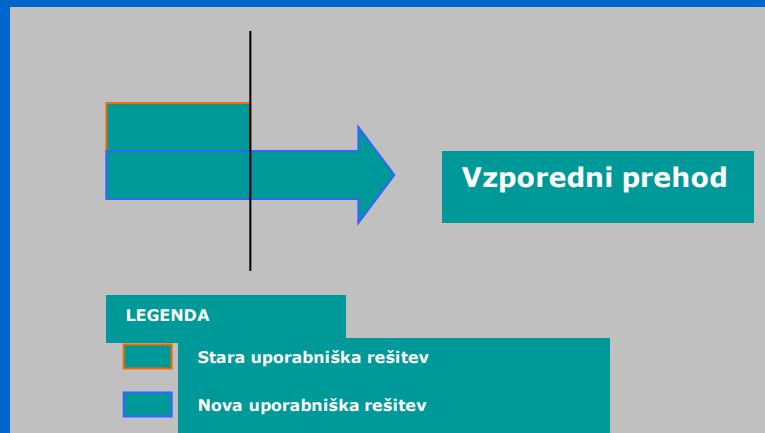


Uvajanje informacijskega sistema

- V praksi poznamo štiri načine prehodov na novo programsko aplikacijo, kjer ima vsak način svoje prednosti in slabosti. Razlike so predvsem v varnosti, stroških in časovnem obsegu prehoda s stare na novo aplikacijo. Načini prehodov so:
 - vzporedno delovanje
 - neposredni prehod
 - postopni ali fazni prehod
 - pilotni način uvajanja

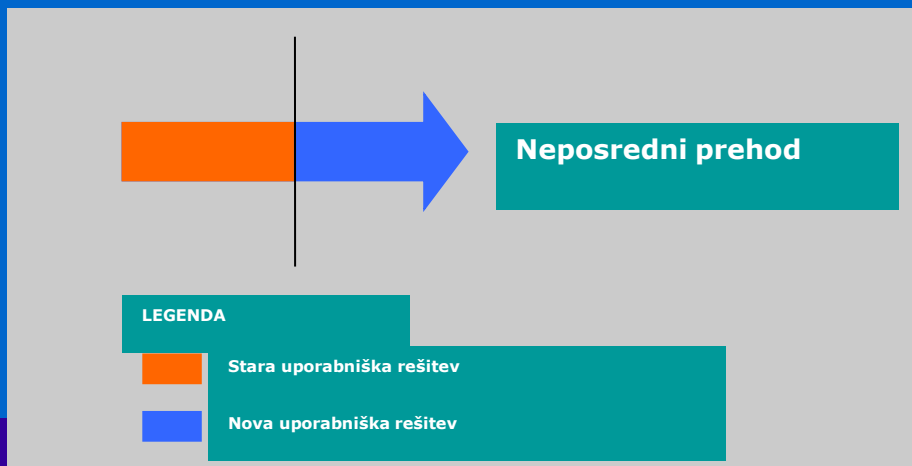
Vzporedno delovanje

- v določenem časovnem obdobju delujeta tako novorazvita kot dosedanja programska aplikacija
- dvojni vnos podatkov
- visoka stopnja varnosti (podjetje ne izgubi podatkov in informacijske podpore)



Neposredni prehod

- v nekem trenutku nova zamenja dosedanjo programska aplikacija
- enkratni vnos podatkov
- nizka stopnja varnosti (podjetje lahko ob napaki izgubi podatkov in izpad informacijske podpore)



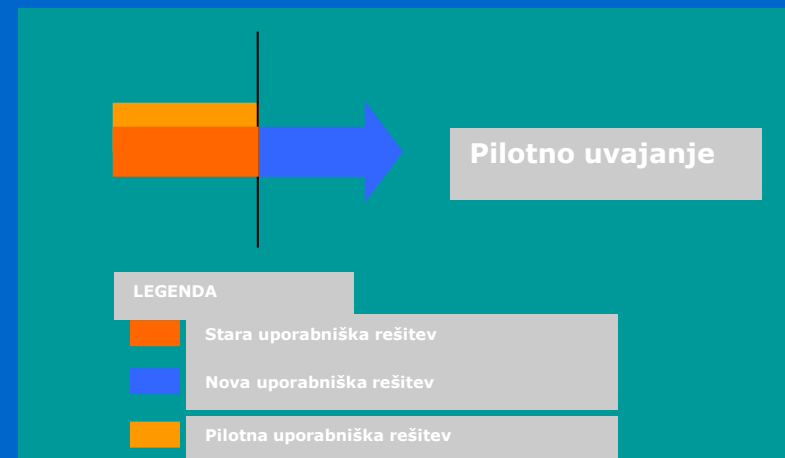
Postopni ali fazni prehod

- vključuje se postopoma po področjih
- enkratni vnos podatkov
- uporabimo le, če sta uporabniški rešitvi dovolj podobni

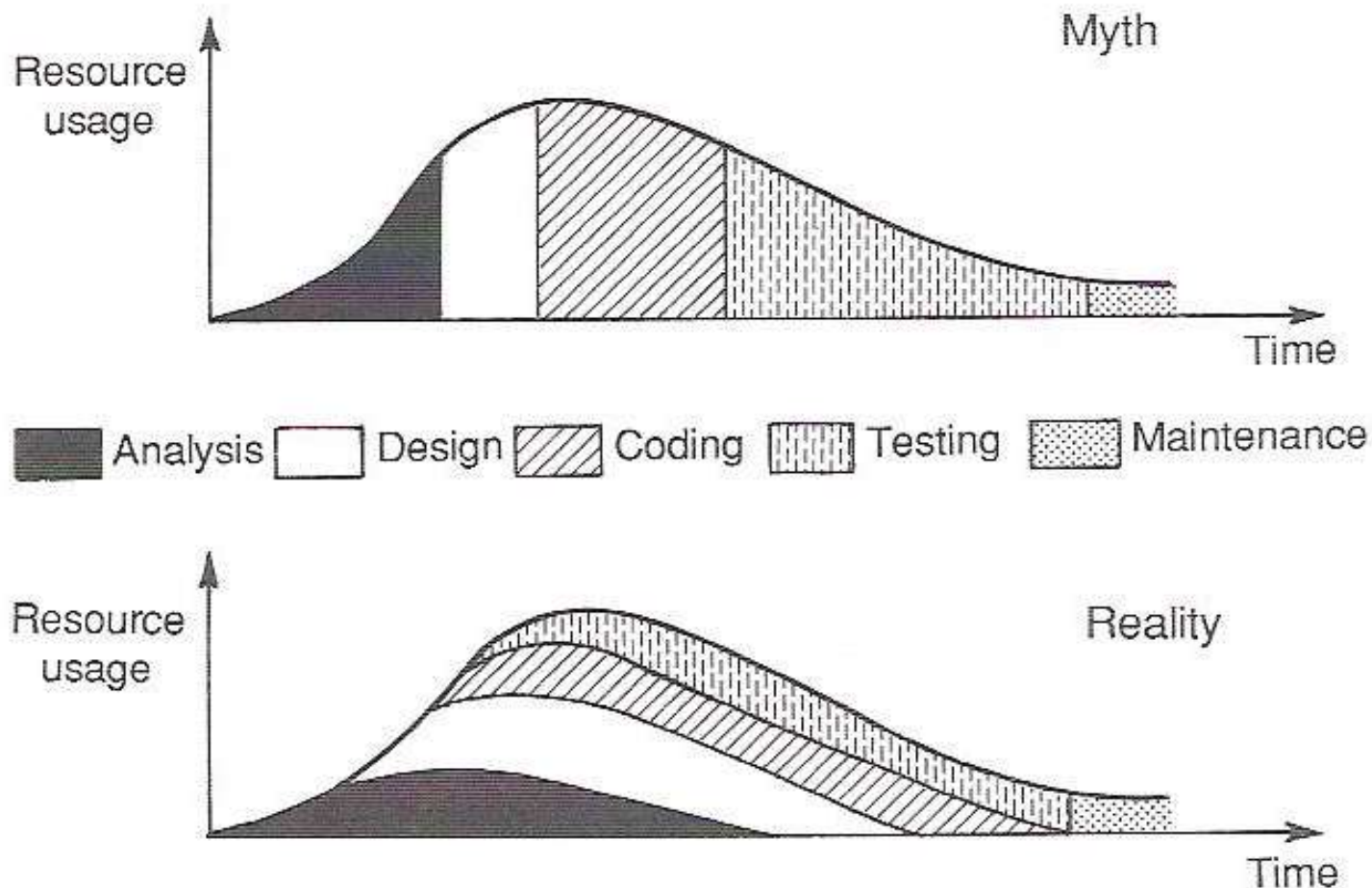


Pilotni način uvajanja

- kombinacija vseh predhodno omenjenih
- izbere se skupina testnih, nad novo uporabniško rešitvijo navdušenih uporabnikov – nekaj časa vzporedno, da se odpravijo napake
- ne silimo vseh
- kako najti prostovoljce za testiranje



•
V idealnem primeru se faze ne prekrivajo:



Dejansko pa se moramo vedno vračati v prejšnje faze

Možni problemi pri metodi življenjskega cikla:

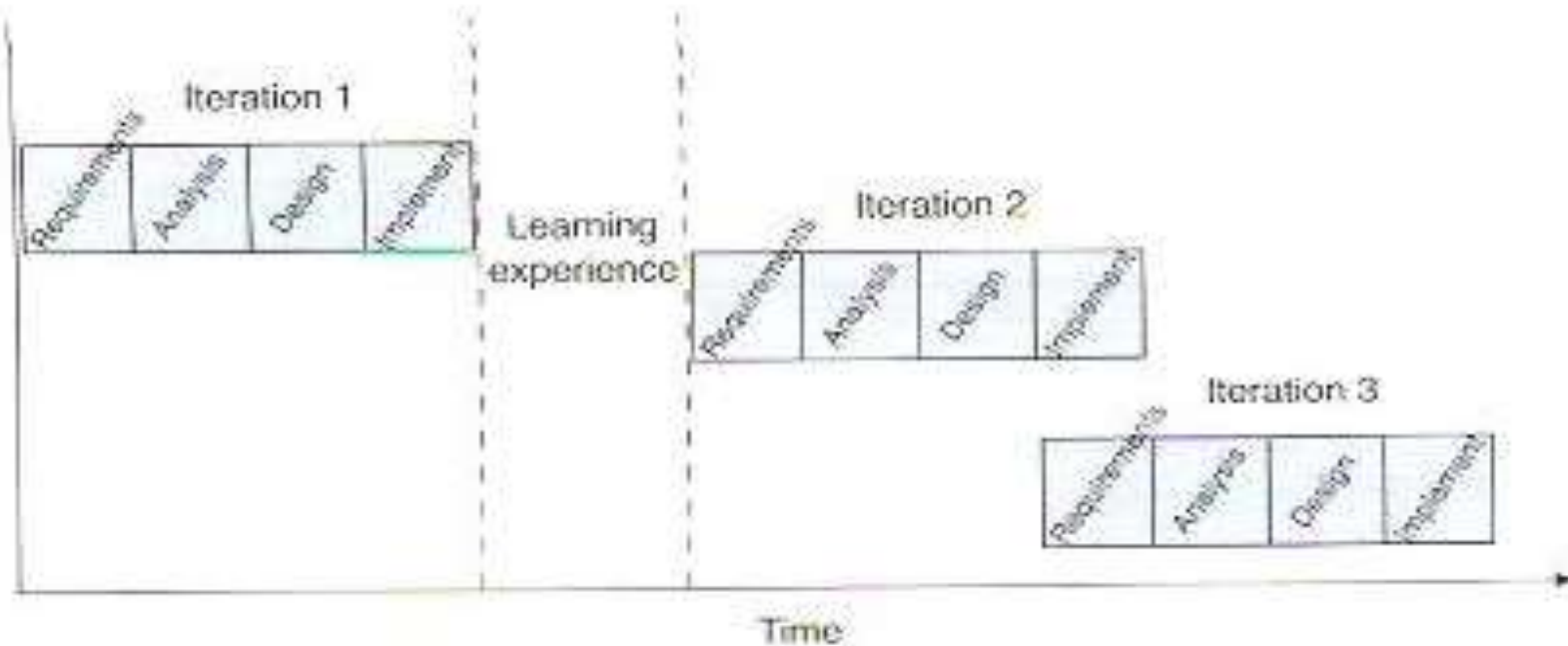
- Ne uspe izpolniti zahtev vodstva (je pa lahko uspešen, kadar razvijamo aplikacijo za podporo operativnih procesov)
- Težko se prilagaja spremembam poslovnih procesov v teku razvoja
- Zahteve so opredeljene predvsem na osnovi zahtevanih vhodov, kar vodi k nefleksibilnosti
- Nezadovoljstvo uporabnikov: dolgo časa preteče od opredelitve zahtev do delujoče aplikacije, uporabniki so premalo vključeni med samim razvojem, nerealna pričakovanja uporabnikov, napačno razumljene zahteve uporabnikov

- Nerealne ocene potrebnih resursov: ljudi, časa
- Nepopolne rešitev: pozablja se na neobičajne, izjemne primere v poslovnih postopkih
- Zaostanki pri razvoju: uporabniki predolgo čakajo na zahtevano aplikacijo
- SDLC predpostavlja da si faze sledijo ena za drugo: pogosto pa se izkaže, da je razvoj programske opreme nujno iterativen proces
- Predpostavlja, da je na začetku razvoja aplikacije že mogoče opredeliti vse zahteve

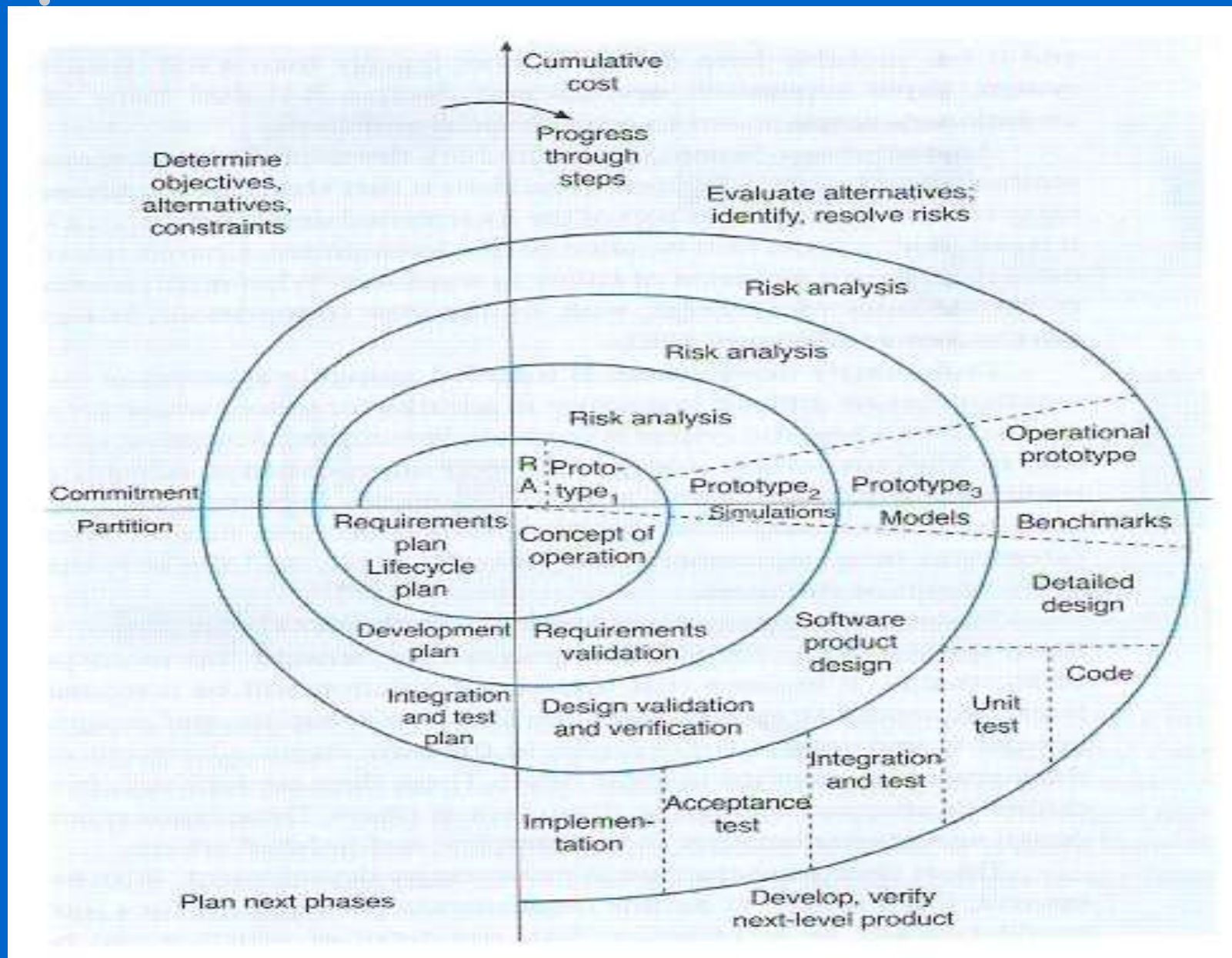
-
-
-
- Da bi presegli pomanjkljivosti razvoja po metodi SDLC, so razvili vrsto metod,
- kot alternativo ali kot dopolnitev pristopu po SDLC:
 - Stopenjski razvoj
 - Prototipni pristop
 - Hiter razvoj aplikacij (Rapid Application development – RAD)
 - JAD (Joint Application Design) delavnice
 - Računalništvo uporabnikov
 -

Stopenjski razvoj (Evolutionary development)

- V več ponovitvah (korakih iteracijah, ...) izdelamo vsakokrat bolj izpopolnjeno aplikacijo, vsakokrat z dopolnjeno funkcionalnostjo. Pri tem velja:
 - zasnova je lahko nepopolna, usmeri naj se na bistvo aplikacije
 - možnosti za nadaljnje spreminjanje aplikacije morajo ostati odprte
 - Izdelana aplikacija lahko pokriva le del uporabnikovih zahtev



Postopni razvoj uporabniških rešitev – časovni potek



http://en.wikipedia.org/wiki/Spiral_model

Spiralni model razvoja in izpopolnjevanja programskih rešitev

Značilnosti stopenjskega razvoja:

- Razmeroma hitro dobimo začetno verzijo aplikacije, izpopolnjuje se potem, ko uporabniki in razvijalci pridobijo nekaj izkušenj z njo
- Spremembe zahtev lahko obravnavamo kot del postopnega razvoja
- Uporabna je tam, kjer so zahteve in cilji aplikacije mehko opredeljeni in se s časom veliko spreminjajo
- Zahteven za vodenje (v primerjavi s pristopom po življenjskem ciklu)