

# Princípy tvorby softvéru

## Proces tvorby softvéru

Robert Lukotka

lukotka@dcs.fmph.uniba.sk

www.dcs.fmph.uniba.sk/~lukotka

M-255

# Akými aktivitami prejde každý “kus” softvéru?

- Business Analýza
- Požiadavky
- Dizajn
- Implementácia (Construction)
- Verifikácia a validácia
- Nasadenie
- Údržba

# Základné modely vývoja SW

- Linear / predictive model (Vodopádový model) - Každá fáza začne až po ukončení predchádzajúcej
- Iterative - agile / adaptive modeô - Vývoj prebieha v cykloch
  - viacero cyklov môže bežať súčasne

Dĺžka cyklov môže byť veľmi rôzna.

# Porovnanie

## Lineárny model:

- Má problém s meniacimi sa a nejasnými požiadavkami.
- Málo feedbacku.
  - Chyby objavime až ku koncu.
  - To, či je projekt úspešný sa dozvedáme neskoro.
- Odpor k zmenám môže mať vplyv na kvalitu produktu.
- + Skôr vieme odhadnúť celkový rozsah, cenu a časový rámec projektu.
- + Viac-menej zaručuje existenciu vhodnej dokumentácie.
- + Kladie menšie nároky na flexibilitu dizajnu.
- + Produkt možno nasadiť naraz.

# Porovnanie

## Cyklický

- + Umožňuje ľahšie zakomponovať zmeny požiadaviek.
  - Nejasné požiadavky a zmeny požiadaviek sú typickou črtou v SW projektoch → I-I prístup
- + Umožňuje získať feedback rýchlejšie
  - Aby sa naplno prejavili benefity I-I vývoja, je potrebná častá spolupráca so zadávateľom projektu
- Vyššie nároky na kvalitu (flexibilitnosť designu, kód). k vývoju.
- Ťažké je predpovedať celkový rozsah a cenu projektu.
  - Deadliny a fixné ceny sú častou požiadavkou zadávateľa projektu.

# Kedy zvážiť použiť lineárneho modelu?

- Stabilné a dobre definované požiadavky, v princípe známe riešenie.
- Problematické testovanie a nasadzovanie
- Extrémne vysoké nároky na bezchybnosť.
- Je potrebné rýchlo poznať cenu a časový rozsah.

# Feedback pri vývoji softvéru. Čo sa môže pokaziť, kedy sa to pokazí a kedy to zistíme?

## Príklad

Softvérový projekt nemá šancu naplniť business ciele zákazníka.

- Business Analýza
- Požiadavky
- Dizajn
- Implementácia (Construction)
- Verifikácia a validácia
- Nasadenie
- Údržba

Čo s tým? Feasibility study; lean startup.

# Feedback pri vývoji softvéru. Čo sa môže pokaziť, kedy sa to pokazí a kedy to zistíme?

## Príklad

Chyba v implementácii.

- Business Analýza
- Požiadavky
- Dizajn
- Implementácia (Construction)
- Verifikácia a validácia
- Nasadenie
- Údržba

Čo s tým? editor pozná programovací jazyk, pair programming, ľahko čitateľný kód, code review, silné typovanie, unit testy, test driven development, continuous integration.



# Feedback pri vývoji softvéru. Čo sa môže pokaziť, kedy sa to pokazí a kedy to zistíme?

## Príklad

Design dvoch rôznych častí systému robený rôznymi ľuďmi je nekompatibilný

- Business Analýza
- Požiadavky
- Dizajn
- Implementácia (Construction)
- Verifikácia a validácia
- Nasadenie
- Údržba

Čo s tým? rýchla iterácia, configuration management, zdokumentovaná architektúra, častá integrácia, continous integration.

# Feedback pri vývoji softvéru. Čo sa môže pokaziť, kedy sa to pokazí a kedy to zistíme?

## Príklad

Design nedokáže splniť všetky požiadavky.

- Business Analýza
- Požiadavky
- Dizajn
- Implementácia (Construction)
- Verifikácia a validácia
- Nasadenie
- Údržba

Čo s tým? Testovanie designu, správne poradie pridávania funkcionality do designu, písanie testov ako súčasť dizajnu, rýchle iterácie.

# Feedback pri vývoji softvéru. Čo sa môže pokaziť, kedy sa to pokazí a kedy to zistíme?

## Príklad

Nesprávne pochopené požiadavky / neúplné požiadavky.

- Business Analýza
- Požiadavky
- Dizajn
- Implementácia (Construction)
- Verifikácia a validácia
- Nasadenie
- Údržba

Čo s tým? Analýza, verifikácia, a validácia požiadaviek; častý kontakt so zákazníkom, časté nasadzovanie softvéru.

# Feedback pri vývoji softvéru. Čo sa môže pokaziť, kedy sa to pokazí a kedy to zistíme?

## Príklad

Implementácia projektu podľa požiadaviek nemá šancu naplniť business ciele zákazníka.

- Business Analýza
- Požiadavky
- Dizajn
- Implementácia (Construction)
- Verifikácia a validácia
- Nasadenie
- Údržba

Čo s tým? Časté nasadzovanie softvéru; meranie výsledkov projektu; lean startup.

# Proces vývoja softvéru

- Nie je praktické mať pre každý projekt samostatný proces vývoja softvéru.
- Typických je niekoľko procesov, a časť workflowu v rámci týchto procesov je zdieľaná.
- Zlepšovanie procesov by mala byť integrálna súčasť procesov.

# Softvérové kontrakty - Diskusia

Fakulta si chce dať urobiť SW systém. Ako má vybrať zhotoviteľa?  
**Ako má vyzerat' zmluva?**

# Softvérové kontrakty

- Kontrakty sú o rozdelení rizika.
- Fixed price, time, and scope sú skôr vhodné na lineárny vývoj.
- Stanoviť presne rozsah kontraktu je veľmi ťažké.
- Požiadaviek je veľa, čiastočne to spôsobuje aj absencia noriem.
- Pevná cena nie je nutne v záujme objednávateľa, dodatky môžu byť drahé.
- „Perhaps the reason these type of contracts (Fixed price, time, and scope) survive is because they can be defended in court.”  
- Allan Kelly

# Fixed price, time, and scope kontrakty

- Zmluva presúva riziko súvisiace s vývojom softvéru na zhotoviteľa.
- Riziko zmeny požiadaviek je na zadávateľovi.
- Zmeny: change boards, navyšovanie ceny.
- Cez zmeny si kontraktor môže výrazne zvýšiť výslednú cenu.
- 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 kontrakty

## Kľúčové aspekty

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

# Agile kontrakty

## Princípy kontraktov

- Platba za človekohodinu
- Pokračujúce kontrakty
- Platba (a produkt) iba v prípade spokojnosti
- Capped time / materials
- Money for Nothing, Change for Free
- Kontrakt na základe business požiadaviek

# Diskusia

Fakulta si chce dať urobiť SW systém. Ako má vybrať zhotoviteľa?

# Ako tenderovať?

- Odhad ceny je možné urobiť výberom reprezentatívnych požiadaviek.
- Je dobré, keď sú projekty malé.
- Môže byť zaujímavé „tendrovať“ iné parametre.

# Diskusia

Čo ak nevytvárame produkt pre konkrétneho zákazníka?

- Už máme zákazníkov.
- Nevieme kto je náš zákazník.

# Čo ak nie je zdroj požiadaviek?

- Naše domienky o tom, kto môže byť náš zákazník, je potrebné verifikovať.
- Naše domienky o tom, čo zákazník chce treba, je potrebné verifikovať.
- LEAN Startup.

# Zdroje

- Wikipédia: Software development process
- <https://developex.com/blog/10-contracts-for-your-next-agile-software-project/> Peter Stevens: 10 Contracts for your next Agile Software Project
- SWEBOK V3 - Chapters 7 and 8