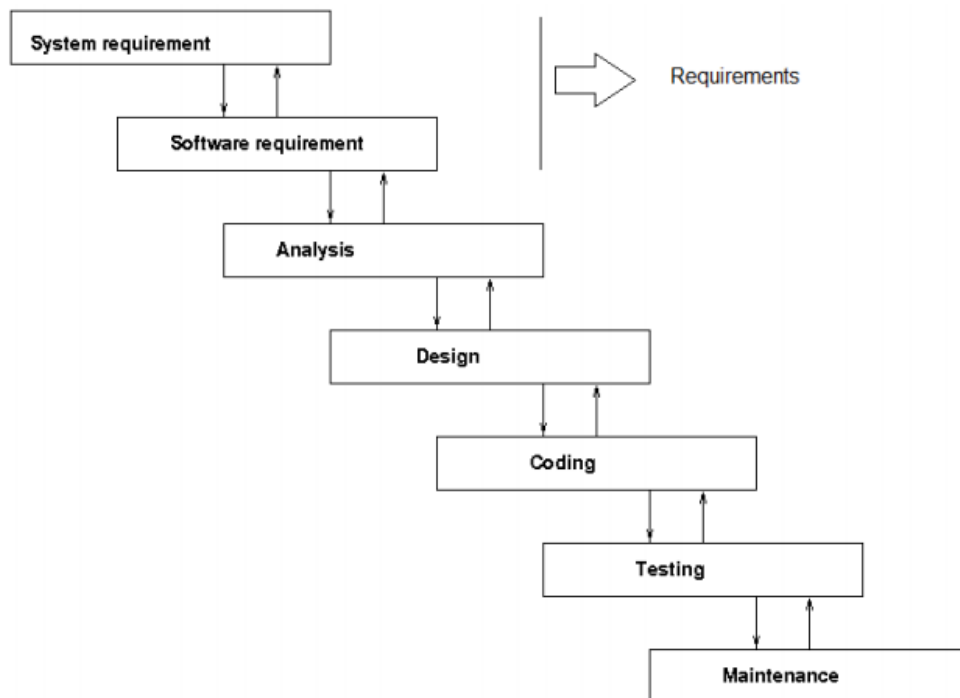


1. Úvod, motivace

- **softwarové inženýrství** = systematický, disciplinovaný a kvalifikovaný přístup k vývoji, tvorbě a údržbě SW; využívá modely a metodiky
- **model** = popis fází životního cyklu SW a vztahy mezi nimi
- **metodika** = určuje postupy každé fáze, vstupy a výstupy
 - o **tradiční**: čas + cena = funkcionality
 - o **agilní**: funkcionality => cena + čas
- finální produkt ovlivňuje zadavatel, právníci, architekti, vývojáři, testéři, bezpečáci, provozáci .. a také cílová infrastruktura, technická omezení, požadavky, cena, konkurence, změny,...
- je nutno plánovat, komunikovat, prioritizovat, přizpůsobit se změnám, řídit směřování,...
- je nutno určit osobu zodpovědnou za rozvoj a směřování produktu

Modely

Vodopád (sekvenční)



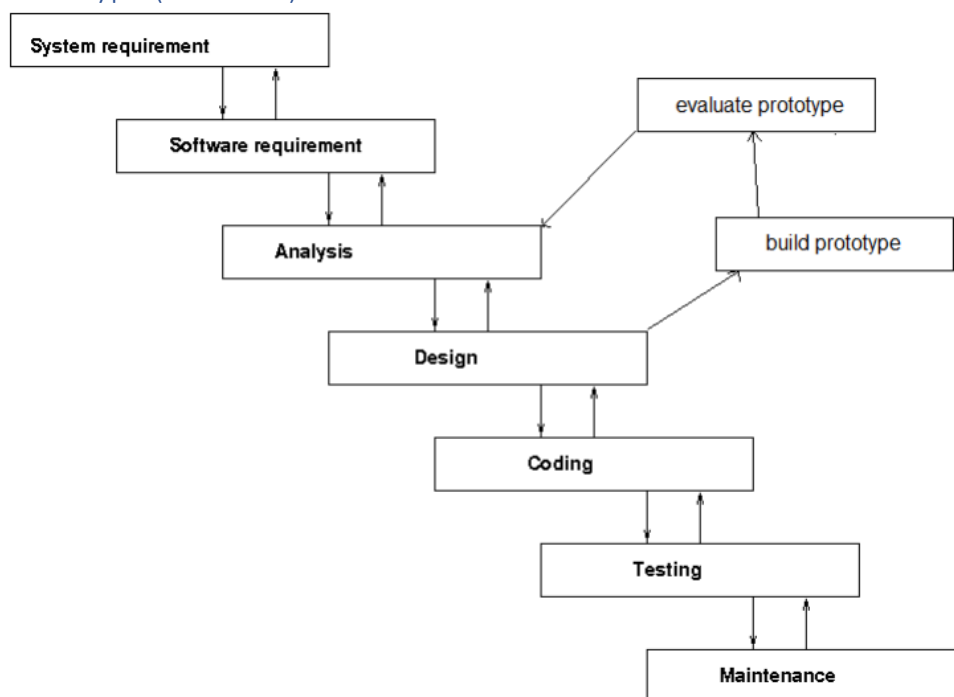
- posloupnost fází, které na sebe navazují
- v základu se postupuje vždy o jeden krok dopředu či dozadu
- **nevýhody**:
 - o velký časový úsek mezi zadáním a prvním releasem
 - o definice všech požadavků již na začátku vývoje
 - o při změně je třeba schválit výstupy všech fází

Spirála (iterativní)



- cyklické opakování jednotlivých kroků vývoje
- je odpovědí na nevýhody vodopádu
- analýza rizik v každém cyklu

Prototype (iterativní)



- prvotní sběr a analýza požadavků
- rychlý návrh
- tvoba prototypu (low/high fidelity) + jeho vyhodnocení zákazníkem
 - o je-li spokojen => posun
 - o jinak zpracování připomínek a posun
- specifikace systému na základě prototypu
- návrh a implementace systému

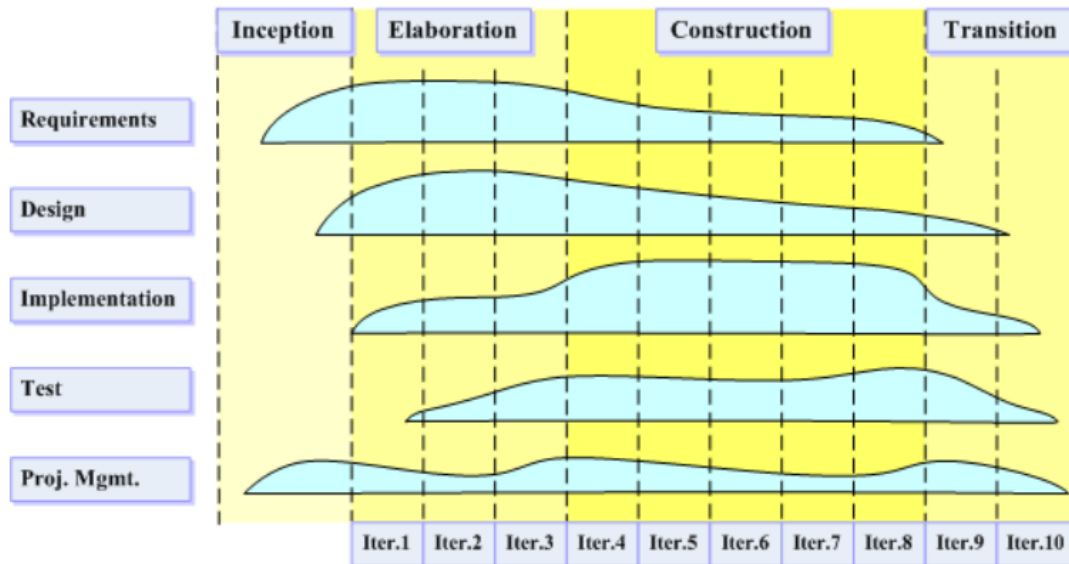
Inkrementální (agilní)

Paralelní (agilní)

Metodiky

- např. Unified Process (UP), Rational Unified Process (RUP), Extrémní programování (XP), SCRUM, Test Driven Development (TDD), Model Driven Development (MDD),...

Unified Process (UP)



- 4 fáze - zahájení, projektování, realizace a předání
- každá fáze obsahuje sběr/revizi požadavků, analýzu, návrh a implementaci

Extrémní programování (XP)

- agilní metodologie vývoje SW
- 4 hodnoty – komunikace, jednoduchost, zpětná vazba a odvaha
- pair programming, testování, návrh modulů, jednoduchost, architektura, integrace, iterace
- 12 praktik:
 - **business praktiky**
 - plánování hry
 - zákazník na pracovišti
 - vydávání malých verzí
 - metafora
 - **týmové praktiky**
 - párové programování
 - společné vlastnictví kódu
 - standardy kódu
 - udržitelné tempo = 40 hodin / týden
 - **programovací praktiky**
 - průběžná integrace
 - jednoduchý návrh
 - refaktORIZACE kódu
 - testování

Test Driven Development (TDD)

- definice funkcionality => napsat test => implementace
- **cyklus:**
 - ○ napsat test
 - spustit testy a ujistit se, že žádný neprojde
 - napsat vlastní kód
 - spustit automatické testy a ujistit se, že dané testy prochází
 - refaktorace

Verzování SW

- číslo verze se většinou skládá ze tří čísel – **X.Y.Z**
 - **X** = hlavní
 - **Y** = vedlejší
 - **Z** = číslo revize
- zvýší-li se nějaké číslo, nižší úroveň se resetuje (1.0.1 => 1.1.0)
- číslo revize se zvyšuje po pár comittech (opravy menších chyb)
- je-li velké množství revizí, přijde doba, kdy je kód třeba přepsat, zpřehlednit a trochu upravit
 - zvýší se vedlejší číslo
- hlavní číslo se zvyšuje při kompletně přepracovaném programu