



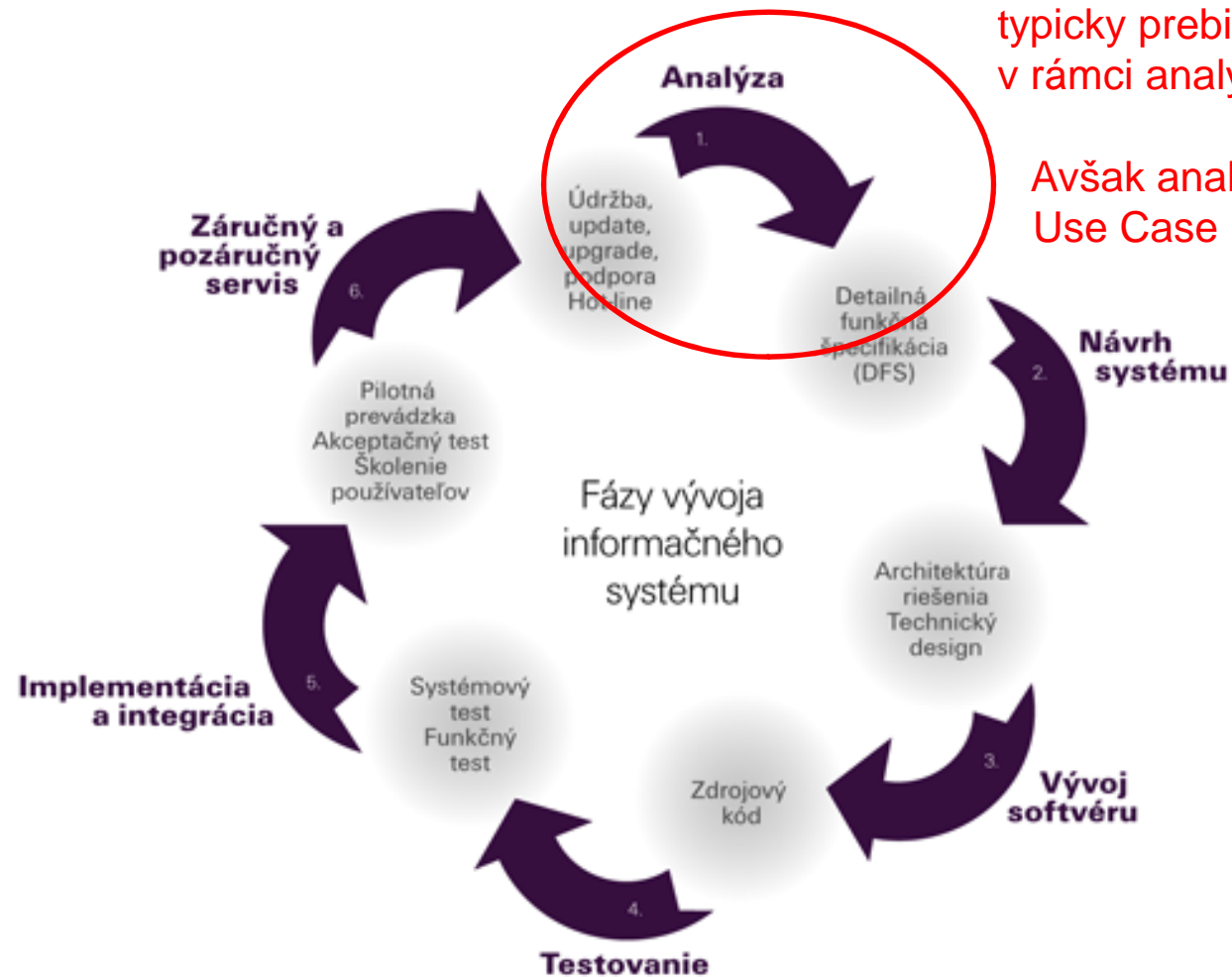
# Use Case modelovanie

.. a jeho zasadenie do SW procesu

Jana Dvořáková

5.10.2011, FMFI UK, OOAM cvičenia

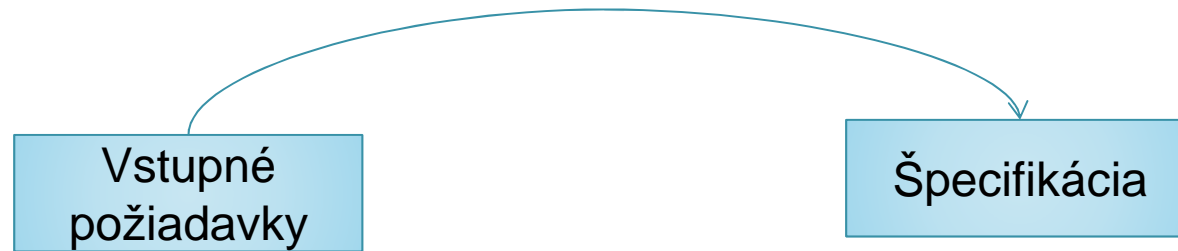
# Proces vývoja SW



Use Case modelovanie typicky prebieha v rámci analýzy

Avšak analýza nie je iba Use Case modelovanie!

# Analýza – vstup a výstup



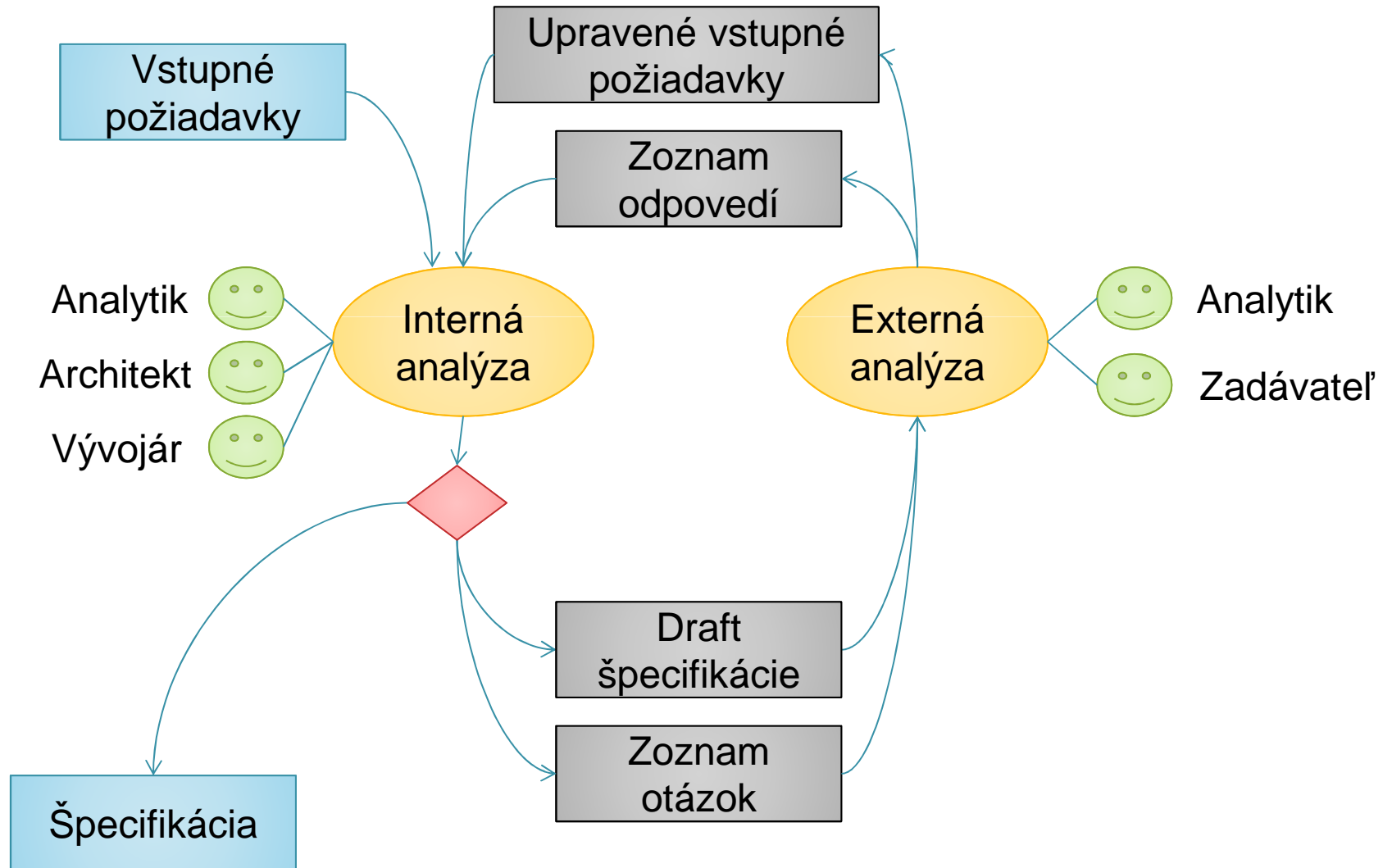
## Vstup:

- Dokument s požiadavkami od zadávateľa
  - Rôzna kvalita
  - „**NIKDY**“ nie je priamo použiteľný pre ďalšiu fázu (design)
  - Nejednoznačný, neúplný, so sivými miestami,..

## Výstup:

- Špecifikácia požiadaviek
  1. Funkčné aj nefunkčné požiadavky
    - Nefunkčné sú napr. požiadavky na performance (čas odozvy pri GUI,..)
  2. Požiadavky, ktoré sa nebudú realizovať
  3. Riziká
    - Jazyk musí byť zrozumiteľný pre zadávateľa (často človek z businessu)
    - Je schvaľovaná zadávateľom

# Analýza - proces





# Interná analýza

## Identifikovať predovšetkým

1. Nejasné požiadavky
2. Chýbajúce požiadavky
3. Nerealizovateľné požiadavky
4. Požiadavky v rozpore s konceptom
5. Drahé požiadavky
6. Požiadavky, ktoré sa nebudú realizovať
7. Riziká



# Externá analýza (1)

- Vykomunikovať identifikované problémové požiadavky so zadávateľom
- Dohodnúť sa na riešení

# Externá analýza (2)

1. Nejasné požiadavky
  - Dodefinovať požiadavky (jednoznačne, zrozumiteľne)
2. Chýbajúce požiadavky
  - Doplniť požiadavky / zaradiť ich do zoznamu požiadaviek, ktoré sa nebudú realizovať
3. Nerealizovateľné požiadavky
  - Vypustiť požiadavky / nahradiť ich alternatívnymi realizovateľnými požiadavkami
4. Požiadavky v rozpore s konceptom
  - Vypustiť požiadavky / ponechať ich za cenu zásahu do konceptu riešenia
5. Drahé požiadavky
  - Vypustiť požiadavky / ponechať ich s vedomím vysokej ceny ich realizácie / nahradiť ich alternatívnymi a lacnejšími požiadavkami
6. Požiadavky, ktoré sa nebudú realizovať
  - Vyplývajú z bodu 2 alebo internej analýzy

# Špecifikácia – funkčné požiadavky

- Popisuje žiadanú funkcionálnosť
- Typické prvky:
  1. Slovný popis
  2. UC model: Diagramy + popis
  3. Modely GUI (čiastočne popisujú aj nefunkčné požiadavky)
- UC model často predstavuje podporný prostriedok pri špecifikácii
  - „Lepidlo“, ktoré spája funkčnú špecifikáciu do celku
  - Diagram je jednoznačnejší ako slovný popis a vhodne ho dopĺňa
  - Častý problém: **zadávateľ nepozná UML...** 😞



# Use Case modelovanie

- Počas analýzy sa modeluje business pohľad na funkcionality, nie technický pohľad
- Zvoliť vhodnú úroveň granularity
- Generický pohľad na funkcionality systému, nemodeluje sekvenciu aktivít/krokov
- *Projekt: 20 - 25 UC*

Ďalej „best practices“ pre

1. Use Cases
2. Aktéri
3. Vzťahy
4. Ohraničenie systému

# Use Cases

- **Názvy**
  - Začínajú rozkazovacím slovesom
  - Napr.: **Otvor** objednávku, **Vytlač** objednávku
- **Jednotný doménový slovník** naprieč modelom
  - ~~Prihlás do systému, Odhlás z aplikácie~~
  - Prihlás do systému, Odhlás zo systému
- Zohľadniť **časový sled UC**
  - prvé UC hore, ďalšie pod ním
- Je možné do modelu doplniť **slovný popis UC**:
  - Cieľ, Počiatočné podmienky, koncové podmienky, Postupnosť krokov, ...

# Aktéri

- **Názvy:**
  - Podstatné mená, jednotné a zrozumiteľné z pohľadu businessu
  - Príklad: Zákazník, Dodávateľ, Tlačiareň
- Aktéri sú umiestnení vždy **mimo systému**
- Každý aktér je asociovaný z jedným alebo viacerými UC
- **Aktér = rola**, nie pozícia
  - Príklad: „Administrátor“, nie „Pracovník oddelenia správy zákazníckych aplikácií“
- Ak aktér predstavuje iný systém, je označený stereotypom **<<system>>**
- Aktéri navzájom **neinteragujú!**
- Je možné doplniť do modelu **slovný popis aktérov**

# Vzťahy

1. Asociácia medzi aktérom a UC
  - Musí byť namodelovaná, ak sa aktér vyskytuje v logike UC
2. Asociácie medzi dvomi UC
  - **<<include>>**
    - Je jasný moment spustenia volaného UC
    - Volaný UC sa kreslí napravo od volajúceho UC
  - **<<extend>>**
    - Rozširujúci UC sa môže volať v rámci viacerých krokov rozšíreného UC
    - Rozširujúci UC sa kreslí pod rodičovský UC
    - V praxi má túce a odporúča sa používať minimálne
3. Generalizácia medzi aktérmi
  - Dediaci aktér sa kreslí pod rodičovský aktér
4. Generalizácia medzi UC
  - Dediaci UC sa kreslí pod rodičovský UC
  - V praxi má túce a odporúča sa používať minimálne

# Ohraničenie systému

- Určuje hranicu systému
- Vnútorne UC reprezentujú funkcionality systému
- Napr. rozdelenie na releas-y
  - Časť systému, ktorá sa bude realizovať v Release 1, Release 2
- V praxi často ohraničenie všetkých UC v jednej „krabičke“