

# Softverski procesi

Jana Protić 1090/2015

Razvoj softvera 2 ~ Vladimir Filipović

# Pregled ciljeva

---

- Softverski procesi kao koncept i njihovi modeli
- Tri modela i kada se koriste
- Izmene u zahtevima
- RUP

- Šta je softverski proces?
- Sastoji se iz:
  1. specifikacije softvera
  2. dizajn i implementacija softvera
  3. validacija softvera
  4. evolucija softvera
- Takođe sadrže i:
  1. proizvode
  2. uloge
  3. pred- i post- uslovi

- Koji je idealan proces?
- Dve vrste:
  1. planski
  2. agilni
- Standardizacija

# Procesni modeli

---

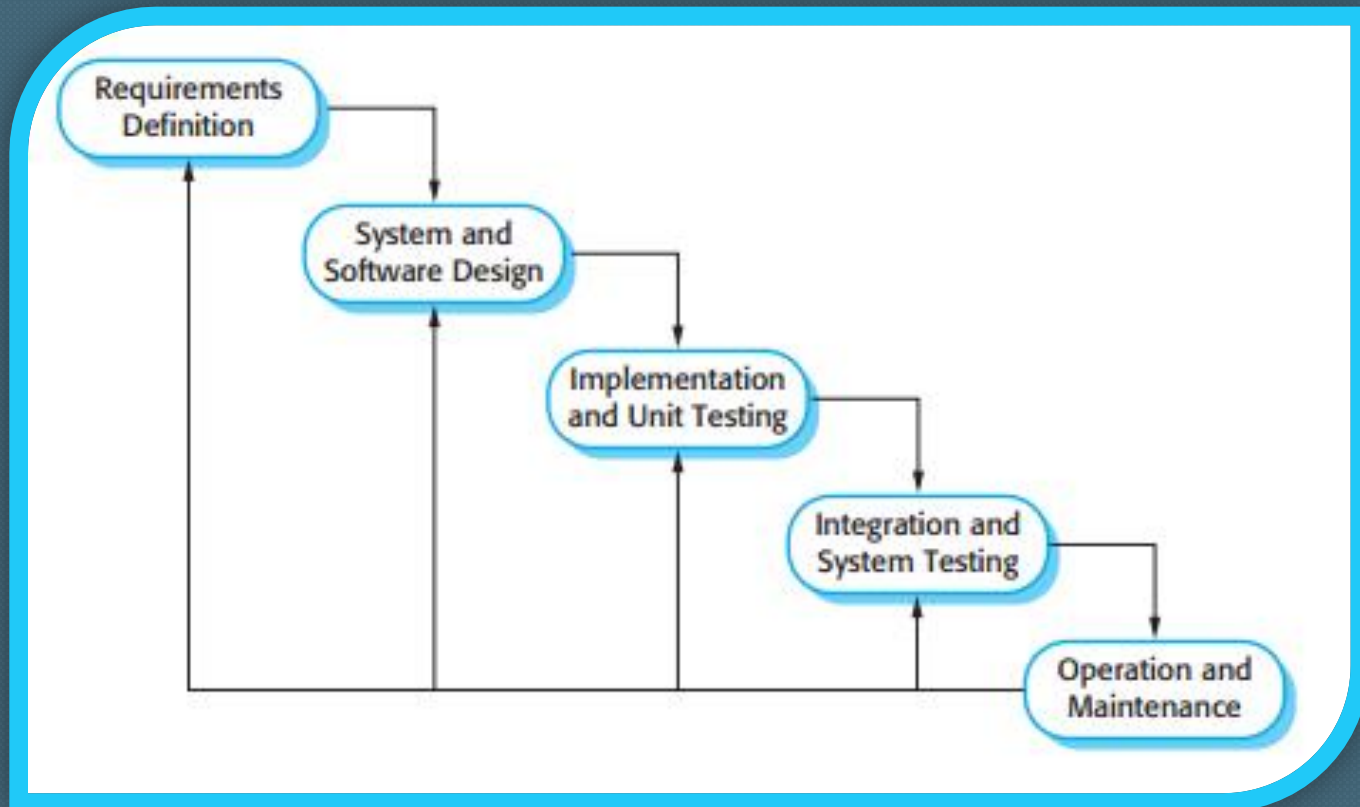
- Šta je procesni model?
- Neki od njih su:
  1. Model vodopada
  2. Inkrementalni razvoj
  3. Model ponovne primene

# Model vodopada

---

- Royce, 1970.
- Vođen planom
- 5 faza:
  1. definicija i analiza zahteva
  2. dizajn sistema i softvera
  3. implementacija i test jedinica koda
  4. integracijski i sistemski testovi
  5. rukovanje i održavanje

# Model vodopada



# Model vodopada

---

- Produkti
- Preklapanja
- Freeze code
- Dokumentacija
- Kada ga koristiti?

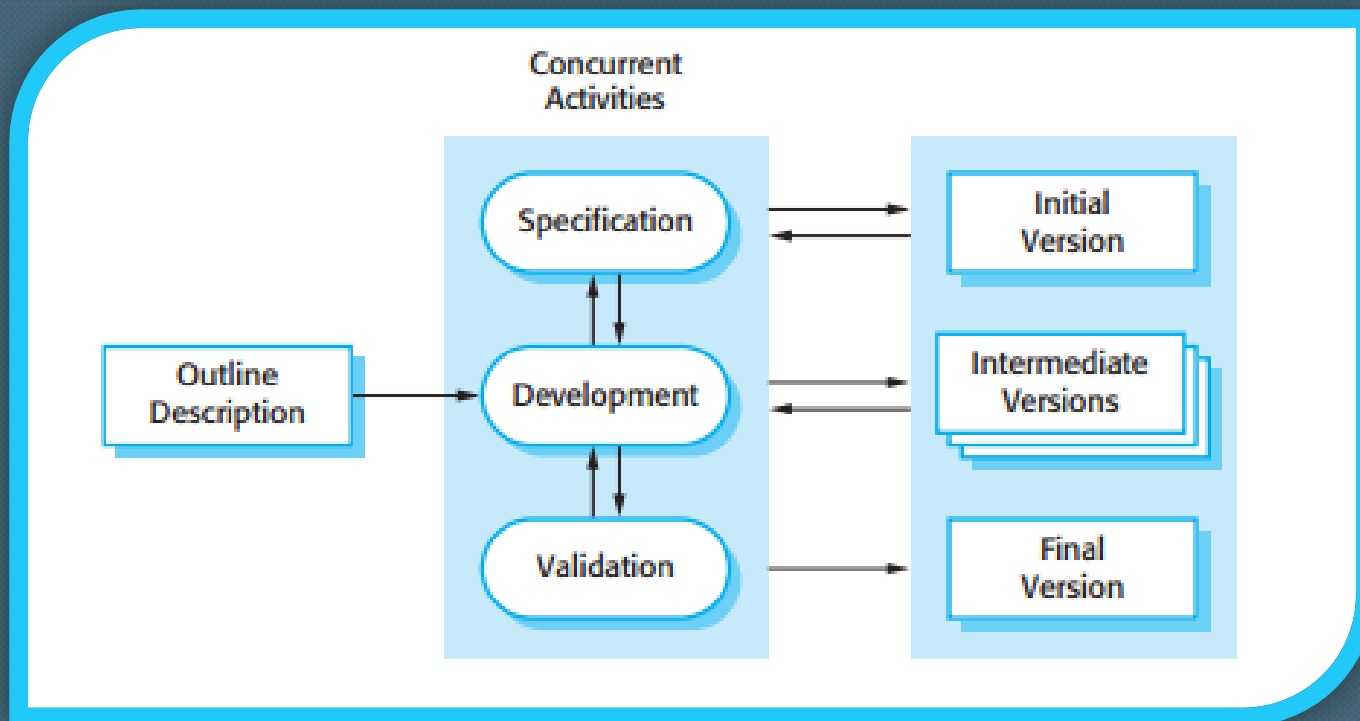


# Inkrementalni razvoj

---

- Ideja
- Agilni pristup
- Isprepletane faze
- Najbliži normalnom pristupu problemu

# Inkrementalni razvoj



# Inkrementalni razvoj

---

- Prvo najbitnije funkcionalnosti
- Prednosti u odnosu na model vodopada:
  1. jeftina promena zahteva
  2. povratne informacije od korisnika
  3. isporuka programa može da bude ranije
- Najčešće korišćen
- Može biti vođen planom, agilan, najčešće kombinacija ova dva

# Inkrementalni razvoj

---

- Problemi:
  1. ne vidi se jasno progres
  2. novim inkrementima se urušava struktura
- Nije pogodan za dugačke projekte gde različiti timovi različite delove održavaju

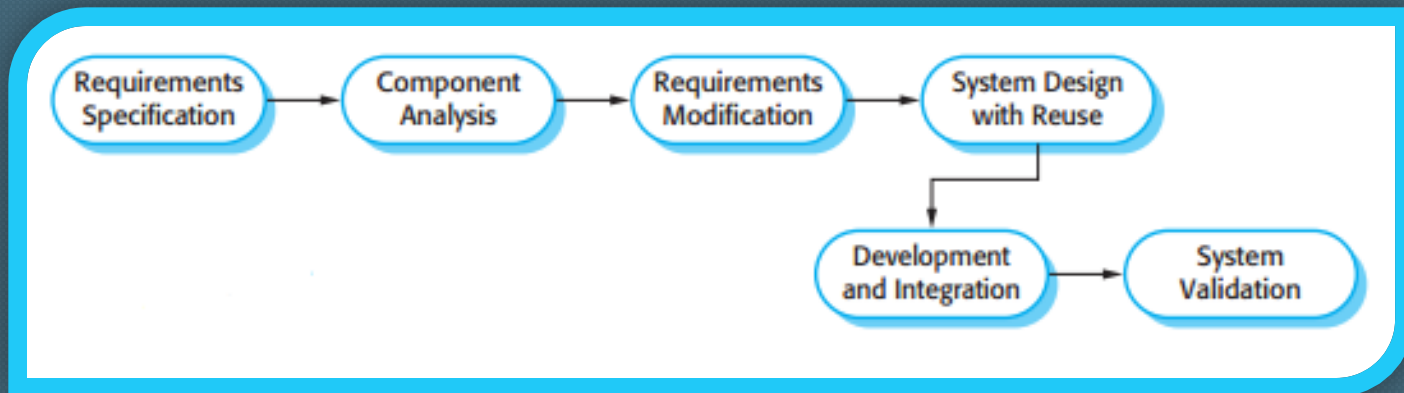
# Model ponovne primene

---

- Javlja se sve češće
- Velika baza ovakvih komponenti kao i podrška framework-ova za pravljenje istih
- Između osnovne dve faze (specifikacije zahteva i validacije sistema) postoje:
  1. analiza komponenti
  2. modifikacija zahteva
  3. dizajn sistema sa ponovnim korišćenjem
  4. razvoj i integracija

# Model ponovne primene

---



# Model ponovne primene

---

- Tri tipa komponenti koje se ovde koriste su:
  1. web servisi
  2. paketi
  3. stand-alone sistemi
- Prednosti i mane

# Aktivnosti procesa

---

1. Specifikacija softvera
  - studija izvodljivosti
  - uvođenje zahteva i njihova analiza
  - specifikacija zahteva
  - validacija zahteva
  
2. Dizajn softvera i implementacija
  - dizajn arhitekture
  - dizajn interfejsa
  - dizajn komponenti
  - dizajn baze



# Aktivnosti procesa

---

## 3. Validacija softvera

- razvojno testiranje
- sistemsko testiranje
- testiranje prihvatljivosti programa

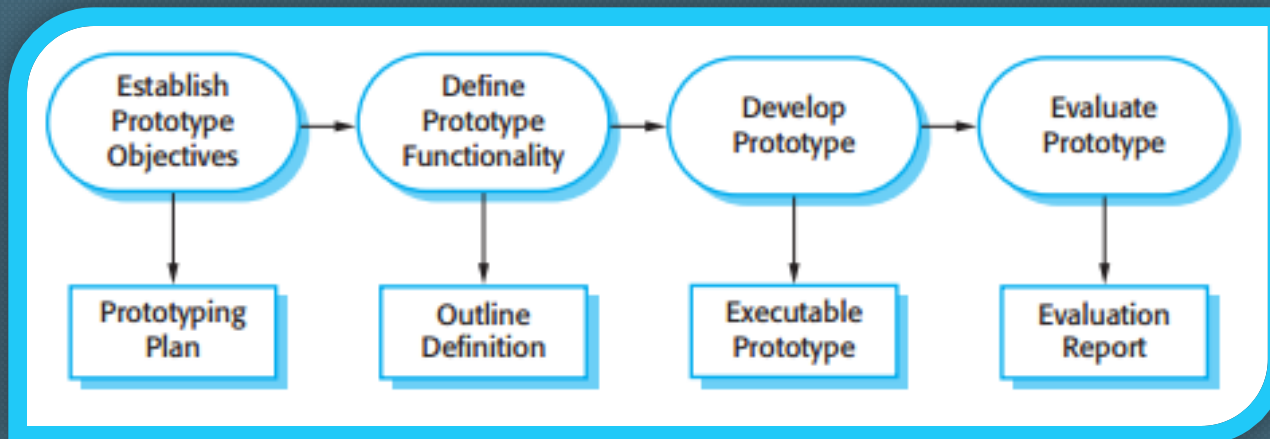
## 4. Evolucija softvera

# Suočavanje sa promenama

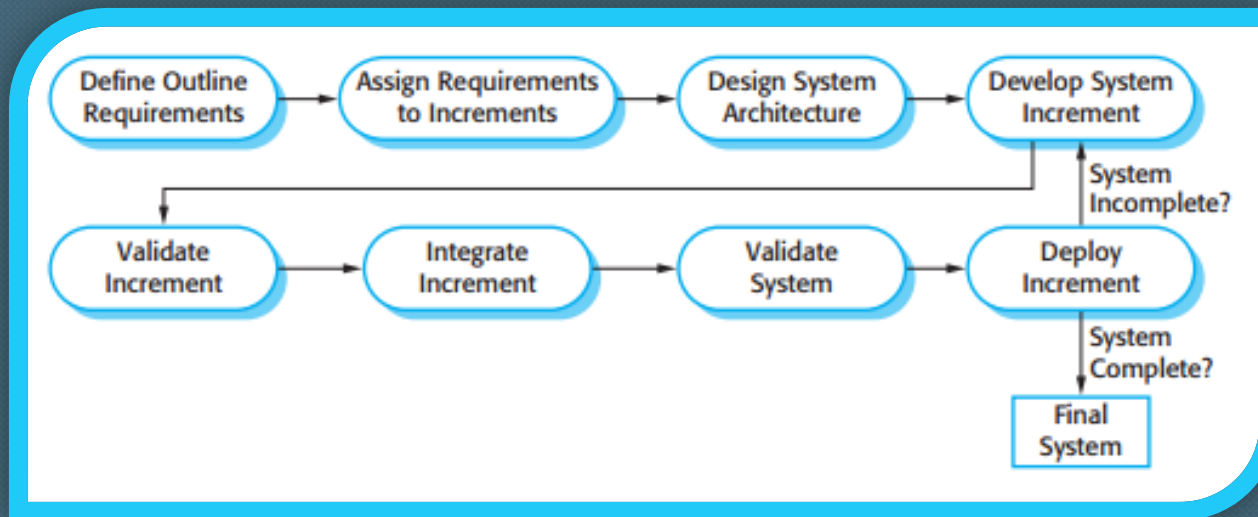
---

- Dva pristupa promenama:
  - Izbegavanje promena
  - Tolerancija na promene
- Par načina suočavanja sa promenama:
  - Pravljenje prototipa
  - Inkrementalno isporučivanje
  - Spiralni model

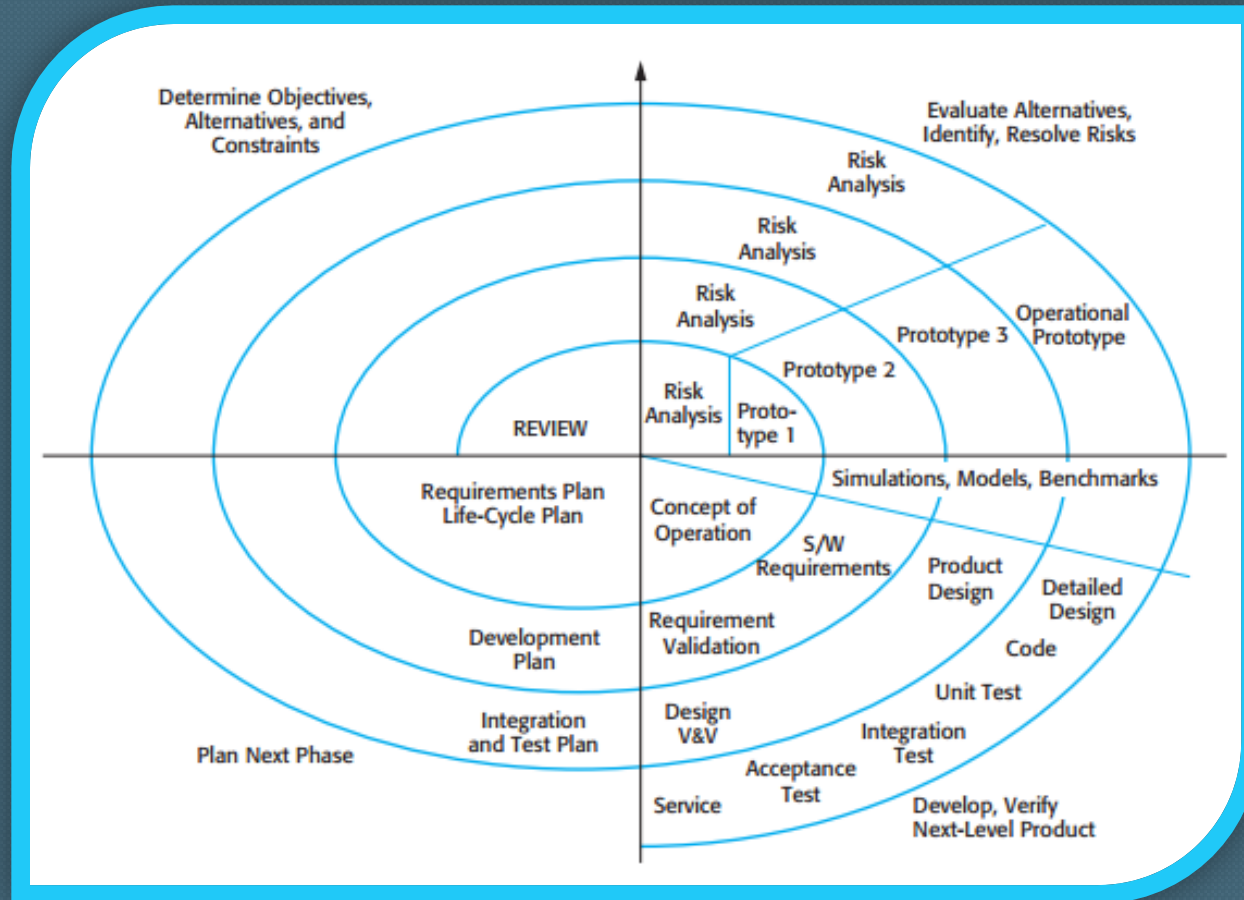
# Pravljenje prototipova



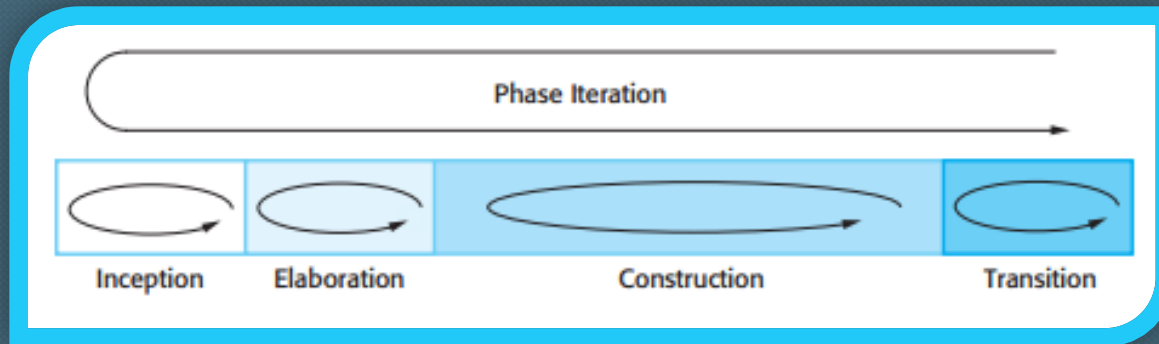
# Inkrementalno isporučivanje



# Spiralni model



# RUP



Bitno:

1. inkrementalan razvoj
2. pažnja na zahtevima
3. koristiti arhitekture koje se sastoje iz komponenti
4. vizuelni model softvera
5. verifikacija kvaliteta softvera
6. kontrola promena

# Zaključak

---

Koji je najbolji model za  
pravljenje softvera?

---

Hvala na pažnji!

