacyjnych systemów do wrodzonych ludzkich własności psychologicznych, instynktów oraz mentalnych mechanizmów percepcji i rozumienia świata.

Wiele aspektów opisu: (złożony system można opisać w skali jednorodnego opisu różnych projektów)

Pojęcia modelowania pojęciowego oraz modelu pojęciowego odnoszą się procesów myślowych i wyobrażeń towarzyszących pracy nad oprogramowaniem. Mod. pojęciowe jest wspomagane przez środki wzmacniające ludzką pamięć i wyobraźnię.

Notacja jest formą zapisu

Rodzaje notacji:

- język naturalny
- notacje graficzne
- specyfikacje – ustrukturalizowany zapis tekstowy i numeryczny

Funkcje:

- narzędzia pracy analityka projektanta, zapis i analiza pomysłów
- współpraca z użytkownikiem
- komunikacja z innymi członkami zespołu
- podstawa implementacji oprogramowania
- zapis dokumentacji technicznej

Technika sposób wykorzystania narzędzi do rozwiązywania problemów

Metoda sposób posługiwania się i wykorzystania techniki

Metodyka jest to zestaw pojęć, notacji, modeli, języków, technik i sposobów postępowania służący do analizy dziedziny stanowiącej przedmiot projektowanego systemu oraz do projektowania pojęciowego, logicznego i/lub fizycznego.

Metodyka jest powiązana z notacją służącą do dokumentowania wyników faz projektu

Metodyka określa:

- fazy projektu, role uczestników projektu,
- modele tworzone w każdej z faz,
- scenariusze postępowania w każdej z faz,
- reguły przechodzenia od fazy do następnej fazy,
- notacje, których należy używać,
- dokumentację powstającą w każdej z faz.

Typy metodyk:

- strukturalna - obiektowa - mieszana

CASE narzędzia informatyczne, które wspomagają proces wytwarzania oprogramowania na wszystkich jego fazach

Dzieli się na:

- Upper-CASE (dla wszystkich etapów)
- Lower-CASE (wspomaganie implementacji)
- Integrated-CASE (wszystkie fazy)

Struktury SI:

- * funkcjonalna (cele, funkcje)
- * informacyjna (jakie dane są wprowadzane i wyprowadzane)
- * techniczna (jaki sprzęt potrzebny do realizacji funkcji)
- * przestrzenna (w którym punkcie ma co być)
- * oprogramowanie (zbiór programów narzędziowych i użytkowych)

Budowa SI:

- analiza - badanie stanu istniejącego
- projektowanie – rezultatem jest projekt czegoś nowego (spojrzenie w przyszłość)
- implementacja – wytwarzanie
- wdrożenie

Cykl życiowy oprogramowania

proces złożony z ciągu wzajemnie spójnych etapów

pozwalających na pełne i skuteczne stworzenie a następnie używanie SI, obejmuje okres od momentu uświadomienia potrzeby istnienia systemu do momentu jego wycofania z eksploatacji

Fazy:

- Faza strategiczna: określenie strategicznych celów, planowanie i definicja projektu
- Określenie wymagań
- Analiza: dziedziny przedsiębiorczości, wymagań systemowych
- Projektowanie: projektowanie pojęciowe, projektowanie logiczne
- Implementacja/konstrukcja: rozwijanie, testowanie, dokumentacja
- Testowanie
- Dokumentacja
- Instalacja
- Przygotowanie użytkowników, akceptacja, szkolenie
- Utrzymanie, konserwacja, pielęgnacja

Modele cyklu życia oprogramowania:

1. Model kaskadowy (wodospadowy) – zakłada stabilny zestaw potrzeb i ich niezmiennosc w trakcie budowy SI, (po każdym etapie powstaje dokument, sprawdzenie po pozytywnej weryfikacji następny etap)

2. Model spiralny

Fazy: - planowanie - analiza ryzyka - konstruowanie
- weryfikacja

3. Prototypowanie

Prototyp – model SI działający lub demonstrujący działanie przyszłego systemu. Obszary działania (szybkie sprawdzenie koncepcji systemu; projektowanie przez prototypowanie; budowa przez prototypowanie)

4. Montaż z gotowych komponentów

(projekt; zakup elementów od dostawców; integracja, łączenie ich w SI) Zalety: wysoka niezawodność, zmniejszenie ryzyka, efektywne wykorzystanie specjalistów

Wady: dodatkowy koszt przygotowania elementów ponownego użycia, ryzyko uzależnienia się od dostawcy

5. Przyrostowy (brak wydzielonej jawnej fazy ryzyka)

Uwarunkowania doboru metodyki projektowania SI:

- wielkość przedsięwzięcia
- im dłuższy czas realizacji tym większe udokumentowanie (czasookres realizacji przedsięwzięcia)
- współpraca z użytkownikami
- wielkość i umiejętności zespołu projektującego SI
- dostępność narzędzi określa wybór metodyki
- ocena opracowania systemu

Rezultaty wyboru metodyki:

§ kolejność i zakres prac

§ zawartość dokumentacji

§ koszty

§ metodyki modelowania i projektowania, notacje

Faza strategiczna jest wykonywana zanim podejmowana jest decyzja o realizacji przedsięwzięcia. Nazywana także strate-gicznym planem rozwoju informatyzacji (SPRI) lub studium osiągalności.

Zadania, rezultaty:

- określenie celów
- określenie zakresu i kontekstu (otoczenie)
- ogólne określenie wymagań
- oszacowanie kosztów i ryzyka
- wstępny harmonogram
- określenie standardów

Analiza SI:

rezultat: sformalizowany model obszaru informatycznego

metody: obserwacja, wywiady, dzienniki, dyskusje,

Metody opisu: słowny, tablice decyzyjne, modele procesów

Określenie wymagań

- wymagania funkcjonalne (przetwarzanie) i нефункционалне
- poziomy opisu:
- definicja wymagań – ogólny opis
- specyfikacja wymagań – diagramy funkcji, przypadków użycia
- specyfikacja oprogramowania – formalna
- rezultat: co system ma robić
- rezultat: logiczny model systemu

Modelowanie

rezultat: logiczny model systemu

techniki modelowania:

- strukturalne (diagramy przepływu danych i diagramy związków encji)
- obiektowe (diagramy klas i obiektów, diagramy iteracji i przejść stanów)

Projektowanie

rezultat: jak system ma być zaimplementowany?

obszary:

- interfejs użytkownika

- fizyczna struktura BD i kodu programu
 - optymalizacja obiektu, dostosowanie do wymagań środowiska
- kodowanie danych
zapewnienie przyjazności SI
projektowanie formularzy (papierowych)
identyfikacja czynności manualnych

Implementacja

wybór języka (proceduralne, deklaratywne)
wykorzystanie gotowych elementów
narzędzia RAD

Testowanie

cel:

- wykrycie i usunięcie błędów
- ocena niezawodności systemu

sposoby testowania:

statyczne, dynamiczne, dowód poprawności

Dokumentowanie

wytwarzanie dokumentów dla różnych odbiorców (autorów, użytkowników, administratorów SI)
przeznaczenie:

- dokumentowanie budowy SI
- Uczy użytkownika i administratora
- pomoc w rozwiązywaniu problemów (instalacja, użytkowanie)

Instalacja

Instalacja sprzętu, systemu operacyjnego, oprogramowania wspomagającego BD i SI w jedną całość

rezultat: SI gotowy do użycia

Wdrożenie

rezultat: SI staje się nierozdzielalną częścią SI organizacji

zadania:

- szkolenia
- dostosowanie do konkretnych wymagań
- napełnienie Bazy danych
- bilans otwarcia
- przekazanie do eksploatacji

Konsekwencja i rozwój

Modyfikacja SI (poprawianie błędów, ulepszanie, dostosowanie do zmian wymagań i potrzeb użytkownika)

Modyfikacja powoduje: naruszenie struktury

wnoszenie błędów

Likwidacja SI

Musi być zaplanowany proces likwidacji

Zwykle jest powiązany z narodzinami nowego SI

Problemy:

- zapewnienie dostępu do danych archiwalnych przez długi czas
- migracja danych do nowego systemu (ludzie, sprzęt)

System może być zdefiniowany jako spójny zbiór niezależnych składowych które:

- istnieją w jakimś celu
- mają pewną stabilność
- są powiązane między sobą

Wejście (zasoby) które system pozyskuje ze swojego otoczenia lub z innych systemów

Wyjściami systemu jest to co system dostarcza do otoczenia lub innych systemów

Analiza

Celem analizy jest rozpoznanie wszystkich aspektów rzeczy. (nie wprowadza żadnych zmian do SI)

Analiza = rozpoznanie + wyjaśnienie + modelowanie + specyfikowanie + dokumentowanie

Analiza włącza następujące czynności:

- Rozpoznanie, wyjaśnianie, modelowanie, specyfikowanie i dokumentowanie rzeczywistości lub problemu będącego przedmiotem projektu;
- Ustalenie kontekstu projektu;
- Ustalenie wymagań użytkowników;
- Ustalenie wymagań organizacyjnych
- Inne ustalenia, np. dotyczące preferencji sprzętowych, preferencji w zakresie oprogramowania, ograniczeń finansowych, ograniczeń czasowych, itd.

Metody analizy: obserwacja, wywiad, narady, dzienniki, dyskusje, analiza dokumentów przepisów, samodzielny opis

Notacja BNF

= składa się; + i; () opcjonalność;

{ } iteracja; [] selekcja; *...* komentarz;

| rozdzielanie alternatyw na liście;

@ oznaczenie elementu identyfikującego;

Zasady budowy MPB

1. zdarzenia wejściowe i wyjściowe
2. blok decyzyjny ma minimum 2 wyjścia
3. po weryfikacji występuje blok decyzyjny
4. dokumenty we i wy (zwrot strzałek)
5. połączenia pomiędzy poszczególnymi proces. w hierarchii
6. poprawność i jednoznaczność opisu
7. dokładność, abstrakcja, czytelność, wyczerpa. możliwości

W rezultacie prac analitycznych powstaje dokument który zawiera:

- wprowadzenie (opis) – cel, zakres, podmiot

- modele procesów biznesowych (diagramy, drzewa decyzyjne)
- tablice z opisami dokumentami we/wy
- specyfikacja zawartości informacyjnej tych dokumentów (BNF + opis danych elementarnych)
- inne dane (kontekst, organizacja, plany zmian, potrzeby)

Faza strategiczna (studium wierzytelności)

Zadania – rezultaty:

- określenie celów
- określenie zakresy i kosztorysu
- ogólne określenie wymagań
- oszacowanie kosztów i ryzyka
- wstępny harmonogram

Studium wykonalności:

wykonalność techniczna

wykonalność ekonomiczna (efektywność)

wykonalność organizacyjna

wykonalność prawna

Opis zawartości dokumentu:

- metoda zstępująca – podział na kolekcje danych z opisem poszczególnych kolekcji
- nie definiować co już zdefiniowano
- nie definiować szaty graficznej, a tylko dane
- poprawnie i szczegółowo opisać elementy danych

Faza określenie wymagań

Celem fazy określenia wymagań jest ustalenie wymagań klienta wobec tworzonego systemu.

Dokonywana jest zamiana celów klienta na konkretne wymagania zapewniające osiągnięcie tych celów.

Klient rzadko wie, jakie wymagania zapewnią osiągnięcie jego celów.

Ta faza nie jest więc prostym zbieraniem wymagań, lecz procesem, w którym klient wspólnie z przedstawicielem producenta konstruuje zbiór wymagań zgodnie z postawionymi celami.

Czynniki uwzględniane przy definiowaniu wymagań:

- * możliwości systemu (zestaw funkcji systemu z pozycji użytkownika)
- * wielkość systemu (liczba użytkowników, objętość BD)
- * szybkość działania (czas realizacji operacji, natężenie operacji)
- * dokładność (liczba miejsc znaczących)
- * ograniczenia (zakres zmienności)
- * interfejsy komunikacyjne (sieci, protokoły)
- * interfejsy sprzętowe (sprzęt, wymagania środow. pracy)
- * interfejsy oprogramowania (kompatybilność)
- * interakcja człowiek-maszyna (interfejs użytkownika, sprzęt, język, komunikaty)
- * adaptowalność (zmiana)
- * bezpieczeństwo (poufność, prywatność, odporność)
- * odporność na awarie

- * standardy
- * zasoby (finansowe, ludzkie)
- * skala czasowa (ograniczenia czasu wykonania)

Trudność określenia wymagań:

- Klient z reguły nie potrafi zdefiniować celów i wymagań
- Cele klienta mogą być osiągnięte na wiele sposobów.
- Wielu użytkowników. Ich cele są często sprzeczne. Różni użytkownicy mogą posługiwać się inną terminologią.
- Zleceniodawcy i użytkownicy to często inne osoby – sprzeczność interesów, przewidywalność potrzeb
- niejednoznaczność języka

Poziom opisu wymagań:

1. definicja wymagań – opis ogólny
2. specyfikacja wymagań – częściowo ustrukturalizowany za-pis wykorzystujący zarówno język naturalny jak i proste, sformalizowane notacje
3. specyfikacja oprogramowania – w pełni formalny opis wy-magań

Metody rozpoznania wymagań

- Wywiady i przeglądy. Wywiady powinny być przygoto-wane (pytania) i podzielone na odrębne zagadnienia. Podział powinien przykrywać całość tematu i powinny być przepro-wadzone na reprezentatywnej grupie użytkowników. Wy-wiady powinny doprowadzić do szerokiej zgody i akceptacji projektu.
- Studia na istniejącym oprogramowaniu. Dość często nowe oprogramowanie zastępuje stare. Studia powinny usta-lić wszystkie dobre i złe strony starego oprogramowania.
- Studia wymagań systemowych. Dotyczy sytuacji, kiedy nowy system ma być częścią większego systemu.
- Studia osiągalności. Określenie realistycznych celów sys-temu i metod ich osiągnięcia.
- Prototypowanie. Zbudowanie prototypu systemu działają-cego w zmniejszonej skali, z uproszczonymi interfejsami.

Wykaz zdarzeń:

- typy zdarzeń:
 - pojawianie się danych na granicy systemu (do i z systemu)
 - wymuszane z zewnątrz lub wewnątrz (związane z upływem czasu)
 - sterowanie – systemowe (np. przekroczenie liczby)
- identyfikacja zdarzeń z punktu widzenia otoczenia systemu
- identyfikacja wyjątków i zdarzeń niepożądanych

Model funkcji:

- funkcja = działanie
- zwięźłość, jednoznaczność opisu, numerowanie funkcji
- na każdym poziomie ten sam poziom szczegółowości
- kompletność dekompozycji
- kolejność funkcji nie ma znaczenia
- funkcje z pozycji użytkownika

- podejście top-down

Wymagania нефunkcjonalne

Wymagania dotyczące produktu.

Np. musi istnieć możliwość operowania z systemem wyłącznie za pomocą klawiatury.

Wymagania dotyczące procesu.

Np. proces realizacji harmonogramowania zleceń musi być zgodny ze standardem opisanym w dokumencie XXXA/96.

Wymagania zewnętrzne.

Np. system harmonogramowania musi współpracować z bazą danych systemu komputerowego działu marketingu opisaną w dokumencie YYB/95. Niedopuszczalne są jakiegokolwiek zmiany w strukturze tej bazy.

Modelowanie struktur danych

Cele:

- ściśle określenie potrzeb informacyjnych obiektu rzeczywistego (firmy, działu)
- identyfikowanie rzeczy ważnych w analizowanym systemie (encji, obiektów) własności tych rzeczy (atrybutów) i sposobów jakimi te encje są za sobą powiązane (związków)
- dostarczenie dokładnego modelu potrzeb informacyjnych jako podstaw do budowy nowych systemów
- dostarczenie modelu niezależnego od sposobu przechowywania danych i od metod dostępu do nich

Diagramy związków encji Diagram prezentujący dane i związki logiczne pomiędzy nimi Na diagramie występują:

- encje o określonych właściwościach (atrybutach)
- związki (relacje, odniesienia)

Encja rzecz lub obiekt grupa, klasa, kategoria a nie konkretny mający dla nas znaczenie, rzeczywisty bądź wyobrażony, o którym informacje muszą być znane lub przechowywane

- każda encja musi być jednoznacznie identyfikowana
- każda instancja encji musi być wyraźnie odróżnialna od wszystkich innych instancji encji tego samego typu

Związki encji

Związek istotne powiązane między dwoma lub więcej encjami

Charakterystyka związku:

- stopień, liczność (jeden, wiele, oznaczenia 1:1, 1:n, n:m)
- nazwa (jednoznaczna)
- opcjonalność

Właściwości atrybutów:

autonomiczność, jednoznaczność, zależność tylko od instancji encji (klucza głównego), typ, format

Atrybut jest elementem danych. Może być jednorodny lub ko-lejką (np. Klient atrybut = nazwisko, imię, kod_pocztowy, ulica, @ pesel)

Algorytm budowy ERD:

1. identyfikacja (wydzielenie) zbioru obiektów (grup danych) w systemie wraz z ich atrybutami kluczowymi
2. identyfikacja powiązań bezpośrednich między obiektami (tablica krzyżowa) oraz ich rodzaju
3. przekształcenie tablicy krzyżowej powiązań w logiczny model danych i identyfikacja pozostałych atrybutów obiektów
4. przekształcenie każdego z powiązań typu m:n na dwa powiązania typu 1:n i identyfikacja dodatkowych atrybutów charakterystycznych dla nowo powst. obiektów
5. sprawdzenie poprawności otrzymanej struktury poprzez porównanie z wymaganiami systemu (encje kojarzące, łączące)
6. weryfikacja DFD względem ERD (tak by był szereg: encja – składnia danych)

Specyfikacja wymagań:

Cel tworzenia SI automatyzacja obsługi procesów ewidencji, rezerwacja, rozliczenia

Zakres

- gospodarka (ewidencja, remonty)
- ewidencja (np. klientów, wypożyczeń)
- rozliczenie
- rezerwacja
- cennik (zmiana cen)
- zestawienia wynikowe

Kontekst systemu (otoczenie)

współpraca z istniejącym systemem finansowo – księgowym

Techniki strukturalne:

- model procesów – diagramy przepływu danych (DFD)
- m. danych w systemie – związków encji (obektów) (ERD)
- model zdarzeń zachodzących w SI – diagramy historii życia obiektu (ELH)
- m. zmiany stanów systemu – diagram zmian syst. (STD)
- schematy struktury aplikacji (STC)
- słownik danych (grupuje opisy obiektów, definicje rekordów) część opisowa
- strukturalny Angielski – strukturalny polski

Metodyki strukturalne a obiektywne

- uważa się że wadą metodyk strukturalnych są trudności w zintegrowaniu metodyki
- metodyki strukturalne nie przesłają do obiektowej implementacji SI
- metodyki strukturalne są dojrzałe ale mogą nie być adekwatne do współczesnych tendencji produkcji SI
- uważa się że metodyki obiektowe dobrze nadają się do syst. czasu rzeczywistego gorzej do tradycyjnych transakcyjnych
- metodyki obiektowe wymagają dużego nakładu pracy
- metodyki obiektowe – młode bez dużych doświadczeń, szybko zmieniające się

Modelowanie procesów

Diagramy przepływu danych (DFD)

- strukturalna specyfikacja funkcji systemu
- identyfikacja zależności między procesami i danymi
- precyzyjne określenie zakresu systemu, podsys. i modułów
- zwięzły i czytelny opis funkcji systemu (łatwy do opanowania przez użytkownika)
- redukcja redundancji funkcjonalnej systemu

DFD elementy składowe:

- obiekt zewnętrzny (terminator) – daje albo pobiera dane do systemu
- proces – element przetwarzający wejściowe strumienie danych na wyjściowe
- magazyny danych – gromadzi i przechowuje dane
- przepływ danych – skąd dokąd dane przepływają i jaka jest struktura tych danych

Diagram kontekstowy – granice systemu

Diagram systemowy – diagram ogólny systemu (podsystemy + główne magazyny danych)

Diagramy procesów – rozwinięcie poszczególnych podsystemów, aż po procesy elementarne

Specyfikacja procesów elementarnych – mini specyfikacja, opis elementarnego algorytmu

Zasady budowy DFD:

- na DFD nie definiuje się sposobu w jaki obiekt zewnętrzny dostarczy lub pobierze dane
- magazyn jest elementem pasywnym do przechowywania danych, może mieć złożoną strukturę
- procesy powinny być wzajemnie niezależne
- DFD nie wyrażają zależności przyczynowo-skutkowych ani czasowych pomiędzy obiektami
- procesy muszą być nazwane (unikalny identyf., nazwa)
- przejrzystość i możliwość analizy (ilość procesów na diagramie : max 7 +/- 2)

Konstrukcje:

a/ proces aktualizuje proces (1 metoda przesyła do 2 dane)

b/ proces aktualizuje magazyn (magazyn strona bierna)

c/ proces czyta dane z magazynu

d/ przesyłanie danych do/z obiektów zewnętrznych

Techniki dekompozycji (podział procesu)

- podział na dwa lub więcej procesy
- mogą się komunikować za pomocą magazynów
- dwa nie związane między sobą procesy
- strumień danych też podlega dekompozycji

Błędy w strukturze

- „czarne dziury” procesy pochłaniające informacje

- źródła danych – proces bez wejść a z wyjściami
- „puchnące” magazyny – pochłaniają dane i nigdy się z nich ich nie usuwa
- stałe magazyny – tylko pobiera się z nich dane
- błędne przepływy: (magazyn – magazyn) (obiekt ze-wewnętrzny – magazyn)

Proces elementarny:

- algorytmiczna definicja procesu
- dowolność formułowania (ale zwykle: nr i nazwa procesu, dane we i wy, opis algorytmu)
- opis algorytmu: słowny, pseudokod, SQL, tablice decyzyjne

Bazy Danych

Dane w programie:

- Dane wewnątrz programów (we/wy), wykorzystane przez jednego użytkownika i w trakcie pojedynczej sesji
- nietrwałe, niedostępne dla wielu użytkowników

Dane – potrzeby aplikacji:

- Dużo danych
- Dane wspólne dla
 - o Wielu programów
 - o Wielu użytkowników tego samego programu
- Trwałość danych: długi czas życia

Dane w plikach – problemy:

- Współdzielenie danych – efektywność i konflikty
- Rozwiązanie. (warstwa pośrednia) System Zarządzania Danych – SZBD

Co to jest BD?

- Zorganizowany zbiór danych przechowywany w zewnętrznej pamięci komputera
- Odzwierciedlenie fragmentu rzeczywistości
- Cechy:
 - Trwałość
 - Zgodność z rzeczywistością

Pożądane właściwości BD:

- współdzielenie danych
- brak redundancji
- spójność
- bezpieczeństwo danych (przed utratą)
- poufność danych
- abstrakcyjność danych
- niezależność danych do programów
- niezawodność dostępu

Poziom odwzorowania danych:

- zewnętrzny (widoczny dla konkretnego użytkownika)

- konceptualny
- wewnętrzny (implementacyjny – z konkretnym syst. zarządzania BD, fizyczny – gdzie konkretnie jest umieszczony)

Zarządzanie SZBD:

- organizacja struktury BD (definiowanie schematu BD)
- konstruowanie BD (system plików)
- przetwarzanie danych:
aktualizacja danych (wprowadzenie, poprawianie, usuwanie)
- wyszukiwanie danych (zapytania do BD)
- administracja BD
- zapewnienie właściwości BD w praktyce

Specjalne cechy SZBD:

- * Transakcyjność przetwarzania
- * Optymalizacja przetwarzania
- * Blokowanie zasobów (rozwiązanie konfliktów dostępu)
- * Przeciwdziałanie zakleszczeniom
- * System kont i uprawnień dostępu
- * Monitorowanie BD

Języki BD:

- definiowania danych (DDL)
- manipulowania danymi (DML)
- zapytań (Query Language) – język raportowania
- sterowania danymi (DCL)

SZBD architektura:

- jednopoziomowa
- dwupoziomowa (klient – serwer)
- trójpoziomowa (serwer www -serwer aplikacji –serwer BD)
- rozproszona (wiele serwerów BD)

Własności BD:

- niezależność aplikacji i danych
- abstrakcyjna reprezentacja danych, wykorzystywanych przez aplikacje
- różnorodność widzenia danych przez różnych użytkowników (filtry, perspektywy)
- fizyczne i logiczne niezależności

Model implementacyjny – typy:

- Hierarchiczny - Sieciowy - Kartotekowy
- Relacyjny - Obiektowo – relacyjny - Obiektowy
- Hypertekst

Relacyjna DB składa się z (pojęcia podstawowe):

- Tablica, plik
- Atrybut, pole, kolumna
- Rekord, wiersz tablicy
- Typ danych

- Domena, dziedzina Wartość null
- Związki wartościowe (preferencje)

RDB – zasady:

- Każda tablica w BD ma jednoznaczną nazwę
- Każde pole (kolumna) ma jednoznaczną nazwę w tablicy
- Wszystkie wartości w kolumnie są tego samego typu
- Porządek kolumn i wierszy nie jest istotny
- Każdy wiersz musi być różny (wartościowo)
- Pola muszą zawierać wartości atomowe

Klucze:

- Klucz główny (Primary Key) – grupa kolumn o nie powtarzających się danych
- Klucz obcy (Foreign Key) – grupa kolumn z jednej tablicy, których wartości odpowiadają kluczowi głównemu innej tablicy (powiązanie)

Integralność:

- Poziom pól (dziedzina wartości)
- Poziom tablic (klucz główny)
- Integralność referencyjna:

o Obowiązkowość związku

o Ograniczone usuwanie

o Usuwanie kaskadowe

o Wstawianie null

- Integralność dynamiczna

Porządkowanie wierszy:

- Wiersz w tablicach – porządek historyczny
- Znaczenie dla interfejsu
- Fizyczne sortowanie – bardzo pracochłonna operacja z wykorzystaniem roboczych plików dyskowych
- Sortowanie logiczne (indeksowanie) – tworzenie i wykorzystanie tablic indeksowych - operacja dynamiczna

Transakcje na BD:

- Zmiana stanu BD
- Logiczna jednostka pracy w BD
- W trakcie trwania transakcji – BD nie jest spójna
- Właściwości transakcji (niepod., spójność, izol. i trwałość)
- Blokowanie – podstawa realizacji transakcji w środowisku współbieżnym

Transakcja a awaryjność w BD:

Ř Transakcje zatwierdzone – mają być odtworzone

Ř Transakcje nie zatwierdzone - wycofane

Ř Metoda osiągnięcia

o Dzienniki transakcji

o Redundancja

Mapowanie modelu konceptualnego na implementacyjny:

Ř Konceptualny (niezależny od SZBD, język programowania modelu bazy danych) – ERD

Ř Implementacyjny – fizyczny(w konkretnym modelu bazy danych i SZBD)

Model implementacyjny służy do wygenerowania skryptu do tworzenia BD wraz z więzami

integralności (słownik BD)

Pojęcie relacyjnego modelu implementacji:

- Ř Tabela (odpowiednik encji, nazwa l mnoga)
- Ř Kolumna - pole (odpowiednik atrybutu)
- Ř Dziedzina (konkretny typ danych i jego parametry)
- Ř Rekord (wystąpienie)
- Ř Indeks
- Ř Klucz główny (indeks unikalny, klucze sztuczne)
- Ř Klucz obcy (indeks nieunikalny)
- Ř Odniesienie (klucz główny - obcy, więzy integralności referencyjnej)
- Ř Klucz alternatywny (indeks unikalny lub nie)

Atrybuty rozszerzone (nie SQL owe) – specyficzny dla danych SZBD:

- o Dodatkowe typy, opisy, etykiety, elementy, wyświetlane na ekranie (komunikaty, helpy)
- o Wartości domyślne
- o Rozróżnialność (lub nie) wielkości znaków
- o Procedury walidacyjne (trigery)
- o Typ indeksu (np. słowny), opis, sposób konstrukcji indeksu
- o Sekwencyjne

Tworzenie modelu implementacji:

- Ř Generowanie z modelu conceptualnego (mapowanie modelu conceptualnego na implementacyjny)
- Ř rewers ze schematami istniejącymi w BD

Problem mapowania:

- Ř nazewnictwo (identyfikatory, nazwy typów, dziedziny)
- Ř ograniczenia ilościowe (np. na liczbę pól w rekordzie)
- Ř brak wielowartościowości pól (płaski model)
- Ř brak zmiennej struktury rekordu (wiersza)

Mapowanie proste

- Ř encja → tabela
- Ř atrybut → pole (kolumna)
- Ř unikalny identyfikator → klucz główny
- Ř związki → klucze obce (dodawane do encji)

Mapowanie złożone:

- Ř mapowanie złożone nieoczywiste (związki wielu encji)
- Ř mapowanie na pojedynczą tabelę (kodowane)
- Ř mapowanie na oddzielną tabelę
- Ř mapowanie związków wykluczających się

Klucze:

- Ř każda encja może mieć wiele unikalnych identyfikatorów – są to klucze kandydujące
- Ř spośród kluczy kandydujących wybiera się główny
- Ř jeżeli brak klucza kandydującego tworzy się klucz sztuczny (generowany automatycznie)

Wybór klucza zlecenia:

Ř klucz główny powinien być jednym atrybutem

Ř klucz główny nie powinien mieć znaczenia w dziedzinie przedmiotowej

Ř klucz główny powinien być generowany automatycznie (RecLD, OJD)

Ilościowe aspekty danych:

Ř informacje charakterystycznych ilości danych

o zajętość pamięci(liczba wystąpień w BD)

o zmienność (prognozowany przyrost w czasie)

o wypełnienie pól wartościami

Ř informacje charakteryzujące dostęp

o wymagana szybkość dostępu

o zakres przetwarzania

Operacje częste (konto klienta) w BD:

- dodanie nowej operacji

- zmiana stanu konta

- zapytanie o stan konta

rzadkie operacje:

- zmiana danych adresowych klienta

Modyfikacja – samodzielne:

- Wprowadzanie pojęcia jadłospis (ustalony zbiór pożywienia, wielokrotne wykorzystanie i identyfikacja)

- Wprowadzenie konieczności przypisania zwierzęcia do gatunku, grupa itd.

- Wprowadzenie słowników: jednostek miar, trybu zatrudnienia pracowników, klasyfikacja dostawców na grupy

Aspekty modelowanie SI:

- funkcjonalny (DFO, hierarchiczny model funkcji) - co zachodzi w systemie

- danych (ERD, LDS, obiektowy) – na czym zachodzi

Diagram historii życia encji ELH

Odwzorowanie zmian stanów obiektów (encji) w czasie:

Ř Oddziaływanie zdarzeń z diagramem DFD na encje ERD

Ř Dynamiczny aspekt istnienia obiektu w systemie

Modeluje los pojedynczej encji (obiektu)

Chronologia zdarzeń mających wpływ na encje

Umożliwia identyfikację wszystkich zdarzeń związanych z encją

Umożliwia zdefiniowanie wszystkich błędnych sytuacji oraz zdarzeń wyjątkowych

ELH – elementy składowe:

Obiekt wybrany obiekt (encja) z ERD

Zdarzenie sekwencyjne pojedyncze zdarzenie związane z obiektem

Zdarzenie złożone zdarzenie, będące korzeniem drzewa opisującego jego strukturę

Zdarzenie powtarzalne zdarzenie, które może zajść więcej niż raz

Zdarzenie selektywne zdarzenie, którego zajście jest uzależnione od spełnienia pewnych warunków

Diagram historii życia obiektu (encji) – etapy budowy

Ř Etap1 przygotowawczy

- o Wybranie obiektu z ERD
- o Identyfikacja zdarzeń dotyczących danego obiektu na podstawie DFD
- o Dodatkowe rozważenie funkcji
- Ř Etap 2 – dla każdego obiektu z ERD
- o Normalny cykl życia obiektu
- o Zdarzenie specjalne (wyjątkowe)
- o Sytuacje błędne

Kolejność budowy

1. wybór zdarzeń oddziałujących na obiekt
2. ustalenie sekwencji zdarzeń
3. sprawdzenie czy pewne zdarzenia mogą zachodzić warunkowo (selektywnie)
4. sprawdzenie czy pewne zdarzenia mogą powtarzać się wielokrotnie
5. sprawdzenie i ewentualna korekta sekwencji zdarzeń pod wpływem iteracji
6. sprawdzenie czy system jednakowo traktuje wszystkie iteracje zdarzeń danego typu

ELH – identyfikacja sytuacji wyjątkowych i weryfikacji

- o gdzie mogą wystąpić sytuacje wyjątkowe?
- o Czy i w jaki sposób są one rejestrowane w systemie
- o Jakie są efekty uboczne sytuacji wyjątkowej (tj. zdarzenia realizowane w systemie)
- o Weryfikacja ELH względem DFD Dla konkretnego zdarzenia na ELH musi istnieć co najmniej jeden przepływ na DFD mu odpowiadający

Modelowanie zależności przyczynowo – skutkowych:

- o Diagram związku sferami STD (sfera Transmission Diagram)
- o Uzupełnienie DFD o zależności czasowe
- o Ważne dla stanów czasu rzeczywistego ale też dla systemów rozproszonych
- o Identyfikacja zdarzeń w systemie i ich zależność przyczynowo - skutkowych

STD – elementy składowe:

Stan systemu zbiór określonych wartości atrybutów systemu – stan w danym momencie czasu
 Przejście zmiana stanu systemu (tj przejście z jednego stanu do drugiego w wyniku określonego warunku(C) i wykonania akcji(A).

Zasady sporządzania STD

- o System musi się zawsze znajdować w jakimś stanie
- o Sprzęgi wejściowe (start) i wyjściowe (stop) – tylko jedno
- o Każdy stan systemu musi być dostępny ze stanu początkowego
- o Stan końcowy powinien być dostępny dla każdego stanu
- o Dany warunek powinien powodować przejście z każdego stanu tylko do jednego innego stanu (jednoznaczność przejścia)

Bilansowanie modelu

- q Różne przekroje modelu projektu SI
- o Funkcjonalny (procesowy) DFD,STC
- o Informacyjny (danych) – ERD LDS

o Zdarzeniowy (stanów) – ELH STD

q Problem: obiekty i nazewnictwo, struktura , zawartość

q Cel – utrzymanie spójności projektu i usunięcie błędów

Dostosowanie implementacji do ograniczeń i uwarunkowań

· Rozmiary danych(bieżące , przyrost – ok. 10 % rocznie)

· Czas odpowiedni (określa różne typy wejść)

· Ograniczenie pozamerytoryczne (dostawca sprzętu , SZBD)

· Ograniczenia środowiska

· Niezawodność (średni czas między awariami MTBF, średni czas naprawy NTTR)

· Bezpieczeństwo systemu

Projektowanie szczegółowe

q Kodowanie informacji

q Interfejs użytkownika (komunikacja)

q Formaty dokumentów we\wy

q Urządzenia we\wy

q Struktura techniczna i przestrzenna

q Problemy eksploatacji (autoryzacja dostępu)

q Elementy projektu poziomu fizycznej aplikacji

q Czynności ręczne (procedury)

Cele kodowania danych

q Zmniejszenie ryzyka błędów i pomyłek

q Uproszczenie i skracanie wprowadzania danych

q Przeciwdziałanie redundancji

q Uproszczenie i przyspieszenie przetwarzania

Rodzaje kodów

q Zewnętrzne (pesel , nip ,kod pocztowy)

q Wewnętrzne (Id_pracownika, nr + czesci,kod operacji)

q Dla celów projektu (moduły, oznaczenia)

Właściwe metody kodowania

q Rozszerzalność

q Kompletność

q Precyzja, jednoznaczność

q Zwięzłość

q Czytelność

q Wygoda użycia (łatwy do zapamiętania)

q Użyteczność (zgodność z istniejącymi)

Kod a maska

- Kod znacząca część (zmienna, niosąca informacje)

- Maska – sposób wyświetlania (format) część stała nie przechowywana jako dana w BD a jedynie w słowniku BD

Typ kodów

q Porządkowy (nr kolejny, np. 123)

q Klasyfikacyjny (pozycyjny np12.56.01)

q Mieszany (np. lu\02\0089)

q Kody z cyfrą kontrolną (np. .pesel)

q Kody mnemoniczne (np. NY, WAR)

q Kody alfabetyczne, numeryczne, mieszane

Definicja struktury kodu

INF/01/0234

kod kierunku numer roku (RR) Nr kolejny
ZP, AG INF

PESEL= RRMDD9999K (K- cyfra kontrolna)

Zalety cyfr kontrolnych:

- Wychwytywanie błędów w kodach bez konieczności odwoływania się do DB z zestawem kodów – pierwotna kontrola danych
- Wykrycie nieprawidłowo zdefiniowanego kodu na etapie jego tworzenia

Kody i symbole w projekcie

Standardowe oznaczenia:

- format daty, godziny (RR.MM.DD)
- opis pól rekordów: opis struktur pól w BD, projektach we/wy (zwykle zgodny z symboliką SZBD)
- symbole i oznaczenia obiektów (wydruków)

Oznaczenia obiektów projektu:

- nr/symbol opcji w układzie projektu
- kod podsystemu/modułu - symbol tabulogramu
- symbol dokumentu we/wy
- symbol/nazwa zbioru/pliku

Projektowanie interfejsu

GUI – Graficzny Interfejs Użytkownika

Komunikacja użytkownik – SI:

- sterowanie SI (polecenia)
- przepływ danych (użytkownik – system)

sterowanie SI – sposoby:

- Ř polecenia (linia komend)
- Ř klawisze sterujące (skrót)
- Ř opcje z menu
- Ř ikony (paski narzędziowe)
- Ř przyciski dialogu
- Ř działania myszą i innymi urządzeniami wskazującymi

Poziom użytkownika:

- początkujący (wstępne wyjaśnienia, pomocniki, pomoc kontekstowa, blokowanie)
- średnio zaawansowany (największa grupa, standard interfejsu, pomoc szczegółowa, indeks)
- ekspert (skrót, szybkość dostępu, indywidualizacja interfejsu)

Przepływ danych

Wejściowych (wprowadzanie):- parametry poleceń- odpowiedzi na pytania- dialog Wyjściowych:- wyświetlanie inf. o dialogu- raporty- graficzne prezentacje danych

Poprawny interfejs:

- Spójny (topologia, słownictwo, otoczenie)
- Prosta obsługa, ilość obiektów 5 –9
- Grupowanie (opcji, działań, kolejność/częstość)
- Możliwość skrótów w dostępie do funkcji
- Informacja o działaniach
- Odwołanie akcji
- Poczucie spełnienia (drobne kroki, informacja)
- Wdrażanie kontroli nad SI

Cele przyjazności SI (zadowolenie klienta):

- ograniczenie ilości pomyłek
- minimalizacja czasu i wysiłku uczenia się
- nie obciążanie pamięci krótkookresowej użytkownika

Architektura interfejsu graficznego (SDI, MDI)

Okna dialogowe:

- dezaktywują aplikację i wymuszają obsługę
- przyciski powrotu do aplikacji (przynajmniej jeden)

Istotne dla interfejsu graficznego – interaktywne projektowanie dialogu.

Problemy interakcyjności:

- tłumaczenia na inny język (problem: prostota stylu)
- zmienność kulturowa i prawna (problem: daty, waluty)
- niuanse kulturowe (problem: kolor np.czerwony, czarny)

Problem języka:

- gra słów (np. B2B – Biznes to Biznes)
- metafor
- skomplikowanego słownictwa
- zbyt długich nazw
- tekstów wewnątrz rysunków

Urządzenia realizacji interfejsu użytkownika:

WE:

- czytniki kart kodowych (perforowane, magnetyczne)
- nośniki magnetyczne i optyczne
- terminale
- urządzenia wskazujące (mysz, pióro)
- teletransmisja

WY:

- wydruki
- terminale
- karty kodowe
- nośniki magnetyczne i optyczne
- ploter, naświetlarka
- teletransmisja

Projektowanie – punkt wyjścia:

- typ formularza
- urządzenia techniczne
- sposób wypełniania
- kształt znaków, kolorów, wzorce

Typy formularzy:

- pojedyncze kartki - książeczki - rozdzielane
- wysyłkowe (w kopertach) - wyjściowe (druki w całości)

Identyfikacja czynności modularnych:

- przygotowanie danych do wprowadzania
- wykorzystanie rezultatów
- przetwarzanie ręczne
- zwiększenie niezawodności czynności manualnych (kodowanie, nadmiarowość, kontrola logiczna)

W projekcie powinno być:

- projekt struktury technicznej
- struktura przestrzenna (użytkownik, sprzęt, moduły)
- struktura oprogramowania (systemowe, narzędziowe, wspomagające)
- projekt elementów eksploatacji (prawo dostępu, zabezpieczenie danych)