

1. Životní cyklus projektu

Úvodní studie (nejčastěji SWOT analýza), globální analýza a návrh (vytvoření logického modelu), detailní analýza a návrh (vytvoření fyzického modelu), implementace a zavádění, provoz a údržba.

2. Model systému – co to je

Model – účelově a záměrně zjednodušené zobrazení objektivní reality (objektu skutečnosti) nebo zobrazení určité teorie, modelování = znázornění chování systému jiným systémem či výpočetním postupem. (viz IZIN1, pokud si to někdo pamatuje) Nebo jinak:

Softwarově zkonstruovaný vyšší stupeň abstrakce, který na určité podněty reaguje stejným způsobem jako systém reálný. Umožňuje odhalit chyby před vlastním vytvořením systému, při použití vhodných nástrojů automaticky generovat dokumentaci atd.

3. Funkční x datový (logický x fyzický)

Funkční model zajišťuje dynamický popis (funkce = co to dělá, tj. něco se děje).

Datový model zajišťuje statický popis (jaké jsou datové typy, kde ta data budou).

Logický (v případě těch našich ERD diagramů máme atributy entit, nezajímá nás, jak je ten atribut realizovaný)

Fyzický – závisí na implementaci – nejde už o atribut nějaké entity, ale o sloupec v nějaké tabulce – musí být jasně zadání, o jaký datový typ se jedná (číslo, datum, text)

4. Typy vývoje IS

Vodopádový typ = klasický přístup, viz otázka číslo 1.

Fontánový typ = varianta vodopádu - po dosažení cíle určité etapy se vracíme znovu na její začátek

Prototypový vývoj = uděláme si prototyp, pokud funguje uspokojivě, tak si ho necháme; pokud ne, tak ho postupně vylepšujeme tak dlouho, dokud nám to nestačí

Inkrementální typ = přírůstkový typ

5. Systémová analýza – metodologie x metoda

Systémová analýza je obor, který se zabývá analýzou a návrhem informačních systémů (to je docela laciná odpověď, ale musí stačit:)

Metodologie - jde o komplex postupů (metod) - poznávání, navrhování - v oblastech a) vědeckého výzkumu b) speciálních věd c) praktických.

Metoda - jedná se již o konkrétní postup, který vychází z nějaké metodologie nebo je její součástí a pomocí něhož se provede nějaká podstatná část životního cyklu projektu.

6. Nástroj – př. CASE pro strukt. x obj. modelování

Těmhle programům se říká CADy (Computer Aided Design, počítačem podporované návrhy). Pro strukturované modelování se používají různé CASE, jako třeba naše CASE Studio 2. Pro Objektově orientované návrhy to je např. Rational Rose (to je to, co jsme měli zapnutý dvakrát a nikdy to nefungovalo).

7. Modelování – přístupy: funkčně –procesní x cílový x dynamický

Funkčně - procesní přístup:

Funkčně orientovaný přístup - modeluje systém jako množinu spolupracujících funkcí, kdy systém je modelován pomocí procesů, které jsou spojené datovými toky a realizují transformace dat (DFD). K návrhu se používají DFD diagramy.

Datově orientovaný přístup - snaží se modelovat základní datové struktury, funkční stránka je zde méně podstatná. Datový model definuje konceptuální databázový (DB) model, který poskytuje prostředky pro zodpovězení dotazů nad DB. Většinou se používá u informačních systémů.

Objektově orientovaný přístup - systém je modelován jako množina spolupracujících objektů, kde operace působící na objekty jsou zapouzdřeny v samotných objektech.

Cílově orientovaný přístup: Používá se pro návrhy menších informačních systémů. Jde v podstatě o zobrazení vstupů transformovaných na výstupy. Potřebné jsou fyzické vstupy, změnové programy, extrakční programy a požadované fyzické výstupy.

Dynamicky orientovaný přístup: Modeluje chování systému v časové posloupnosti stavů systému, jsou založeny na okamžitém vyhodnocování stavů a následné okamžité reakci. (Př.: real-time systémy, které se používají k řízení technologických procesů, navádění raket, kosmickému výzkumu apod.).

8. Strukturovaná analýza (17.2. obrázky)

De Marco – modelování pomocí DFD, hierarchie těchto DFD od kontextového diagramu až k nejnižší úrovni - rozhraní mezi jednotlivými procesy tvoří datové toky. Datový slovník (DD - Data Dictionary) k popisu procesů, entit a vztahů mezi nimi. Neobsahuje ERD!

Gane/Sarson – už ERD. Modelování touto metodou bylo založeno také na použití diagramu datových toků (DFD), ale nově byla do modelování zahrnuta relační analýza pomocí ERD, která u DeMarca scházela.

Datový model Warnier/Orr (cílový) - modifikace cílově orientovaného přístupu. Z požadovaných výstupů se odvodí logický datový model, následně fyzický. Grafické nástroje, které tato metoda používá, jsou obdobou ERD diagramů a nazývají se diagramy entit. Procesy se popisují pomocí tzv. assembly-line diagramů.

Yourdan - Používá funkční model, datový model a model přechodů stavů na dvou základních úrovních: 1. Úroveň esenciálního modelu a 2. Úroveň uživatelského implementačního modelu. Esenciální model vyjadřuje konceptuální úroveň popisu systému. Hlavní myšlenkou tohoto modelu je pomocí funkčního, datového modelu a modelu přechodů stavů vyjádřit esenci (podstatu) systému.

9. Postup Yourdan. metodologie (shora dolů)

Tvorba esenciálního modelu: Vytvoříme seznam událostí => pro každou událost jeden proces v DFD, následuje vyvažování DFD – proces normalizace (viz otázka 14) - zohledňuje vedle ostatních implementačních požadavků i požadavky uživatele.

Tvorba implementačního modelu = návrh STD (diagramu přechod stavů).

10. Notace: totální vztah x parciální vztah

V ERD diagramu jde o to, zda mezi entitami vztah být musí (totální vztah) nebo nemusí (parciální). Např. uživatel knihovny může dostat upomínku, nebo taky nemusí => parciální vztah, v CASE Studiu je parciální vztah označen prázdným kolečkem u entity.

11. Vztah M:N (název, převod) kardinalita, 1:N, N:1

Vztah M:N Každá přednáška může mít více odkazů na literaturu, stejně tak každý odkaz může příslušet více přednáškám. (Zde jde o oboustranně parciální vztah, neboť spojení nemusí nutně existovat.). Každý vztah M:N se dá zavedením nové entity převést na dva vztahy 1:N. Tento převod je nutný, chceme-li datovou základnu implementovat v relační databázi.

Kardinalita - 1 autor napsal víc knih, jeden zákazník si může půjčit víc knih...

Vztah 1:N Každý předmět obsahuje více přednášek, každá přednáška však přísluší pouze jednomu předmětu.

12. Co je atribut

Atribut je údaj charakterizující entitu. Ve fyzické reprezentaci jsou atributy tvořeny sloupci tabulky v databázi - entita = tabulka (např. Studenti školy), atributy = Jméno, Příjmení, ID studenta atd.

13. Primární klíč

Primární klíč je atribut, který jednoznačně identifikuje prvek tabulky/entity (např. uživatel FISu – mohou tu být dva lidé jménem Jan Šlégr, ale FIS je rozliší podle loginu – slegrja1 a slegrja2, takže login by mohl být v tomto případě primárním klíčem, aby si uživatelé navzájem neležli do e-indexu).

14. Tvorba ERD – pravidla

Analýza reality => návrh systému entit, jejich atributů a vztahů. (tedy jaké atributy odpovídají skutečným vlastnostem reality, které jsou důležité a které ne) Následuje normalizace (vyvažování), běžně ve třech fázích – 1., 2. a 3. normální forma (existuje i 4. a 5., ale používají se jen zřídka). V případě detailního (fyzického) návrhu se ještě uvažuje o datových typech (implementace).

Postup shora dolů – postupujeme od entit k atributům

Postup zdola nahoru – na začátku máme atributy, které sdružujeme do entit, které propojujeme.

15. Relace – identifikační, neidentifikační

Identifikační relace se liší od neidentifikační zobrazením plnou čarou a hlavně tím, že entita, ve které jsme připojení ukončili, je druhou relací identifikována. To má za následek, že u druhé entity nelze nastavit nepovinnou účast ve vztahu (parcialitu). Identifikační relaci způsobí přenesení primárního klíče z jedné entity do entity navazující (ve směru šipky), který se v této entitě objeví jako tzv. cizí klíč.

Klíče: PK (primární klíč), PFK (Primary Foreign Key – cizí primární klíč, získaný identifikační relací z jiné entity), FK (Foreign Key – cizí klíč).

16. Tok datový x řídicí

Řídicí tok určuje návaznost, posloupnost procesů, datový tok reprezentuje tok dat;] (resp. jejich transformaci).

17. Evidence dat – pomocí čeho ji provádíme

DD (Data Dictionary), PS (kniha funkcí) – evidence dat

FSD (Function Structure Diagram, diagram funkční struktury), DFD (Data Flow Diagram, diagram datových toků) - evidence funkcí a procesů

ERD (Entity Relationship Diagram, diagram entit a jejich vztahů), OOD (objektový diagram, někdy též ORD) – evidence datové struktury

18. Pojmy:

Terminátor x aktor: Terminátor je v ERD diagramu při strukturovaném návrhu, aktor v Class Diagramu při objektově orientovaném návrhu. Obojí má význam uživatele systému (je buď na začátku nebo na konci).

Entita x třída objektů: Entita je v ERD, třída objektů v Class Diagramu. Ve fyzickém pohledu (v hotové databázi) má obojí význam tabulky (spojující data, která mají něco společného).

Datastore x Repository: Datastore je ERD, repository v Class Diagramu. Obojí má význam dočasného úložiště dat.

19. Databáze – typy

Hierarchická - skládá se z rodičů a potomků. Rodič je vždy jen jeden, ale může mít více potomků. Velká redundance dat a obtížně se v nich vyhledává, používána např. v 60. letech v programu Apollo.

Sít'ová – zdokonalení hierarchické koncepce, masově rozšířené v 80. letech.

Relační – máme my. Data jsou uspořádána do tabulek (relací), nad kterými jsou definovány přípustné operace. (pro pana Kořínka – ano, jedná se o relační algebru). Příkladem je SQL.

20. Logický x fyzický model

Logický model (konceptuální návrh) není závislý na implementaci (řeklo by se, že proměnné v něm mají neomezenou délku a univerzální datový typ).

Ve fyzickém modelu už je zapotřebí přemýšlet nad tím, jaký datový typ se použije (integer v Pascalu sice zabírá v paměti málo místa, ale pokud bych měl víc jak 32767 faktur, tak tu další už tam prostě nenarvu a musím upravit systém tak, aby používal datový typ, který pojme delší číslo, třeba longint).

21. Jazyk SQL (SELECT, ...) x UML

SQL = Structured Query Language, strukturovaný dotazovací jazyk, používá se (překvapivě) ve strukturovaném modelování. Jeho nejpoužívanějším příkazem je SELECT, který z databáze vybere prvky, které mají určitou vlastnost, specifikovanou jako parametr příkazu SELECT.

UML = Unified Modeling Language, sjednocený modelovací jazyk, se používá k objektově orientovanému návrhu informačních systémů.

22. Objektové modelování:

Booch (design) - zaměřený na design, implementaci a nasazení IS. Pro modelování statické stránky systému používá diagramy tříd a objektů. Pro dynamickou stránku používá diagramy interakce, stavové diagramy a také diagramy modulů, procesů a kategorie tříd.

Rumbaugh (USE CASE, z ERA) - vychází ze strukturovaného modelování (z ER diagramu, který byl rozšířen o objektové rysy). Později byl z této metodiky vypuštěn diagram DFD. Naopak byl do tohoto modelování vložen Use Case model, který pochází z metodiky Object-Oriented Software Engineering a používá se ve fázi analýzy k vymezení hranic systému a interakci systému s vnějším okolím (prostě je to takovej splácanec strukturovaného a objektového modelování v jednom).

23. OO – Jacobson (business), Coad-Yourdon (USE CASE, CLASS)

Jacobson - vhodná pro Business modelování, vznikla z metody **objectory** ("object" a "factory"), zaměřena především na tvorbu počítačově orientovaných systémů. Její podstatu tvoří Use Case modelování. Málo používaná, ale výchozí pro ostatní metodiky

Coad-Yourdon - nejpoužívanější v OOM. Vhodná v aplikacích typu klient/server. Model je v ní rozdělen na vrstvy: vrstva subjektů, třídy a objekty a struktury, atributy a služby.

vrstvy v architektuře C/S (obdoba log. modelu 3-vrstvé archit.): vrstva zajišťující interakci s uživatelem (Human Interaction Layer), vrstva problémové oblasti (Problem Domain Layer) a vrstva datových objektů.

24. Asociace, agregace, generalizace (Class diagram)

Asociace – vztah mezi dvěma (teoreticky i více) třídami v Class diagramu.

Agregace - asociace, kde objekt na jedné straně hraje daleko významnější roli než objekt na straně druhé. Typickými příklady je třída, která obsahuje třídu druhou nebo třída, jejíž metoda vyvolává metody druhé třídy.

Generalizace - atributy a metody jedné třídy jsou zároveň atributy a metodami třídy jiné. Atributy, metody a vztahy nadřazené třídy jsou plně zděděny ve třídě podřazené. V programování je generalizace nejčastěji interpretována jako dědičnost

25. Objekt – atributy, metody, činnosti

Pohled používaný v objektově orientovaném návrhu systému. V programu Rational Rose je v horní části obdélníku reprezentujícího objekt jeho jméno, vprostřed jeho atributy a úplně vespod vprostřed metody a činnosti (viz výborná přednáška slečny Čermákové ve složce UML).

26. Souvislost objekt – třída

Objekt je podmnožinou třídy, neboť objekty se stejnými nebo podobnými vlastnostmi jsou do tříd sdružovány. Lze i říci, že objekt je instancí třídy.