

# Analýza požadavků a tvorba objektového návrhu – postup, použité modely a diagramy.

Wednesday, May 29, 2013 4:55 PM

Analýza požadavků obsahuje tři typy aktivit:

- **Sběr požadavků:** komunikace se zákazníky a uživateli za účelem získání jejich požadavků na systém.
- **Analýza požadavků:** identifikování nejasných požadavků, nekompletních, nejasných, nebo protichůdných a následně řešení těchto nesrovnalostí.
- **Zaznamenání požadavků:** dokumentování požadavků v různých formách, jako běžný textový dokument, případy užítí (use case), nebo specifikace procesů → dokument specifikace požadavků

## Analýza a klasifikace požadavků

- Identifikace zúčastněných stran ("Stakeholderů")
- Případy užítí (Use Case)
- XP (Extrémní programování) – user stories

- **Funkční**
- **Mimofunkční**
  - Výkonostní
  - Designové
  - Zákonný rámec

### Lidé v analýze

- **Zákazník**
  - Externí, interní
  - Doménový expert
- **Zainteresovaný hráč**
  - Ředitel, investor, standardizační orgán, daňový poplatník...
  - Vliv na úspěch projektu
- **Analytik**
  - Hlavní úloha
    - Zprostředkovatel mezi zákazníkem a programátory
  - Dovednosti a vlastnosti
    - Komunikační schopnosti, naslouchání a pozorování
    - Vedení schůzek, organizační
    - Schopnost abstrakce, nadhled, tvořivost
    - Detailní znalost problémové oblasti
    - Psaný projev, modelování
- Requirements development
  - Elicit
  - Analyze, negotiate
    - Potential -> stable requirements
  - Document
  - Review
    - Baselined requirements
- Requirements management
  - Change management

### Metoda **FURPS** – model klasifikace funkčních a mimofunkčních požadavků

- F (functionality) – funkčnost
- U (usability) – užitečnost
- R (reliability) – spolehlivost
- P (performance) – výkon
- S (supportability) – rozšiřitelnost

### Kontextový model

- Zobrazení vztahů systému s okolními entitami
  - Systém jako černá skříňka
  - Aktéři, stakeholders
  - Ostatní systémy
- Rozsah systému
- Rozhraní na okolí
  - HCI, API, data

### 4(+1) kategorie požadavků

- Klasifikace funkčnosti do 4 kategorií
  - Jaké informace systém obsluhuje, udržuje
  - Jaké funkce poskytuje uživatelům
  - Jaké analýzy dat provádí
  - Jaké jsou interakce s jinými systémy
- Pro každý druh příslušné vlastnosti
  - Stačí jednoduché seznamy
- +1 = mimo rozsah zadání, doplňková funkčnost

### Vize produktu a rozsah projektu

- Co má vzniknout
  - Esenciální, klíčové, vysokourovňové požadavky
  - Problem behind the problem
  - Business case (proč se to tak chce)
    - Zdůvodnění výhodnosti-návratnosti projektu
  - Požadavky: co to má dělat
- Popis problému
  - Obchodní příležitost
- Přehled stakeholders a uživatelů
  - Vize zahrnuje pohled všech zainteresovaných účastníků
  - Kdo jsou zájemci o systém, potenciální konkurence
- Přehled očekávaných schopností a funkcí produktu

# Analýza požadavků a tvorba objektového návrhu – postup, použité modely a diagramy.

Typu požadavků:

- Funkční
- Mimofunkční

Lidé v jejich analýze:

- Zákazník
  - Může být externí nebo interní
  - Doménový expert
- Zainteresovaný hráč
  - Ředitel, investor, ...
  - Má vliv na úspěch projektu
- Analytik
  - Jeho hlavní úlohou je zprostředkovat kontakt mezi programátory a zákazníky
  - Musí mít dobré komunikační schopnosti, schopnost abstrakce, detailní znalost problémové oblasti, organizační dovednosti

Postup sběru požadavků

- **Elicit**
- **Analyze, negotiate**
- **Document**
- **Review**

Kategorie požadavků

- Podle toho, jaké informace systém obsluhuje, udržuje
- Jaké funkce poskytuje uživatelům
- Jaké analýzy dat provádí
- Jaké má interakce s jinými systémy

Kontextový model

- Zobrazení vztahů s okolními entitami
- Systém jako černá skříňka
- Zobrazuje vztahy se stakeholders + ostatními systémy včetně rozhraní

Vize produktu

- Co má výsledný produkt dělat
- Kdo jsou stakeholders
- Kdo jsou uživatelé
- Omezení
- Časový rámec plánu projektu

Model užítí

- Model = aktéři + případy užítí
- Případ užítí = sekvence akcí, které systém provede v důsledku nějakého podnětu zvenčí a jejichž výsledek je viditelný pro uživatele

Doménový model

- Pro modelování datové části
- Musíme od klienta zjistit, co tyto doménové třídy budou
- S těmito třídami budeme najisto počítat
- Doménový model -> logický datový model

Datový model

- Logický (konceptuální), fyzický – obvykle ERA, viz DB1
- Hodnoty atributů datových entit se vyvíjí v čase

CRUDL matice – Create, Rename, Update, Delete List – cílem je vědět kdo a jak manipuluje s údaji

Modely základních struktur systému

Glosář

- Pojmy a jejich vysvětlení, prevence nedorozumění

Doménový model

- Model základních entit a jejich vztahů, doménové objekty – diagram tříd

Model nasazení

- Vztah produktu k prostředí

- o Popis, kvalitativní charakteristiky, priority
- Omezení, standardy, závislosti vztahující se k projektu
- Rámec plánu projektu
  - o Časový rozsah

#### Model užití

- Popis požadavků na vnější funkčnost systému
  - o Jací uživatelé k systému přistupují
  - o Co to pro ně dělá
  - o Jak systém zpracovává požadavky
  - o Kde je hranice systému
- Model = aktéři + případy užití
  - o Kontext, primární funkčnosti
  - o Další iterace, podružná funkčnost
  - o Případy užití
    - o Co to je případ užití
      - Sekvence akcí, které systém provádí v důsledku nějakého vnějšího podnětu a které vedou k výsledkům viditelným pro některého jeho uživatele
    - o Aktéři ...
    - o Hledáme dialogy typu aktér-systém
    - o Základní popis případu užití: název, stručný popis účelu, základní kroky postupu, odkazy na zdrojové informace
    - o UML diagram případů užití

## Tvorba objektového návrhu

### Diagram aktivit

#### Diagram aktivit (UML)

- **akce** – atomické dále nedělitelné kroky
- **vnořené aktivity** – volání jiných procesů (aktivit), tyto aktivity mohou být reprezentovány dalším **diagramem aktivit**. Sekvenci jednotlivých kroků v diagramu aktivit určuje řídicí tok.

### Modelování datové části

#### Použití doménového modelu

- Komunikace
  - o Dorozumění s klientem
- Objektová analýza a návrh
  - o S těmito třídami můžeme najisto počítat
  - o Analýza přidá zodpovědnosti, detaily vlastností a chování
  - o Návrh je transformuje a přidá implementační třídy
- Doménový model -> logický datový model

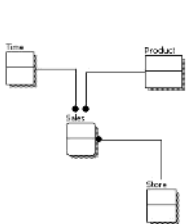
#### Datový model

The three level of data modeling, [conceptual data model](#), [logical data model](#), and [physical data model](#), were discussed in prior sections. Here we compare these three types of data models. The table below compares the different features:

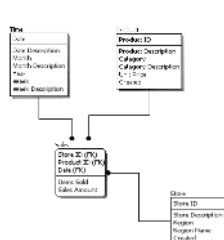
Feature	Conceptual	Logical	Physical
Entity Names	✓	✓	
Entity Relationships	✓	✓	
Attributes		✓	
Primary Keys		✓	✓
Foreign Keys		✓	✓
Table Names			✓
Column Names			✓
Column Data Types			✓

Below we show the conceptual, logical, and physical versions of a single data model.

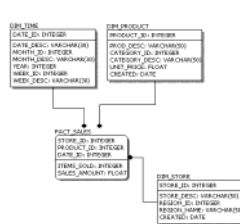
#### Conceptual Model Design



#### Logical Model Design



#### Physical Model Design



We can see that the complexity increases from conceptual to logical to physical. This is why we always first start with the conceptual data model (so we understand at high level what are the different entities in our data and how they relate to one another), then move on to the logical data model (so we understand the details of our data without worrying about how they will actually implemented), and finally the physical data model (so we know exactly how to implement our data model in the database of choice). In a data warehousing project, sometimes the conceptual data model and the logical data model are considered as a single deliverable.

- Logický (konceptuální), fyzický
  - o Obvykle jeden z prvních nezávisle na OO technikách (viz db1/db2/psds)
- Vývoj datových entit
  - o Datová entita typicky prochází vývojem (myšleno asi tak že hodnoty atributů entity se mění)
    - o Životní cyklus
  - o Model = stavový diagram
    - o Vazba na doménový / datový model
- CRUDL matice
  - o Cíl: vědět, kdo manipuluje s jakými údaji
  - o Create-Read-Update-DeleteList
  - o Úroveň detailu
    - o Analytická - uživatel, případ užití x informace
    - o Návrhová - třída, funkce, proces x tabulka
- Úplnost modelu požadavků
  - o Fáze zahájení projektu
    - o Přesně cíl/vize projektu
    - o Seznam klíčových aktérů, jejich cíle
    - o Seznam/diagram podstatných případů užití (dle cíle)
    - o Stručný popis klíčových PU, znalost klíčových vlastností
  - o Fáze projektování
    - o Kompletní seznam aktérů, popis důležitých
    - o Kompletní specifikace, 80-100% funkčnosti, stručné povědomí o všech
    - o Přesný popis mimofunkčních vlastností
- Shrnutí
  - o Kontext - vize, aktéři
  - o Funkčnost - případy užití, user stories, procesní model...
  - o Struktury - doménový model, stavový
  - o Vlastnosti - u PU, funkcí, doménových tříd, samostatně

## Product backlog

- Obrisy požadavků agilně
- Společné rysy agilních specifikací požadavků:
  - o Cílem je říct, produkt by měl umět tohle
- Forma
  - o Epic, user story
- Obsah
  - o Název, stručný popis, odhad pracnosti, testy
- **Vize produktu -> Feature (více než 2 sprinty) -> Epic (větší funkcionality, > 1 sprint) -> Skupina user stories**

## Modely základních struktur systému

### Glosář

- Seznam důležitých pojmů
  - o Klíčové
  - o Nejasné
  - o Sporné
- Stručný, všemi odsouhlasený popis = společný slovník, prevence nedorozumění
- Formát různý (word, excel, access...)

### Doménový model

- Popis struktury problémové oblasti
  - o Jaké jsou základní abstrakce používané v oblasti aplikace?
  - o Jaké mají názvy, vzájemné vztahy a vlastnosti?
  - o Jakým postupem je získáme?
  - o Podle čeho si máme vymyslet stabilní třídy pro realizaci?
- Východisko = glosář
- Model = doménové objekty (diagram tříd)

### Doménové objekty/třídy

- Doménové objekty - > třídy
  - o Věci vyskytující se v problémové doméně
  - o Klíčové pro fungování systému
  - o Systém udržuje informace

- Podstatné aspekty
  - o Terminologie uživatele, pojmy -> názvy tříd
  - o Jen základní obrysy
  - o Vztahy mezi třídami (asociace, kardinality)
  - o Nezávislost na implementaci
- Jak je najít? Doménovou analýzou (konzultace, doménový expert, části systému podstatné z jejich pohledu - pomůže obrázek, rozhovor s uživatelem, pozorování práce)

#### Model nasazení

- Vztahy produktu k prostředí
  - o Porozumění runtime a fyzickému prostředí
  - o Charakteristiky, parametry
  - o Odhad nákladů + podklad pro architekturu
- Alternativy
  - o Zelená louka - součást návrhu architektury
- UML diagram nasazení

#### Strukturální model

- Postup se zaměřuje na data a jejich transformaci pomocí procesů systému.
- Hlavními nástroji jsou tedy DFD popisující procesy a toky dat a ERD (Entity Relationship Diagram) popisující data a vztahy mezi nimi
- Zaměřuje se na vytvoření logického modelu nového navrhovaného systému (**esenciální model**) a následně přizpůsobení implementačním požadavkům (**implementační model**).
- Model **prostředí**
  - Definiuje hranice systému a okolí
  - Obsahuje kontextový diagram a seznam událostí (event list)
- Model **chování**
  - Definiuje vnitřní chování systému, tak aby plnil požadavky okolí
  - Používané modely (diagramy) DFD, ERD, DD (datový slovník), SP (specifikace procesů), případně STD (stavový diagram)

**Strukturovaná analýza.** Klasický postup "shora dolů". Chápe systém jako celek a postupně ho dekomponuje na jednodušší části. Je metodicky velmi dobře zvládnut. Nevýhodou je absence možnosti ukončit předčasně analýzu a implementovat konkrétní část systému. Typickým příkladem je metoda SSADM (Structured Systems Analysis and Design Method).

#### Objektový model

- Často se používá **modelovací jazyk UML**
  - o Model případů užití
  - o Doménový model
  - o Diagram tříd
  - o Stavové diagramy, sekvenční diagramy a další
- **CRC karty** (Class-Responsibility-Collaboration cards)
  - o Umožňuje zvládnout návrh i složitých a velkých systémů
  - o Není na první pohled vidět kudy vedou vztahy, musí se vyčíst z karet (barevné rozlišení, čáry na nástěnce...)
- Hierarchie objektů – sdružovány podle logických souvislostí do balíků a podbalíků
- Přirozený přechod od analýzy k návrhu

#### Doménová analýza

- Doménová analýza nepřihlíží podstatně k jednomu účelu a způsobu použití (kontextu použití), nýbrž shromažďuje pojmy a vazby pro doménu podstatné.
- Hledají se objekty, operace a vazby, které znalci z problémové oblasti pokládají za důležité (často používají jejich názvy apod.)

**Objektově orientovaná analýza.** Přístup "zdola nahoru", jehož základní myšlenkou je zobecňování. Začíná se z konkrétní části systému, která se analyzuje a popíše. V dalších částech se pak hledají společné rysy, z nichž se odvozují obecné zákonitosti. Výhodou je možnost rychlého paralelního vývoje prototypového software; nevýhodou je nedostatek ověřených metodických postupů a hlubší zkušenosti s rozsáhlejšími systémy.

From <<https://d.docs.live.net/e3534876709763a3/Dokumenty/ZCU/Statnice/Statnice.docx>>