

# Princípy tvorby softvéru

Robert Lukočka

lukotka@dcs.fmph.uniba.sk  
www.dcs.fmph.uniba.sk/~lukotka  
M-255

# Akými fázami prechádza kód

- 1 Requirements
- 2 Design
- 3 Implementation
- 4 Verification
- 5 (Deployment)
- 6 Maintenance

# Základné modely vývoja SW

## Vodopádový model

- 1 Priama aplikácia „inžinierskych“ postupov.
- 2 Neintuitívny.

## Iteratívno-inkrementálny

- 1 Široký pojem, najčastejšie sa vyobrazuje ako opakujúci sa vodopád
- 2 Často však niekoľko vodopádov prebieha paralelne.
- 3 Každý kus kódu si prejde svojím vodopádom.

# Porovnanie

## Vodopádový model

- 1 Má problém s meniacimi sa a nejasnými požiadavkami, preto je často nerealistický.
- 2 Chyby objavime až ku koncu (a chyby sú často v požiadavkách a teda sa musíme vrátiť na začiatok).
- 3 To, či je projekt úspešný sa dozvedáme neskoro.
- 4 Odpor k zmenám môže mať vplyv na kvalitu výsledku.

# Porovnanie

## Vodopádový model

- 1 Skôr vieme odhadnúť celkový rozsah, cenu a časový rámec projektu
- 2 Viac menej zaručuje existenciu vhodnej dokumentácie, čo je výhodou ak sa obmieňajú členovia tímu.
  - Toto uľahčuje určité „malé zmeny”.
- 3 Kladie menšie nároky na flexibilitnosť dizajnu.
  - Väčšie zmeny sú ťažšie.
  - Code rot.
- 4 Nasadenie je možné urobiť raz.

# Kedy použiť vodopád

- 1 Stabilné a dobre definované požiadavky.
- 2 Problematické testovanie a nasadzovanie
- 3 Extrémne vysoké nároky na bezchybnosť.
- 4 Je potrebné rýchlo poznať cenu a časový rozsah.
- 5 **Kontrakt to defacto vyžaduje.**

# Iterácie

- 1 Niekedy môže byť ťažké rozbiť úlohu na menšie časti.
- 2 Rozdeliť úlohu je možné vždy, otázkou je efektivita.
- 3 Nechať väčši celok alebo ho ešte rozbiť?

# Agile

- 1 Sada princípov pre vývoj softveru.
- 2 Iteratívny prístup.
- 3 Úzka kooperácia so zákazníkom a stály feedback
- 4 ...



# Explaining Agile part 1

Explaining Agile - Martin Fowler and Neil Ford

# Softvérové kontrakty

Ako vyzerajú softvérové kontrakty?

- 1 Kontrakty sú o rozdelení rizika
- 2 Štandardné kontrakty sú skôr vhodné na vodopádový vývoj.
- 3 Stanoviť presne rozsah kontraktu je veľmi ťažké.
- 4 Požiadaviek je veľa, čiastočne to spôsobuje aj absencia noriem.
- 5 Ideálne je keď podrobný kontrakt nie je potrebný.
- 6 Pevná cena chráni zákazníka, naozaj?
- 7 „Perhaps the reason these type of contracts (štandardné) survive is because they can be defended in court.”

# Softvérové kontrakty

Kontrakty sú o rozdelení rizika.

- 1 Zmluva presúva riziko na dodávateľa.
- 2 Výnimkou sú zmeny.
- 3 Zmeny: change boards, navyšovanie ceny.
- 4 Cez zmeny si často skúsený kontraktor znásobí výslednú cenu.
- 5 Kontraktor ktorý robí zmeny draho môže ponúknuť nižšiu cenu.

V podstate je všetko ok, zadávateľ je zodpovedný za požiadavky a nesie zodpovednosť za to, že sú nesprávne. Zhotoviteľ zodpovedá, že vykoná plan.

# Agile contracts

## Kľúčové aspekty

- 1 Požiadavky sú len hrubé, možno iba projektová vízia a ciele a spôsob merania naplnenia cieľov.
- 2 Zdieľanie rizika (obe strany majú mať záujem o úspešný výsledok projektu)
  - 1 neúspech, predraženie, pomalý vývoj, čiastočná funkcionality
- 3 Možnosť ukončiť kontrakt vo viacerých bodoch (fail fast = menej nákladov na advokátov)
- 4 Kontrakt definuje procesy a nie detajlné požiadavky.
- 5 Zadávatel' kontraktu musí úzko spolupracovať so zhotoviteľom.

# Agile contracts

Znižovanie rizika kontraktora.

- Minimum viable product
- Fixed cost
- Možnosť bezplatného ukončenia projektu

# Agile contracts

Ako teda na to keď chceme pracovať iteratívne?

- 1 Skryť, že robíme inkrementálne (zadávateľ kontraktu dokumentácii aj tak väčšinou nerozumie).
- 2 Platba (a produkt) iba v prípade spokojnosti
- 3 Pokračujúce kontrakty
- 4 Money for Nothing, Change for Free
- 5 Capped time / materials
- 6 Kombinácie

# Ako tenderovať?

- Možno postupovať podobne ako pri klasických tendroch, akurát „tenderovať“ iné parametre.
- Je dobré, keď sú projekty malé.
- Odhad ceny je možné urobiť výberom reprezentatívnych požiadaviek.