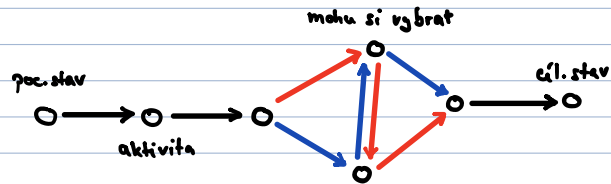


1, Proces vývoje softwaru



- software:
 - výroba? (pevně dané aktivity) = řemeslo
 - vývoj? (pevně daný proces)
 - tvorba? (volné aktivity a proces) = umění

- proces: po částech uspořádaná posloupnost aktivit vedoucí od počátečního do cílového stavu



- Capability Maturity Model (CMM)

- jednoduchý model ohodnocení kvality (vyspělosti) procesu vývoje SW

level 1: initial = intuitivní ad-hoc vývoj

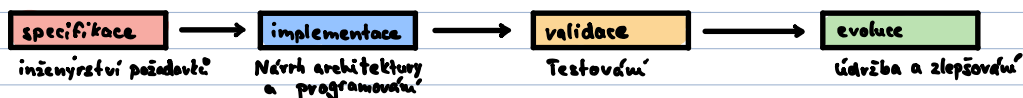
level 2: repeatable = použití osvědčených postupů

level 3: defined = proces je zcela popsán se všemi eventualitami

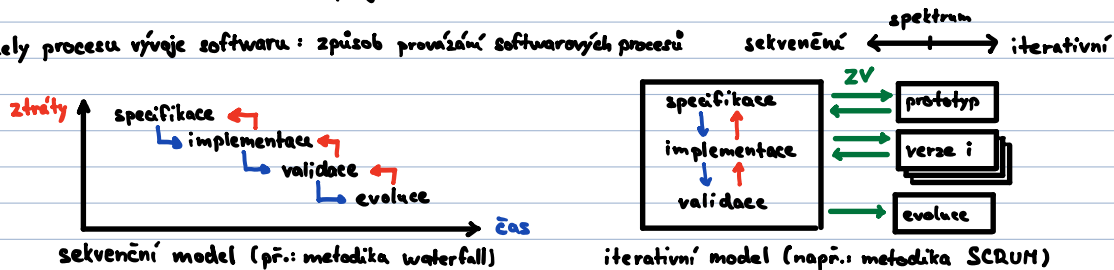
level 4: measured = proces je měřen a vyhodnocován

level 5: optimizing = proces je zpětnově zkontrolován od odchylek

- softwarové procesy: soustava vysokoúrovňových úzce spolu svázaných aktivit používaných při vývoji SW



- Modely procesu vývoje softwaru: způsob provádění softwarových procesů



- Metodika vývoje softwaru

tradiční ←→ agilní

= zhmotnění (konkrétní způsob) realizace modelu procesu vývoje SW (role, priority, parametry, filosofie)

- tradiční metodiky:

- + snadné řízení
- + pevně dané role a plán
- + vhodné pro velké projekty
- velká cena za chybu
- zdlouhavá testovací fáze
- velké Δt od startu do konce

př.: waterfall, RUP, Böhmův spirálový model

- agilní metodiky

- + snadné řízení změn
- + testování během implementace
- + snadné zapojení klienta
- neřídí se snadno (jen pro malé týmy)
- potřeba kvalitního spolehlivého týmu existujícího podniku
- špatně se zavádí v hierarchickém prostředí

př.: SCRUM, XP, TDD, FDD, BDD, Lean (z ní vychází DevOps)

- agilní manifest: prohlášení předních světových vývojářů definující filosofii agilních metodik

jednotlivci a interakce	před	procesy a nástroje
fungující software	před	vyčerpávající dokumentace
spolupráce se zákazníkem	před	vyjednávání o smlouvě
reagování na změny	před	dodržování plánu

• agilní metody

a, Vývoj řízený testy (TDD) = nejdřív test, pak kód

1, assert sum(4,5) == 9

test je základ SW

2, def sum(a,b): return a + b

metoda je jen nástroj pro splnění testu

b, Extrémní programování (XP) = co funguje dělejme do extrému

- délka iterací je extrémně krátká (dny, den, hodiny)
- specifikace se mění i několikrát za den (zákazník je součástí týmu)
- párové programování
- samodokumentující kód (týmové standardy pro názvy, atd.)
- společné vlastnictví a zodpovědnost (kdokoliv upravuje cokoliv)
- vývoj je řízen testy

c, Štíhlý vývoj = principy štíhlé výroby převedené do vývoje

- eliminuje vše, co nepřináší zákaznickou hodnotu
- základ filosofie DevOps

• samostatná cvičení

SCv.1: Principy agilního vývoje

- kromě agilního manifestu existuje ještě 12 principů agilního vývoje. Seznamte se s nimi na internetu.

SCv.2: CASE nástroje

- Pro podporu řízení SW procesů existují tzv. CASE (Computer-Aided Software Engineering) nástroje. Nalezněte některé nástroje a zjistěte, k čemu se používají.

SCv.3: Gumová kačenka

- Pokud nemá programátor k sobě partáka jako v XP, pak oblíbená technika pro sdílení myšlenek vztahovým kódem je Rubber duck debugging. Podívejte se na e-shopy po vaší vysněné kačence (nákup není povinný).