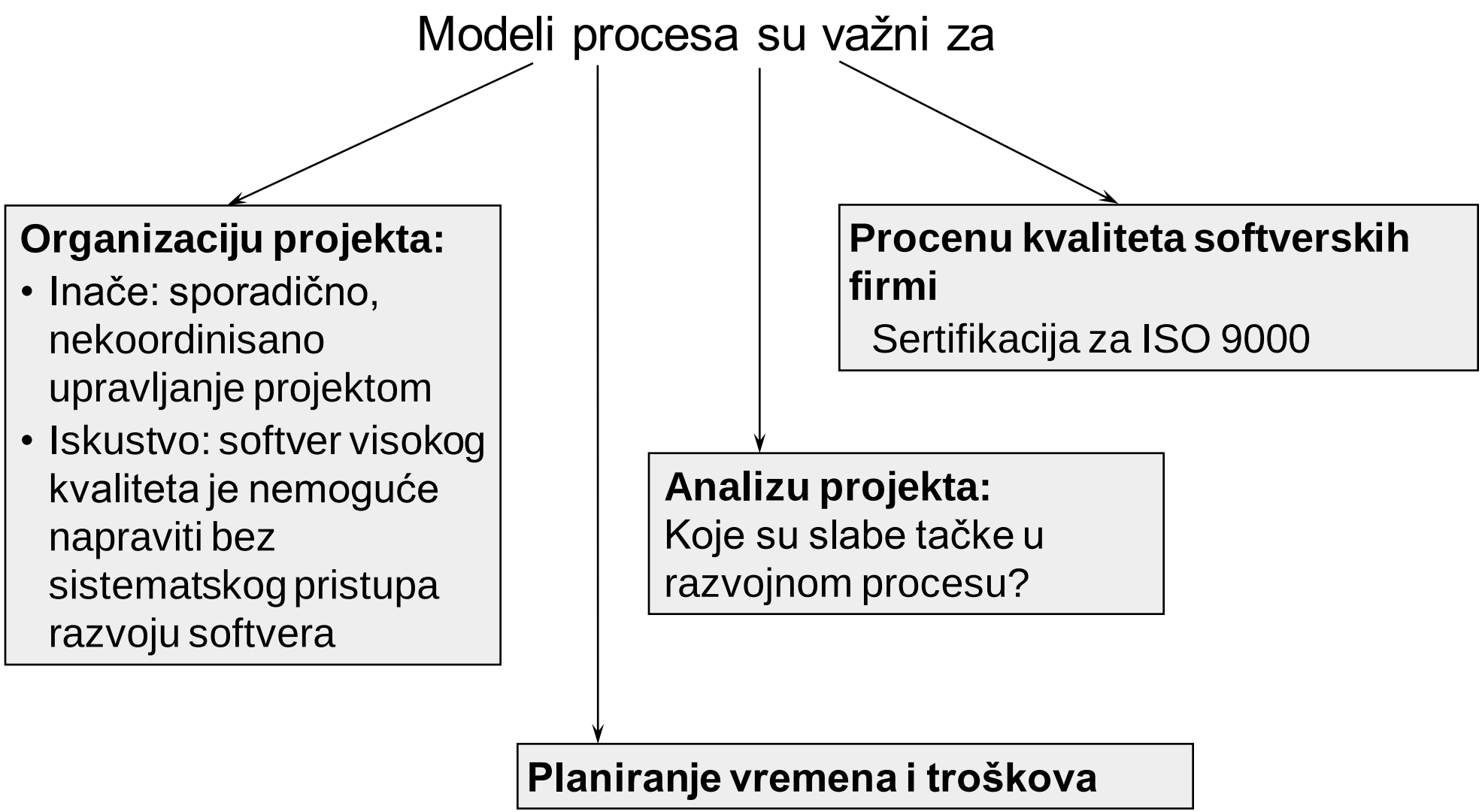
Modeli softverskih procesa

MODELI ŽIVOTNOG CIKLUSA SOFTVERA

- životni ciklus softvera obuhvata period od definicije zahteva do prestanka upotrebe.
- *model životnog ciklusa* opisuje procese iz razvoja, korišćenja i održavanja softverskog proizvoda u toku njegovog životnog ciklusa.
- za konkretan proizvod potrebno je izabrati konkretan model (implementirati standard).
- ne postoji jedinstveni optimalan model za sve softverske proizvode, obično se primenjuje neki od standardnih modela ili neka kombinacija istih.

Važnost modela procesa (životnog ciklusa)



Šta je model procesa?

Model procesa

- **Uopšteno**: razvojni plan koji definiše opšti proces razvoja softverskog proizvoda.
- **Preciznije**: Definicija koja odredjuje koje aktivnosti se izvršavaju, od strane kojih osoba u kojim ulogama; kojim redosledom će aktivnosti biti izvršavane, koji proizvodi će biti razvijani i kako će se procenjivati njihov kvalitet.

Šta je specifično za softver?

Uloga

- Saradnik koji izvršava odredjenu aktivnost
  npr. Test-inženjer, vodja projekta, projektant, programer, softverski ergonomist

3. Modeli procesa

a) Uvod

b) Pregled postojećih modela

c) Model vodopada

d) V-model

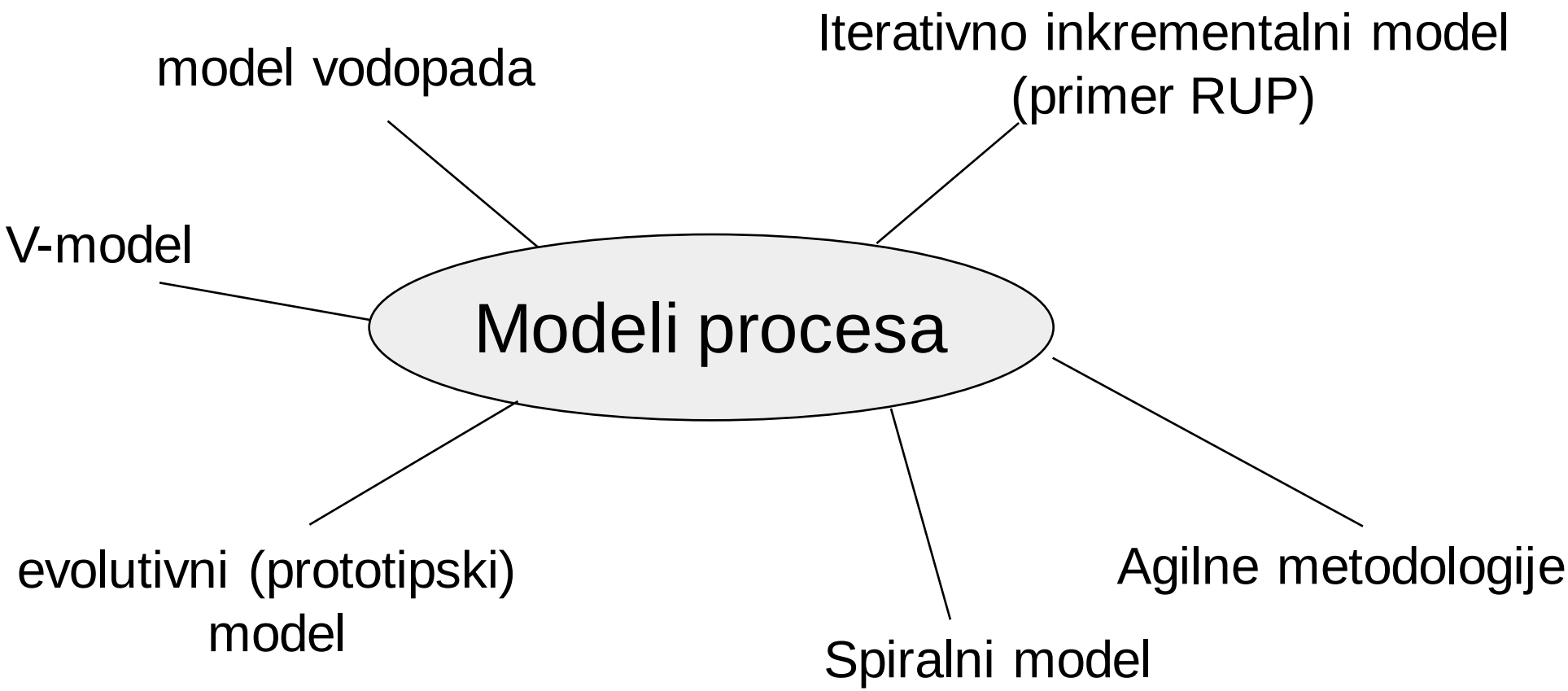
e) Iterativno inkrementalni model

f) Evolutivni (prototipski) model

g) Spiralni model

h) Agilne metodologije

Najpoznatiji modeli procesa: pregled



3. Modeli procesa

a) Uvod

b) Pregled postojećih modela

c) Model vodopada

d) V model

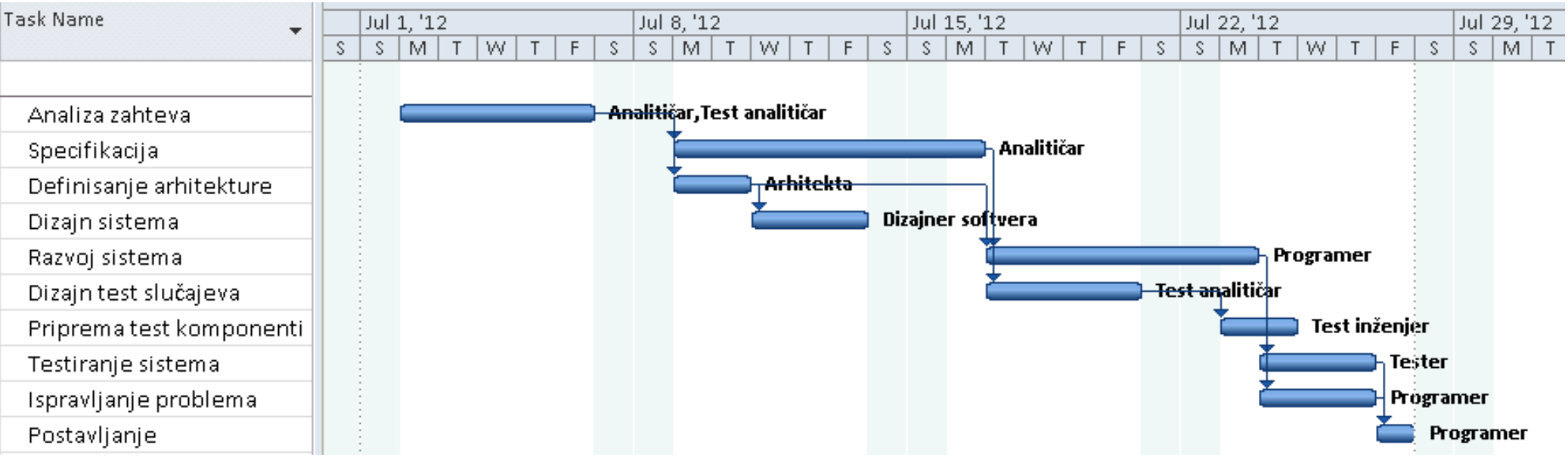
e) Iterativno inkrementalni model

f) Evolutivni (prototipski) model

g) Spiralni model

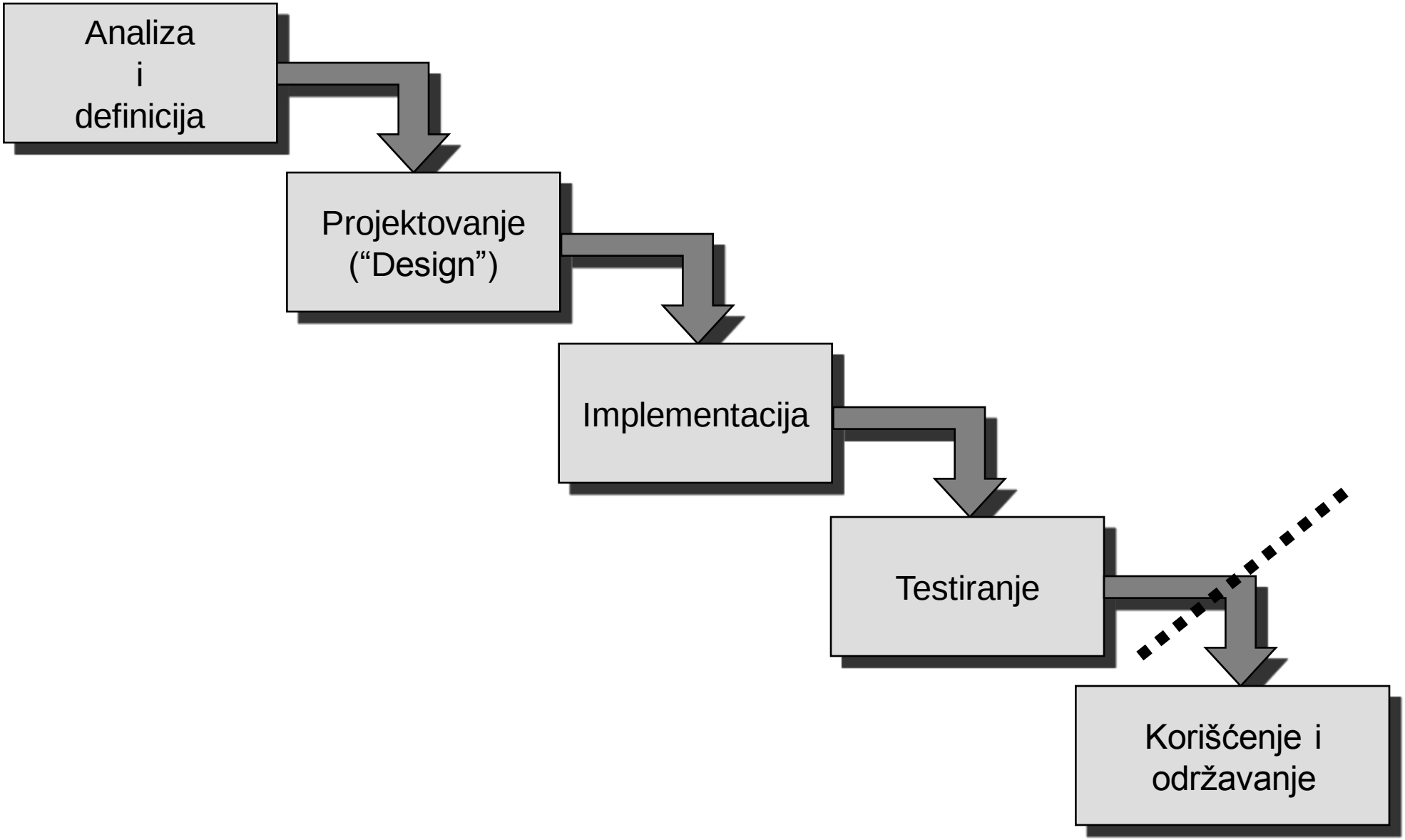
h) Agilne metodologije

Predstavljanje procesa Gantogramom



- Gantov dijagram prikazuje aktivnosti koje je potrebno izvršiti, ko će ih izvršiti, kada će početi, kada će se završiti, kao i koje su međusobne zavisnosti među aktivnostima.
- Na slici se vidi da analitičar i test analitičar zajedno rade na analizi zahteva. Kada završe analizu,analitičar počinje sa izradom specifikacije, a arhitekta sa definicijom arhitekture. Gantogram pored aktivnosti i članova tima koji će raditi na njima prikazuje i međusobne zavisnosti među aktivnostima u vidu aktivnosti koje se moraju izvršiti pre nego što druge počnu. Način na koji se organizuju i grupišu ove aktivnosti predstavlja model razvoja softvera.

Klasični model vodopada (1970)



Model vodopada

- Faze se sekvencijalno izvršavaju dok se ne završi projekat. Faze u modelu vodopada su:
- Analiza zahteva – faza gde se sakupljaju zahtevi od krajnjih korisnika, analizira šta je potrebno raditi, kreiraju ugovori kojima se definiše šta će biti urađeno i potvrđuju zahtevi koji će biti implementirani.
- Dizajn softvera – faza gde se detaljnije analiziraju zahtevi prikupljeni tokom analize, specificira kako će se implementirati softver, kreira tehnička specifikacija i dizajn softvera. Dizajn softvera predstavlja konkretan plan kako će biti implementiran sistem od generalne arhitekture softvera do detaljnog opisa implementacije pojedinih komponenti i algoritama. Na kraju ove faze je poznato kako će se sistem implementirati.
- Implementacija – faza gde se projektovani softver implementira u određenom programskom jeziku i platformi. Na kraju ove faze softver je završen i spreman za upotrebu.
- Verifikacija – faza u kojoj se planira testiranje, testira sistem koji je implementiran u prethodnoj fazi, prijavljuju i otklanjaju problemi nađeni u softveru. Na kraju ove faze softver je testiran i spreman na predaju krajnjim korisnicima.
- Održavanje – tokom faze održavanja softver je predat krajnjim korisnicima i vrše se eventualne dorade u skladu sa izmenama traženim od korisnika. Ova faza traje sve dok korisnici upotrebljavaju softver.

MODEL VODOPADA

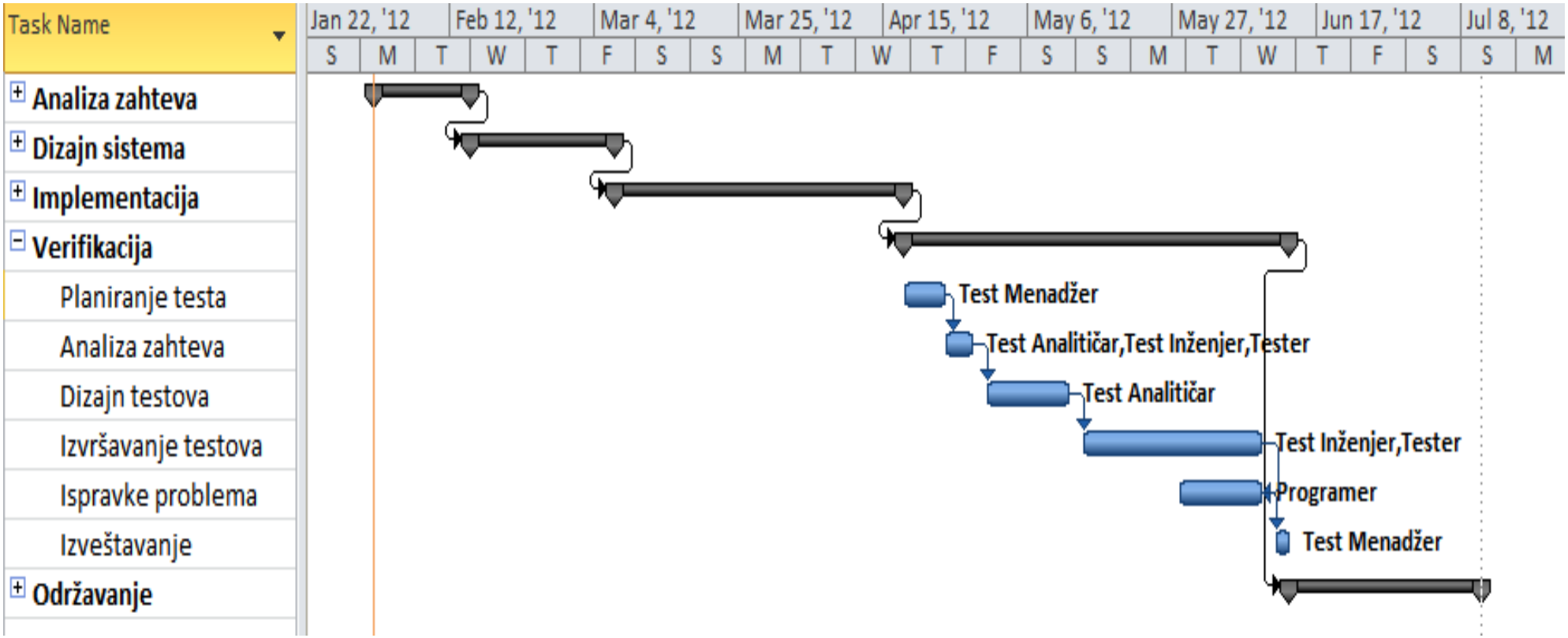
- sekvencijalno izvršavanje pojedinih aktivnosti.
- u stvarnosti postoje povratne grane zbog grešaka i nepreciznosti zahteva.

- Faktori rizika za primenu modela:
  - zahtevi inicijalno nisu precizni
  - sistem preveliki da se uradi u jednom koraku
  - predviđaju se znatne tehnološke promene
  - predviđaju se znatne izmene zahteva
  - ograničeni resuri (novac, ljudstvo)
- Preference za primenu modela:
  - odmah se dobija puna funkcionalnost sistema
  - kada je neophodno odjednom zameniti stari sistem.

Problemi sa vodopadom

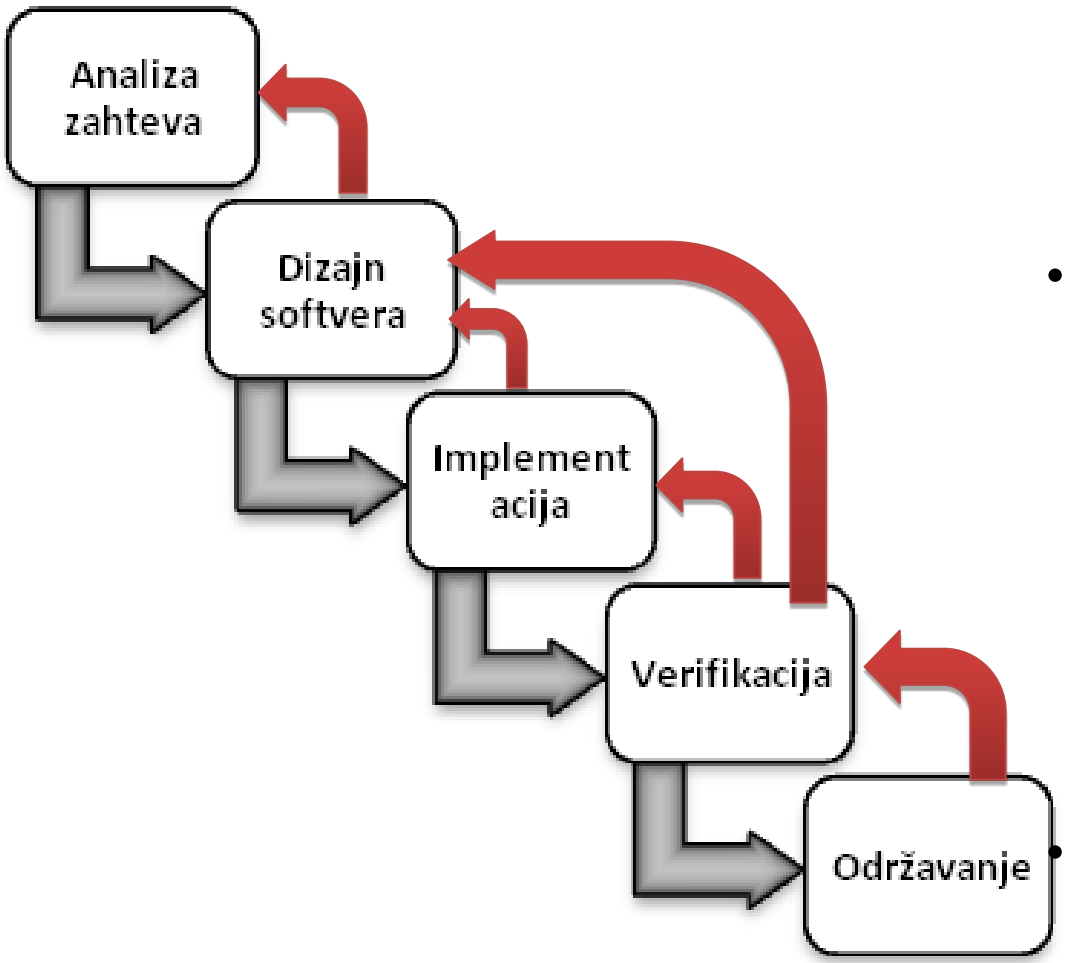
- U idealnom slučaju projekat koji se radi modelom vodopada će prolaziti kroz sve faze razvoja, članovi projektnog tima će se uključivati tačno u one faze u kojima su potrebni, završavaće posao za koji su bili planirani i odlaziće iz tima kada više nema potrebe da budu na projektu. U svakom trenutku će biti poznato dokle se stiglo sa poslom i u kojoj fazi se projekat nalazi.
- Na žalost, ovakav idealan slučaj sekvencijalnog razvoja u praksi nije moguć zato što se često neplanski projektni tim vraća u prethodne faze. Često se dešava da se tokom kasnijih faza projekta pronalaze propusti i nepredviđene stvari nasleđene iz prethodnih faza, koje uzrokuju povratke u prethodne faze.
- Rezultat => kašnjenje projekta

Problemi sa vodopadom



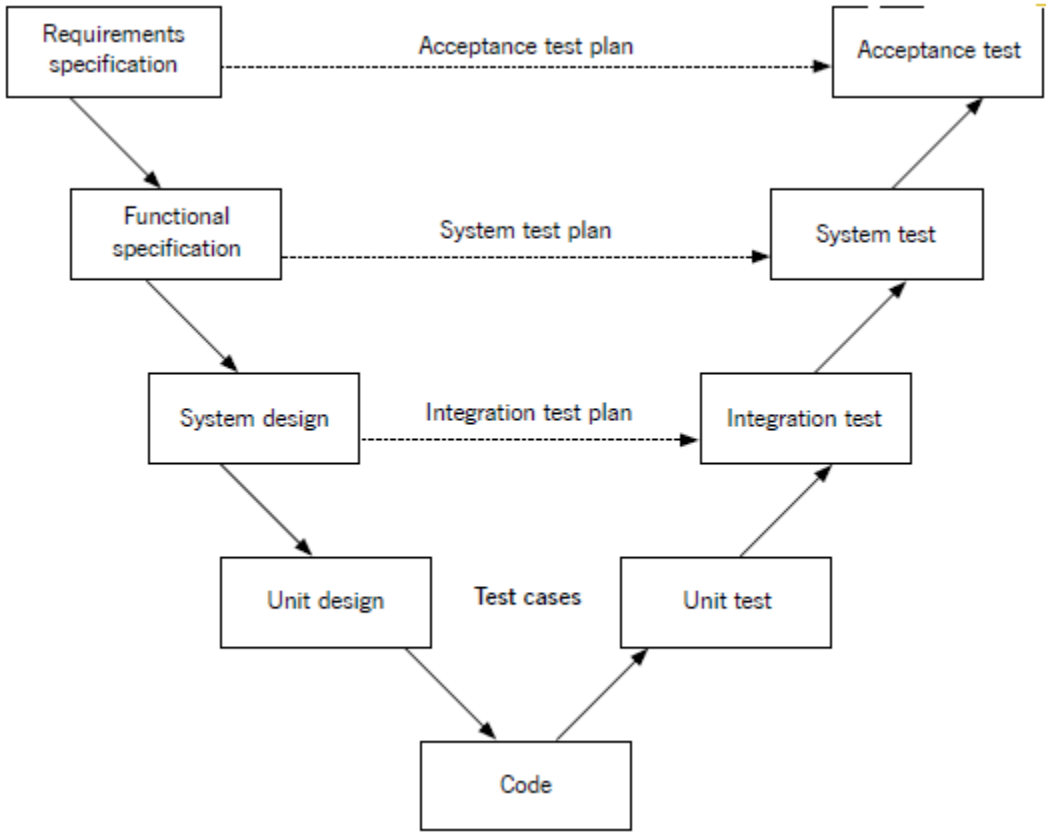
Detalj plana projekta sa fazom verifikacije, koja može da počne pošto se završe prethodne faze

Neželjeni povratni tokovi u modelu vodopada



- Problemi koji se pronađu tokom faze verifikacije mogu biti toliko veliki da vrate projekat nazad u fazu implementacije, a nekada čak iz implementacije nazad u fazu dizajna.
- Na primer, neko može da definiše zahtev u fazi analize, taj zahtev će se dizajnirati i implementirati, a onda tek u fazi verifikacije se zaključi da taj zahtev nema smisla. U tom slučaju projekat se vraća nazad na analizu. Kada se utvrdi šta bi u stvari trebao da bude smislen zahtev opet se prolazi kroz sve faze razvoja.
- Planiranje bazirano na količini rada a ne vremensko (nije pogodno da se dinamički doda ili oduzme deo)

V model sa naglaskom na kontrolu kvaliteta



V-model kao i vodopad ima sve sekvencijalne faze analize, dizajna, programiranja i testiranja gde se u sledeću fazu prelazi samo ako je završena prethodna, ali uz dve značajne izmene:

Svaka faza razvoja ima odgovarajuću fazu testiranja kojom se ta faza validira.

Posle svake faze razvoja, pre prelaska u sledeću fazu, se planira kako će se izvršiti verifikacija trenutne faze (iako se odgovarajuća faza verifikacije neće izvršiti do kraja projekta).

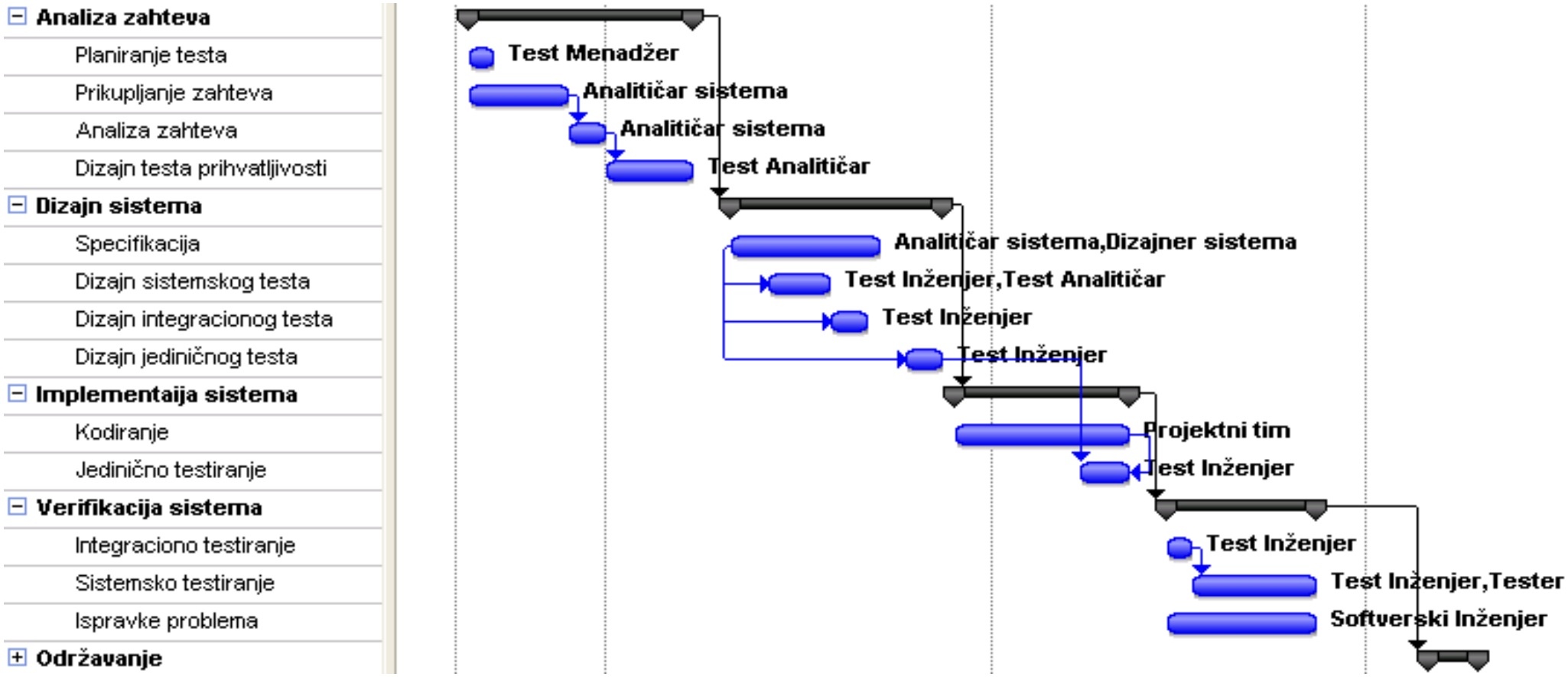
V-model

U V-modelu postoje pravila kojima se definišu preduslovi za prelazak u sledeću fazu:

- Iz faze analize zahteva se može preći u fazu specifikacije samo ako su analizirani zahtevi i ako je definisano kako će se ti zahtevi testirati tokom testa prihvatljivosti.
- U fazu dizajna sistema se može preći ako je završena faza specifikacije sistema i definisano kako će se testirati kompletan sistem.
- U fazu dizajna pojedinih modula se može preći ako je dizajnirana arhitektura sistema i definisano kako će se testirati komponente tokom integracije.
- U fazu kodiranja se može preći ako su dizajnirani moduli koji će se kodirati i ako je definisano kako će se ti moduli testirati.

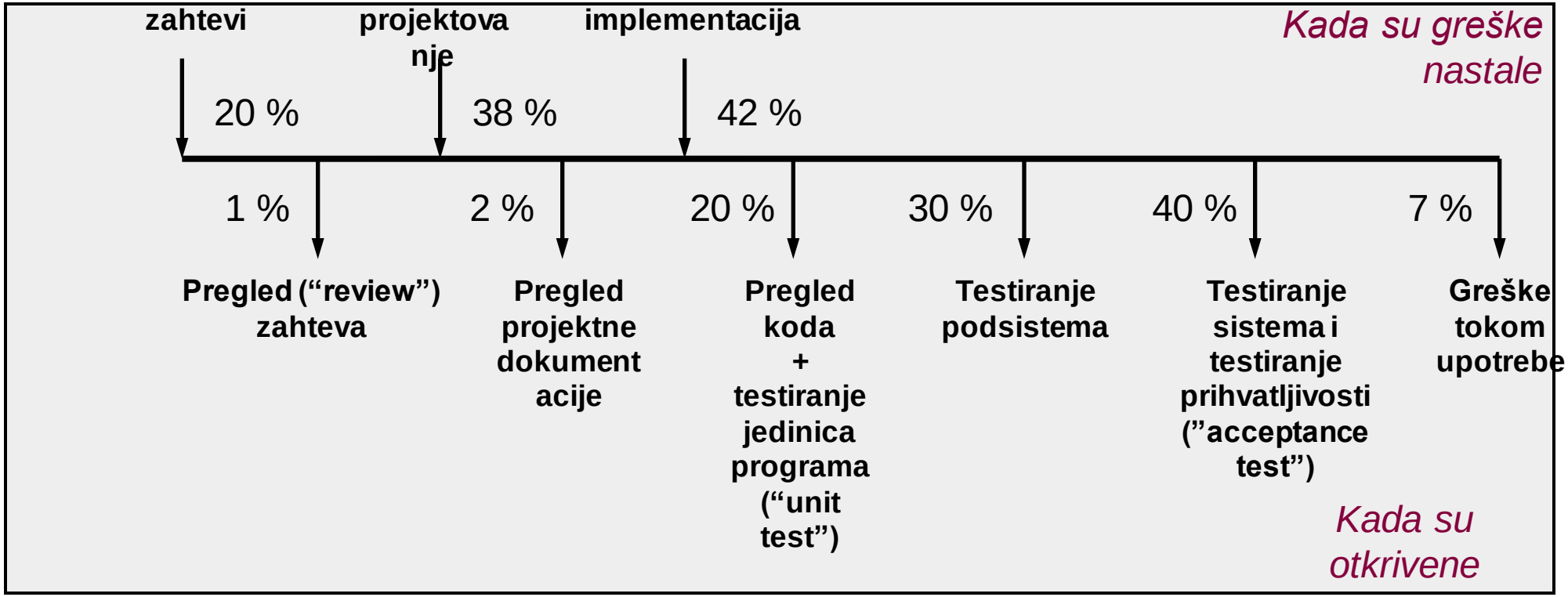
- Druga značajna izmena je definisanje više nivoa testiranja kojima se proveravaju različiti delovi sistema. Nivoi testiranja po V-modelu su:
  - Jedinično testiranje kojim se testiranju pojedini delovi sistema (moduli, komponente, forme).
  - Integraciono testiranje kojim se testira komunikacija, povezivanje i tokovi među modulima.
  - Sistemsko testiranje kojim se testira sistem u celini.
  - Test prihvatljivosti kojim krajnji korisnici potvrđuju da aplikacija radi upravo ono što im treba.

Primer projektnog plana po V-modelu



- Neke aktivnosti testiranja vrše se u paraleli sa prethodnim fazama

Statistika grešaka: nastanak i ispravka



Izvor: Software Metrics Symposium 1996, p. 176

3. Modeli procesa

a) Uvod

b) Pregled postojećih modela

c) Model vodopada

d) V model

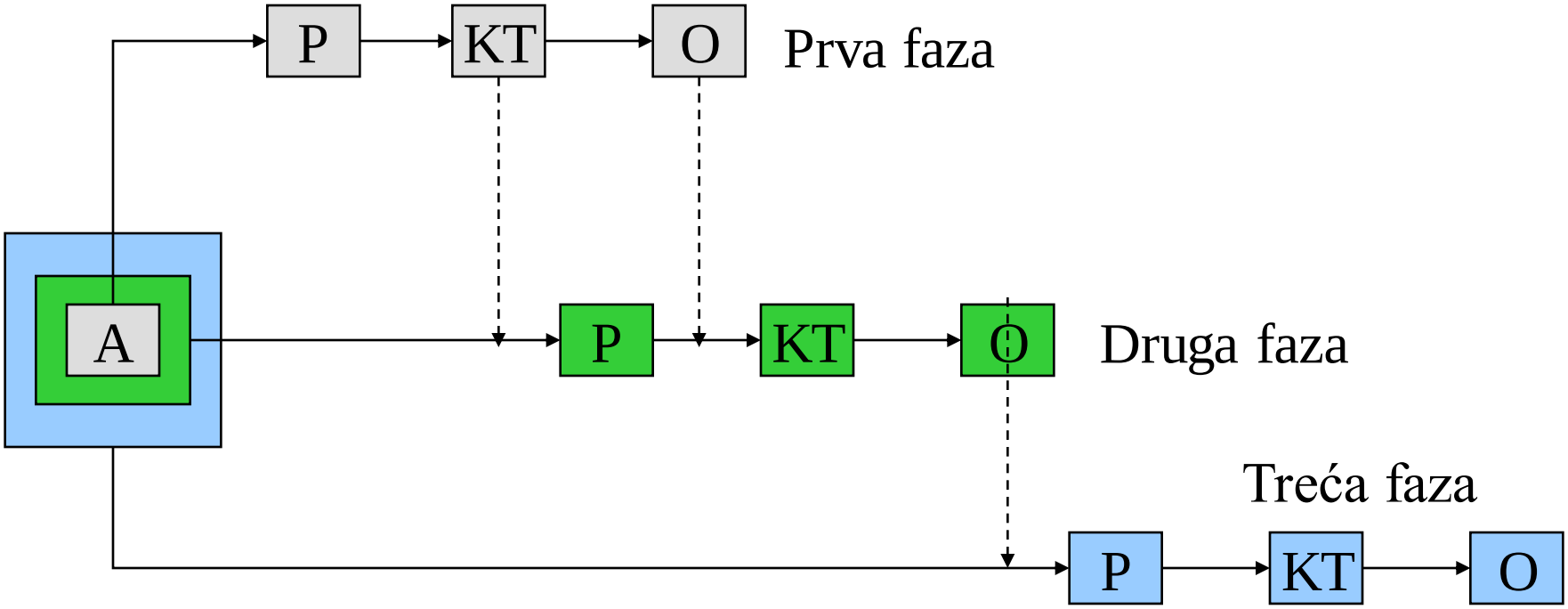
e) Iterativno inkrementalni model

f) Evolutivni (prototipski) model

g) Spiralni model

h) Agilne metodologije

ITERATIVNO INKREMENTALNI MODEL



A -analiza

P - projektovanje

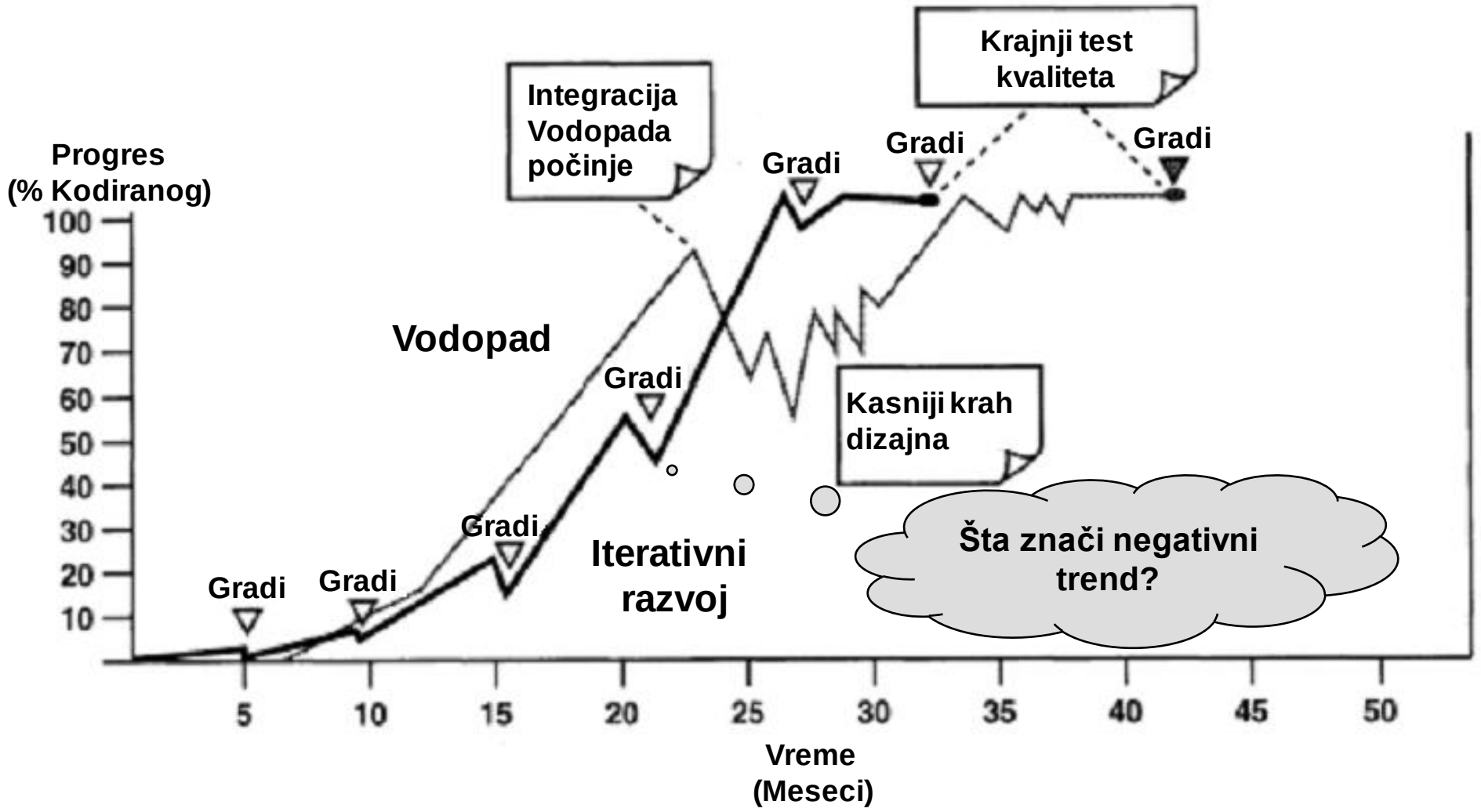
KT - kodiranje i testiranje

O - korišćenje

ITERATIVNO INKREMENTALNI MODEL

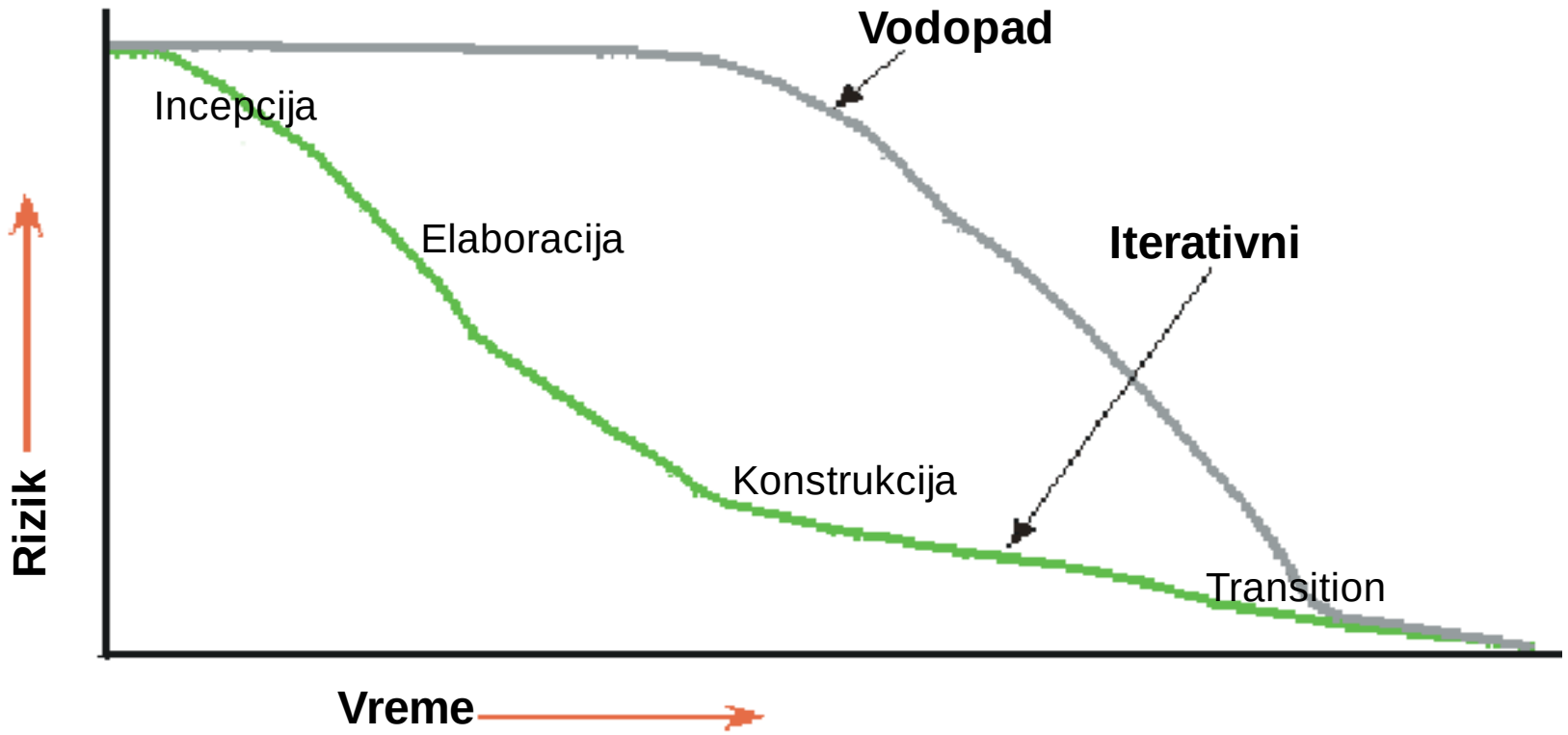
- startuje sa poznatim skupom zahteva, a razvoj se odvija po fazama (inkrementima). Prva faza uključuje deo zahteva, svaka sledeća realizuje deo preostalih zahteva i tako dalje, dok se sistem ne kompletira. U okviru svake faze aktivnosti razvoja se sprovode sekvencijalno, a među fazama može postojati delimično preklapanje aktivnosti.
- Druga izmena u odnosu na model vodopada je uvođenje iteracija. Za razliku od dugih faza u vodopadu koje mogu trajati i po nekoliko meseci, u ovom modelu se faze dele na kratke iteracije od po dve ili tri nedelje u kojima se očekuje da će se proizvesti neka zaokružena celina (završena lista zahteva, prototip, specifikacija itd).

Iterativni naspram Vodopada



Source: I. Jacobson, G. Booch, J. Rumbaugh: The Unified Software Development Process, 1999

Iterativni naspram Vodopada (2)



IBM/Rational Unified Process, v2003

ITERATIVNO INKREMENTALNI MODEL

- Faktori rizika:
  - zahtevi inicijalno nisu precizni
  - zahteva se da sistem odmah ima punu funkcionalnost
  - predviđaju se znatne tehnološke promene
  - predviđaju se znatne izmene zahteva
  - resursi se ne mogu vezati na duži rok
- Preference:
  - Potrebno je odmah obezbediti delimičnu funkcionalnost.
  - Sistem se prirodno deli u inkrementalne celine.
  - Inkrementalni karakter materijalnih i/ili ljudskih resursa.

3. Modeli procesa

a) Uvod
b) Pregled postojećih modela
c) Model vodopada
d) V model
e) Iterativno inkrementalni model
f) Evolutivni (prototipski) model
g) Spiralni model
h) Agilne metodologije

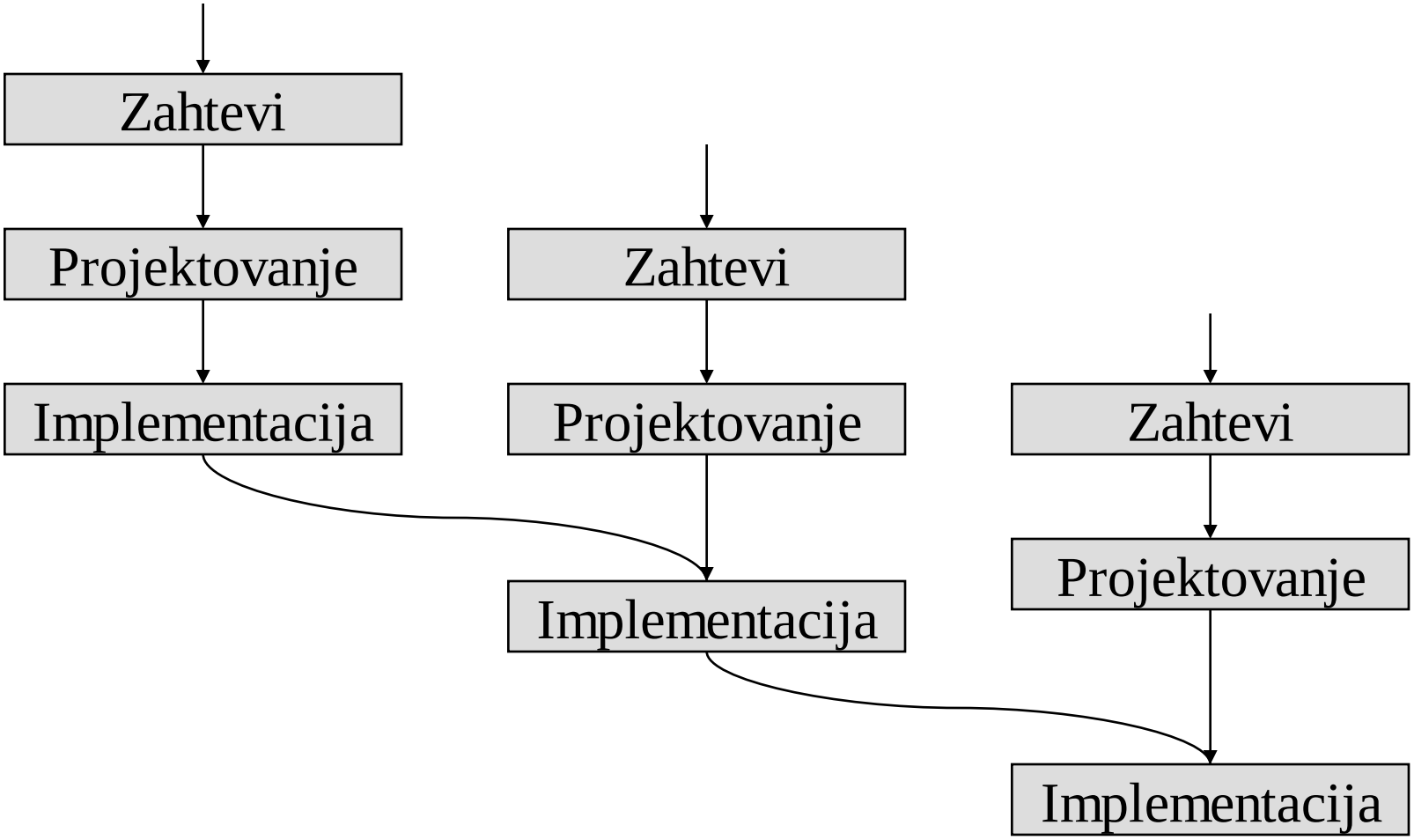
Evolutivni prototipski model

- **Područja primene:**
  Zahtevi u početku nisu precizni ili se često menjaju

- **Prototip:**
  izvršivi softverski sistem,
  - Značajni delovi završnog proizvoda su već završeni (npr. Korisnički interfejs, osnovna funkcionalnost),
  - Ostali delovi tek treba da se realizuju (npr. specijalni slučajevi)

- **Proces primene:**
  - Prototip (može biti odbačen)
    (dodatak analizi zahteva: *brzo pravljenje prototipova ("rapid prototyping")*
  - Postupno napredovanje ka finalnom proizvodu
    (*evolutivni razvoj softvera ("evolutionary software development")*)

EVOLUTIVNI MODEL



EVOLUTIVNI MODEL

- razvija sistem po fazama, ali za razliku od inkrementalnog modela dopušta da zahtevi inicijalno nisu sasvim precizirani i definisani. Zahtevi se inicijano parcijalno definišu i preciziraju u kasnijim fazama.

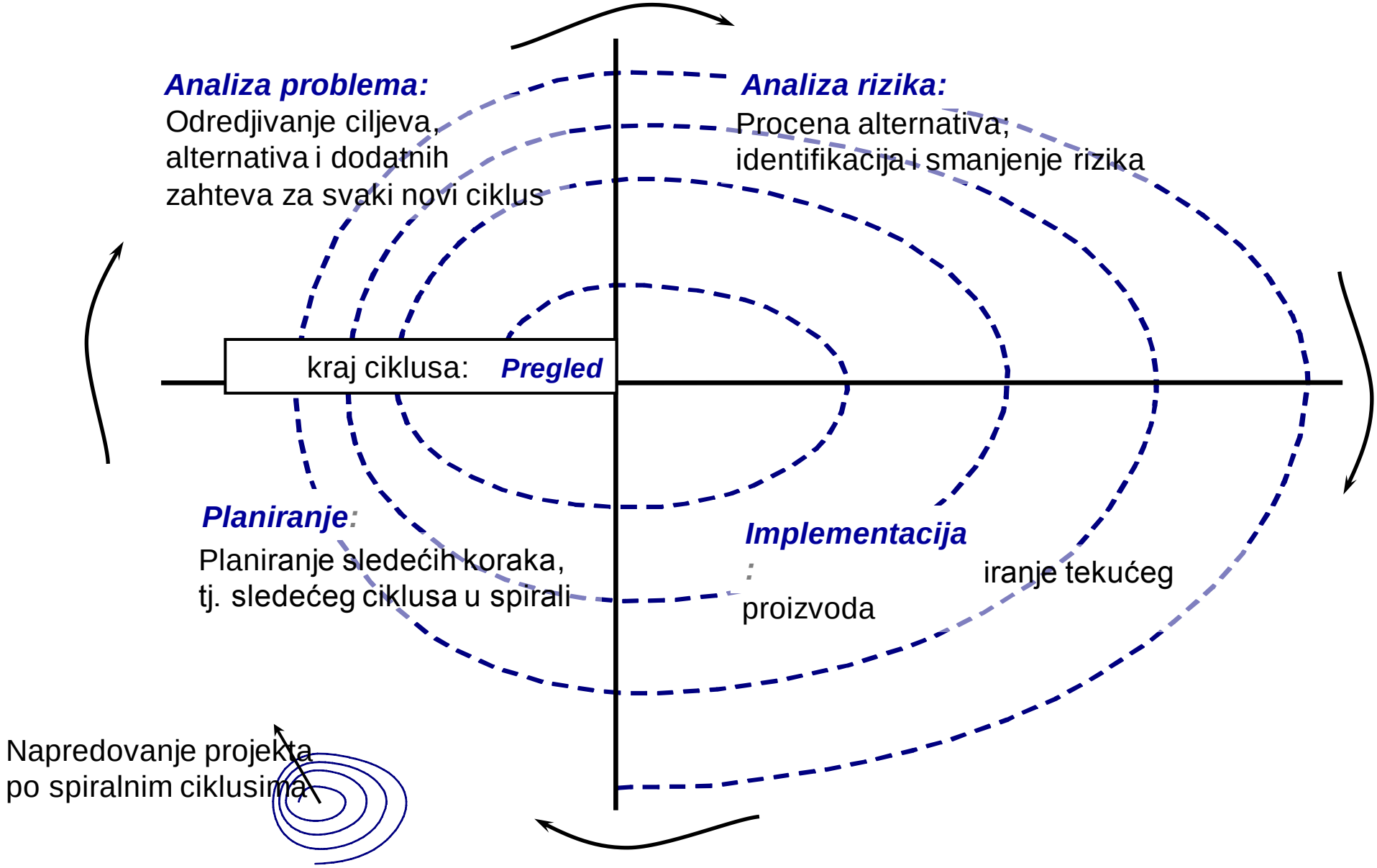
- Faktori rizika:
  - zahteva se da sistem odmah ima punu funkcionalnost
  - resursi se ne mogu vezati na duži rok
  - loša struktura softvera, teškoća za održavanje
  - teškoće u praćenju napredovanja razvoja

- Preference:
  - Potrebno je odmah obezbediti delimičnu funkcionalnost.
  - Sistem se prirodno deli u inkrementalne celine.
  - Inkrementalni karakter materijalnih i/ili ljudskih resursa.
  - Potrebna je povratna veza sa korisnikom da bi se u potpunosti sagledali zahtevi. inteligenciju).
  - Omogućava praćenje tehnoloških promena.

3. Modeli procesa

a) Uvod
b) Pregled postojećih modela
c) Model vodopada
d) V model
e) Iterativno inkrementalni model
f) Evolutivni (prototipski) model
g) Spiralni model
h) Agilne metodologije

Spiralni model po Boehm-u (1988)



Napredovanje projekta po spiralnim ciklusima

SPIRALNI MODEL

- dopunjava evolutivni model uzimajući u obzir potrebe upravljanja velikim projektima. Razvoj ide po fazama i vođen je ciljem smanjivanja rizika neuspeha. Uspešan završetak faze znači smanjenje rizika. Najriskantniji delovi se prvo realizuju. U pojedinim fazama se mogu koristiti makete (bacaju se) ili prototipovi (prerastaju u proizvod) za preciziranje specifikacije, procenu rizika i slično.