



# SOFTWAREOVÉ INŽENÝRSTVÍ

---

Business model

Martin Komárek



# Úrovně tvorby systému

---

- Business/byznys modelování.
- Analytické modelování.
- Návrhové modelování.
- Kódování.

Všechny úrovně musí být mezi sebou konzistentní.



## Význam úrovní

---

- Business model – Jak funguje organizace?
- Analytický model – Co IS dělá?
- Návrhové modelování – Jak to IS dělá?



# Rozdělení práce na projektu

---

- Metoda příčného řezu.
  - Programátor-analytik dostane k řešení určitou část problematiky.
- Rozdělení po úrovních.
  - Rozdělení rolí na analytiky, návrháře a programátory. Iterační a inkrementální vývoj.

Výhody a nevýhody?



# Business model

---

- Business Proces Model (BPM)  
= Model podnikových/obchodních procesů.
- Business Domain Model (BDM)  
= Model podnikových/obchodních entit.



# Business proces - Co to je ?

---

- Obchodní/firemní postupy.
- Po částech uspořádaná množina aktivit, které společně realizují podnikatelský nebo strategický cíl.

Detailněji např. na  
[http://www.cs.vsb.cz/radecky/files/radecky\\_technologie04.pdf](http://www.cs.vsb.cz/radecky/files/radecky_technologie04.pdf)



## Business procesy – dva pohledy

---

- AS IS – Jaké jsou postupy teď?
- TO BE – Jaké budou postupy po zavedení systému?



# Business procesy – proč popisovat?

---

- Pro potřeby tvorby nabídky.

Model (business) procesů slouží jako podklad pro odsouhlasení rozsahu aplikace klientem.

- Pro vylepšení (reengineering) procesů.
- Pro potřeby nasazení systému.
  - a) pro testery - tester může testovat i business logiku
  - b) pro tutorý - školitelé musí vědět nejen co dělá které menu, ale i k čemu systém slouží





# Business procesy – příklad

---

Př. "Přichází zákazník k přepážce banky" -> "Přijetí žádosti o hypotéku" -> atd... .

I bez znalosti bankovní problematiky dovodíme, že žádost bude pravděpodobně zadána do systému bankovní úřednicí.

Pokud bychom ale neměli v modelu zachyceno, že zákazník "Přichází k přepážce" mohli bychom si myslet, že zákazník může podat žádost o hypotéku i přes webové rozhraní nebo po telefonu.

I z tohoto triviálního příkladu je zřejmá důležitost zachycení business procesů v CIM modelu.



# Business procesy – modelování

---

Existuje mnoho standardů.

- Eriksson-Penker Business Extensions
- BPMN
- UML – diagram aktivit

Zdroje:

<http://www.sparxsystems.com/downloads/whitepapers/businessProcessModelTutorial.pdf>

<http://www.bpmn.org/>

[http://www.soc.staffs.ac.uk/kch1/teaching/short\\_course/unit\\_10\\_pro.ppt](http://www.soc.staffs.ac.uk/kch1/teaching/short_course/unit_10_pro.ppt)



# Business procesy – příklad v EA

---

Klíčové slovo „business“ v nápovědě EA.

Pozn.: EA ~ Enterprise Architect



## Business procesy

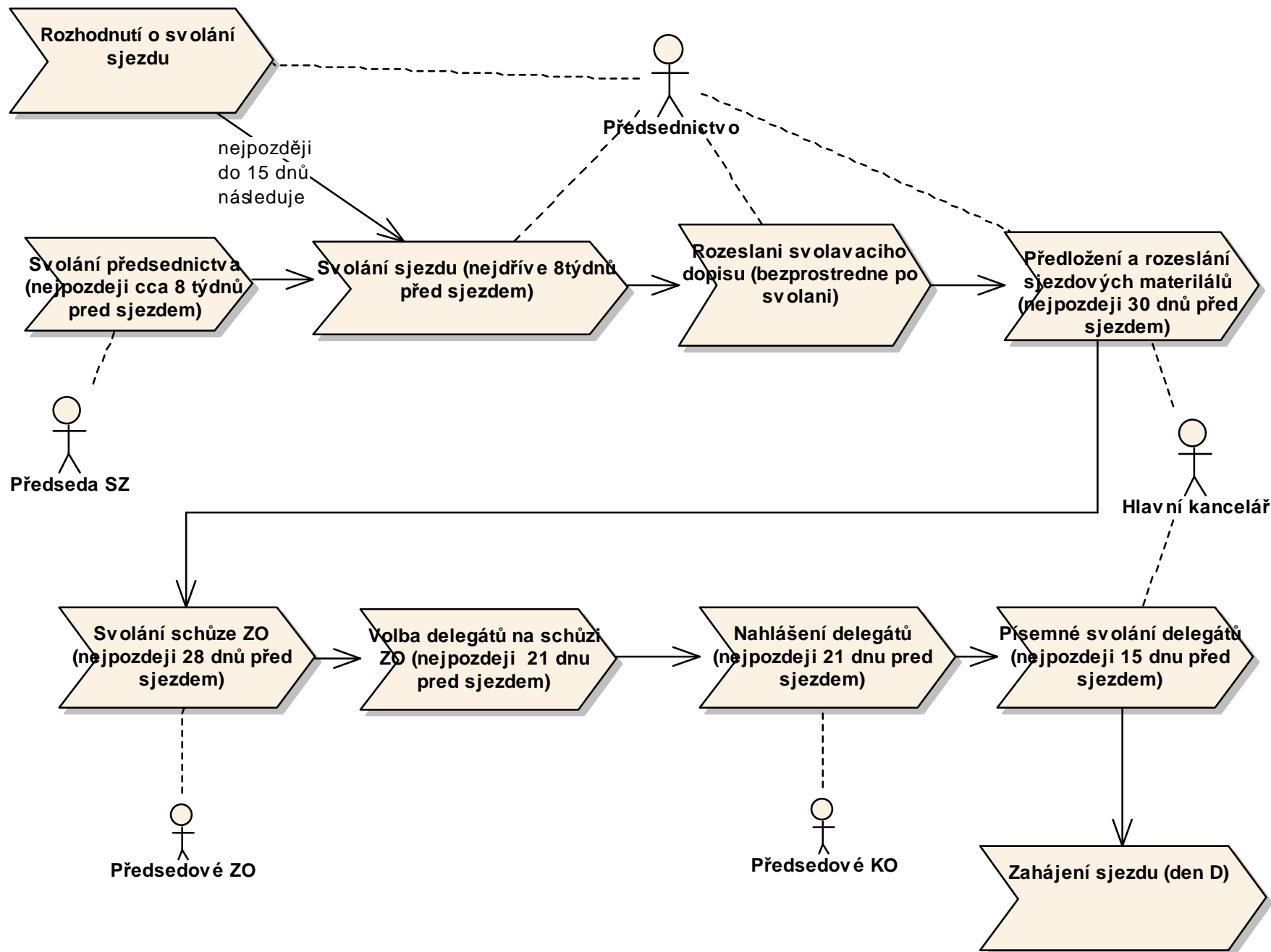
---

Jednoduchý příklad zachycení business procesů na nižší úrovni abstrakce.

Svolání a příprava sjezdu politické strany.

(Notace: Eriksson-Penker Business Extensions)

analysis Příprava sjezdu - procesní model





# UML – digramy aktivit

---

Viz. Arlow.

- a) První část přednášek, str. 203-214
- b) Kniha, str. ??



# UML – digramy aktivit (procesů)

---

Ukázka modelování v EA:

Objednávka zboží ve velkém podniku.

Subproces.

Swimmlanes.

Začátek.

Konec.

Větvení rozhodováním.

Paralelismus



# Business Domain Model

---

Model podnikových/obchodních entit.

Příklad – turistický oddíl.

Rozdíl mezi byznys a analytickým modelem entit.





# Organizace práce

---



# Prokrastinace

---

- notorické odkládání práce na později
  - vysoké zastoupení prokrastiniků mezi VŠ studenty (v němčině *Studentensyndrom*)
  - Postižený vyvíjí nějakou činnost, není to ale ta, která směřuje k cíli.
- 
- [procrastinus.com](http://procrastinus.com)
  - GOOGLE keyword: prokrastinace



# Prokrastinace - Jak se bránit?

---

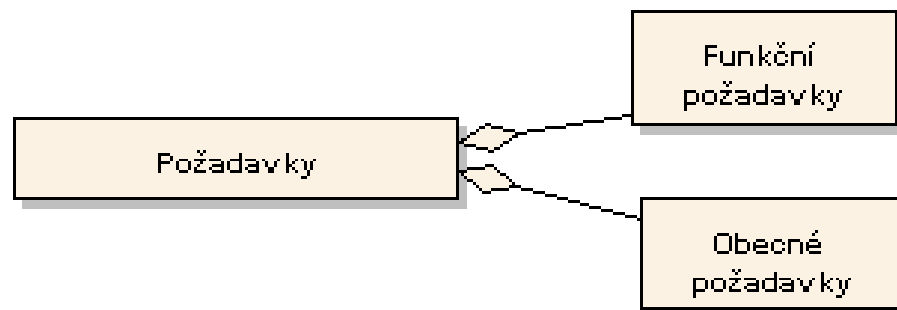
- **Pracovat metodou timeboxingu.** Stanovit si úkol na následující časový úsek (45 až 60 minut) a tomu se věnovat s plným soustředěním. Pak se odměnit pauzou. Je možné naplánovat si práci na nepříjemném úkolu na pouhých pět minut. Tím oblafneme naši vlastní vůli v tom základním – začít. Je celkem pravděpodobné, že po pěti minutách se do práce zabereme natolik, že jí věnujeme více času.
- **Nenechat se rušit.** Pokud pracuji na nějakém úkolu, nepřijímat e-maily, nemít zapnuté komunikátory, ztišit mobil, nejlépe ani nehledat na internetu. Komunikaci se věnovat jen jednou za hodinu nebo za dvě.
- **Prokrastinovat strukturovaně.** Když se nedokážu vyhnout otálení, je dobré mít seznam aktuálních úkolů ty v rámci prokrastinace plnit.
- **Realisticky plánovat.** Představit si pokud možno krok za krokem, co pro daný úkol musím udělat a jak dlouho to trvá. Pak udělat odhad přidat nejméně deset procent jako rezervu.



# Požadavky a jejich specifikace

---

- Zachycují očekávání zákazníka
- mají jasné identifikátory (číslují se)
- hierarchicky děleny (zde na "funkční" a "obecné", ale možno i jinak)



- jeden požadavek=pouze jedno měřitelné "očekávání",
- obsahují (pokud možno) minimální množství informací o implementaci



# Funkční požadavky

---

- definují co bude systém umožňovat
- měly by mít stanoveny priority

Př.:

- 1.1 Systém bude evidovat(minimálně po dobu pěti let) kdo a kdy změnil emailovou adresu na kterou je uživatelům zasíláno heslo.
- 1.2. Systém bude vždy při změně emailové adresy, na kterou je uživatelům zasíláno heslo, o této změně uživatele informovat a to tak, že mu na původní i novou emailovou adresu pošle zprávu o tom, kdo a kdy změnil jeho adresu na zasílání hesla.



## Obecné požadavky

---

- vztahují se k celému systému
- spíše omezují způsob, jak bude systém navržen
- většinou se realizují vhodnou volbou jádra systému



# Typové příklady obecných požadavků

---

- požadavky na parametry RAMS (reliability, availability, maintainability, safety/security)
- požadavky na výkon
- požadavky na použitou platformu (operační systém, hardware, ...)
- požadavky na rozhraní s uživateli i jinými systémy (vícejazyčnost, WEB/WAP/SMS brány/... , ....)
- požadavky související s právními aspekty (výhradní / nevýhradní / otevřená licence, otevřený kód, standardy atd.)



# Příklad obecného požadavku

---

## **6.5. Systém bude spolehlivý.**

Takovýto požadavek může být v závěru projektu oběma stranami interpretován velmi rozdílně. Proto je nutné požadavek upřesnit třeba takto:

## **6.5. Systém bude spolehlivý.**

6.5.1. Střední doba do výpadku systému bude maximálně 20 dní.

6.5.2. Střední doba do opravy systému bude maximálně 12 hodin.





# Dotazy ?

---