



Softversko inženjerstvo

Elektronski fakultet Niš

Softverski procesi



Elektronski fakultet u Nišu



Sadržaj

- ✧ Modeli softverskih procesa
- ✧ Aktivnosti u procesu
- ✧ Kopiranje i menjanje
- ✧ Rational Unified Process (RUP)
 - Primer modernog softverskog procesa



Elektronski fakultet u Nišu



Softverski proces

- ✧ Struktuiran skup aktivnosti neophodan za razvoj softverskog sistema.
- ✧ Aktivnosti zajedničke za različite procese:
 - Specifikacija – definisanje šta sistem treba da radi;
 - Projektovanje i implementacija – definisanje organizacije sistema i njegova implementacija;
 - Validacija – provera da li sistem radi ono što naručilac želi;
 - Evolucija – promena sistema kao odgovor na promenu potreba naručilaca.
- ✧ Model softverskog procesa predstavlja apstraktnu reprezentaciju procesa. Reč je o opisu procesa iz određene perspektive.



Elektronski fakultet u Nišu



Opis softverskih procesa

- ✧ Kada se opisuju i diskutuju procesi, najčešće se govori o aktivnostima koje se sprovode u okviru procesa (npr. definisanje modela podataka, projektovanje korisničkog interfejsa, itd.) i o njihovom redosledu.
- ✧ Opisi procesa mogu takođe uključiti:
 - Proizvode, koji predstavljaju izlaz neke aktivnosti procesa;
 - Role, koje označavaju odgovornosti ljudi uključenih u proces;
 - Preduslovi i posledice, koje se vezuju za pojedine aktivnosti u procesu.

Planom vođeni i agilni procesi

- ❖ Planom vođeni procesi su procesi kod kojih se sve aktivnosti unapred planiraju, te se i progres meri na osnovu tog plana.
- ❖ Kod agilnih procesa planiranje je inkrementalno, pa je lakše promeniti proces nakon promene zahteva naručioca.
- ❖ U praksi najpraktičniji procesi uključuju elemente kako planom vođenog tako i agilnog pristupa.
- ❖ Ne postoje dobri i loši softverski procesi.

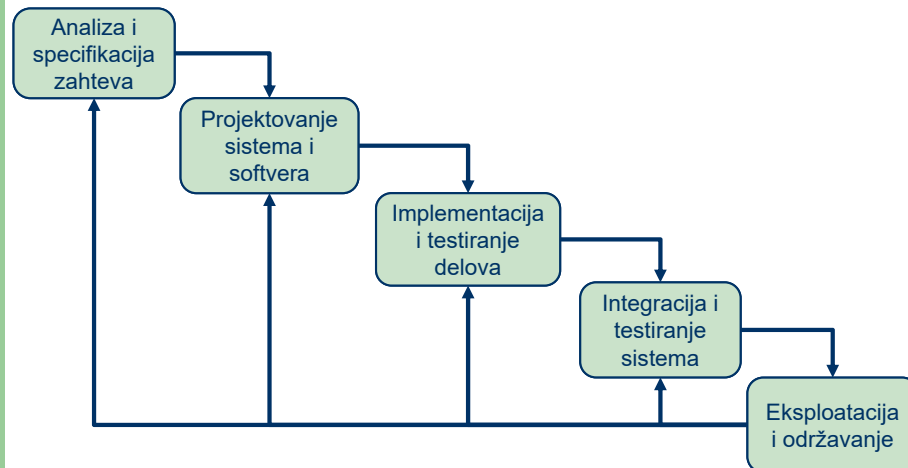
5

Modeli softverskih procesa

- ❖ Model vodopada
 - Planom vođen model. Potpuno odvojene faze specifikacije i razvoja.
- ❖ Inkrementalni razvoj
 - Specifikacija, razvoj i validacija se preklapaju. Može biti planom vođen ili agiln.
- ❖ Razvoj zasnovan na korišćenju gotovih komponenti
 - Sistem se sklapa od gotovih komponenti. Može biti planom vođen ili agiln.
- ❖ U praksi veliki sistemi se najčešće razvijaju procesima koji inkorporiraju elemente sva tri prethodna modela.

6

Model vodopada



7

Faze u modelu vodopada

- ❖ Model vodopada identifikuje sledeće odvojene faze:
 - Analiza i specifikacija zahteva
 - Projektovanje sistema i softvera
 - Implementacija i testiranje delova
 - Integracija i testiranje sistema
 - Eksploatacija i održavanje
- ❖ Osnovni nedostatak ovog modela je nemogućnost efikasnog prihvatanja izmena u zahtevima korisnika kada je proces u toku. U principu, jedna faza mora biti završena da bi se otpočela naredna, pa izmene zahteva resetuju čitav proces.

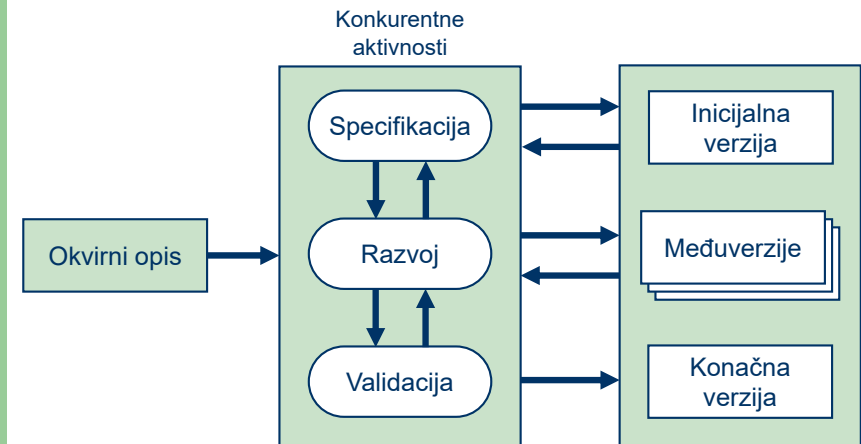
8

Problemi sa modelom vodopada

- ❖ Nefleksibilna deoba projekta na disjunktne faze otežava odgovor na promenu korisničkih zahteva.
 - Zbog ovoga je model vodopada jedino prihvatljiv kada su zahtevi dobro definisani i usaglašeni na početku projekta sa malom verovatnoćom izmene u toku razvoja.
 - Najčešće zahtevi nisu u toj meri jasni na početku razvoja.
- ❖ Model vodopada se najčešće koristi kod velikih sistema gde se razvoj razdeljen na nekoliko lokacija.
 - U ovim situacijama planom vođena priroda modela vodopada pomaže u koordinisanju posla.

9

Inkrementalni razvoj



10

Prednosti inkrementalnog razvoja

- ❖ Smanjena cena izmene korisničkih zahteva u toku razvoja.
 - Količina analiza i dokumentacije koju treba ponovno uraditi je mnogo manja u odnosu na model vodopada.
- ❖ Olakšano dobijanje povratne informacije od korisnika u toku razvoja.
 - Korisnici imaju uvid i mogućnost komentarisanja trenutno implementiranih funkcionalnosti, a samim tim i u progres realizacije projekta.
- ❖ Moguća brža isporuka korisnog softvera naručiocu.
 - Korisnici mogu da koriste softver pre nego u slučaju modela vodopada.

11

Problemi inkrementalnog razvoja

- ❖ Proces nije vidljiv.
 - Menadžerima su potrebne redovne isporuke kako bi merili napredovanje. Ako se sistem brzo razvija nije efikasno praviti dokumentaciju za svaku verziju sistema.
- ❖ Struktura sistema ima tendenciju degradacije sa dodavanjem novih inkremenata.
 - Ukoliko se ne ulaže novac u refaktorisanje i unapređenje softvera, periodične izmene imaju za posledicu degradaciju strukture. Nove i nove izmene softvera postaju sve složenije i skuplje.

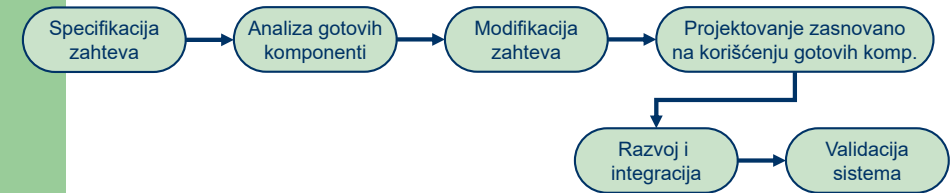
12

Razvoj zasnovan na korišćenju gotovih komponenti

- ❖ Zasnovan na sistematskog korišćenju ranije razvijenih komponenti ili komercijalno dostupnih komponenti (COTS - *Commercial-off-the-shelf*).
- ❖ Faze procesa
 - Analiza komponenti;
 - Modifikacija zahteva;
 - Projektovanje sistema korišćenjem gotovih komponenti;
 - Razvoj i integracija.
- ❖ Korišćenje gotovih komponenti je standardni pristup za razvoj mnogih poslovnih sistema.

13

Razvoj zasnovan na korišćenju gotovih komponenti



14

Tipovi softverskih komponenti

- ❖ Veb servisi koji su razvijeni na osnovu određenih standarda u pogledu interfejsa.
- ❖ Kolekcije objekata organizovanih u pakete sa mogućnošću ugradnje u odgovarajuće komponentne okvire (npr. .NET or J2EE).
- ❖ Nezavisni softverski sistemi (COTS) koje je moguće konfigurisati za korišćenje u odgovarajućem okruženju.

15

Aktivnosti u okviru procesa

- ❖ Realni softverski procesi predstavljaju isprepletenu mešavinu tehničkih, kolaborativnih i menadžerskih aktivnosti čiji je krajnji cilj specifikacija, projektovanje, implementacija i testiranje softverskog sistema.
- ❖ Četiri osnovne aktivnosti softverskog procesa specifikacija, razvoj, validacija i evolucija su različito organizovane u različitim procesima. U modelu vodopada aktivnosti su organizovane sekvencijalno, dok se kod inkrementalnog razvoja preklapaju.

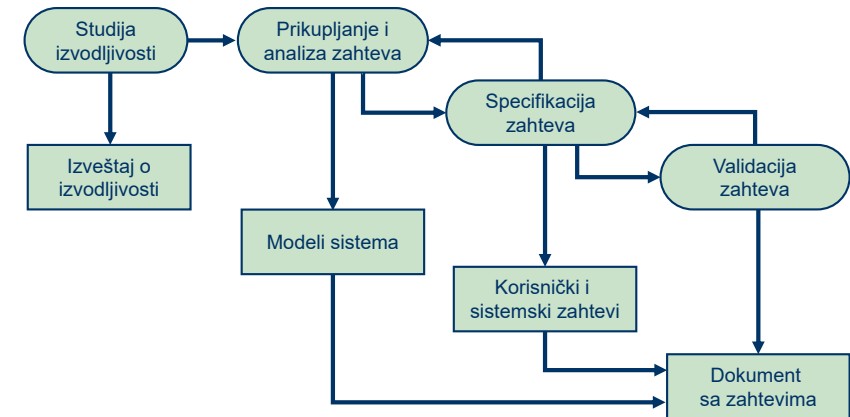
16

Specifikacija softvera

- ❖ Specifikacija softvera je proces određivanja servisa koji su zahtevani od strane korisnika, kao i ograničenja u pogledu funkcionisanja i razvoja sistema.
- ❖ Proces inženjeringa zahteva
 - Studija izvodljivosti
 - Provera da li je tehnički i finansijski izvodljiv razvoj sistema
 - Prikupljanje i analiza zahteva
 - Određivanje šta zainteresovane strane žele ili očekuju od sistema
 - Specifikacija zahteva
 - Detaljno definisanje zahteva
 - Validacija zahteva
 - Provera validnosti specificiranih zahteva

17

Proces inženjeringa zahteva



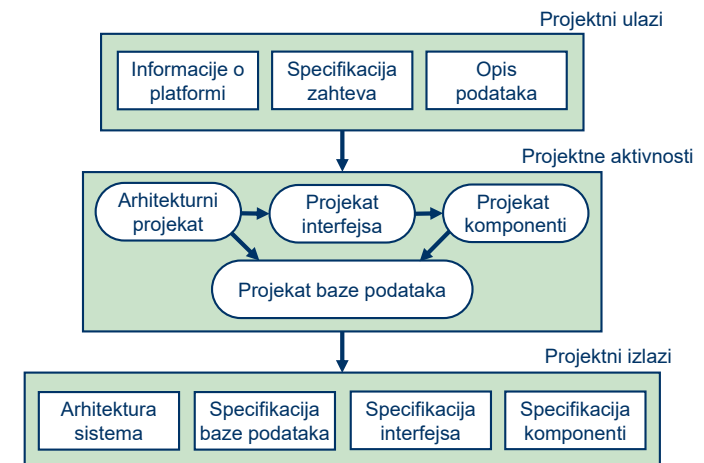
18

Projektovanje i implementacija softvera

- ❖ Reč je o procesu prevođenja specifikacije sistema u realni upotrebljiv sistem.
- ❖ Projektovanje softvera
 - Projektovanje strukture softvera tako da realizuje zadatu specifikaciju.
- ❖ Implementacija
 - Prevođenje definisane strukture u izvršni program.
- ❖ Aktivnosti projektovanja i implementacije su blisko povezane i često se preklapaju.

19

Generalni model procesa projektovanja



20

Aktivnosti u projektovanju

- ❖ *Arhitekturno projektovanje* u okviru kog se identifikuje osnovna struktura sistema, osnovne komponente (često se nazivaju podsistemi ili moduli), njihova međusobna veza i način distribucije.
- ❖ *Projektovanje interfejsa* u okviru kog se definišu interfejsi između komponenti sistema.
- ❖ *Projektovanje komponenti* u okviru kog se za svaku identifikovanu komponentu projektuje željeno funkcionisanje.
- ❖ *Projektovanje baze podataka* u okviru kog se projektuju strukture podataka sistema i način njihove reprezentacije u bazi podataka.

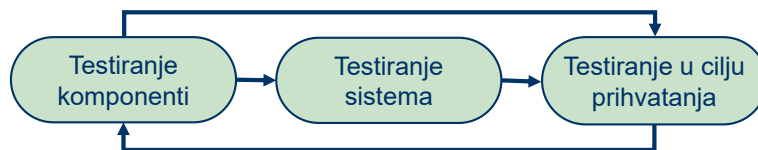
21

Validacija softvera

- ❖ Verifikacija i validacija (V&V) ima za cilj da pokaže da sistem odgovara sopstvenoj specifikaciji i zahtevima naručioca.
- ❖ Uključuje proveru i pregled procesa, kao i testiranje sistema.
- ❖ Testiranje sistema uključuje izvršenje test slučajeva koji se izvode iz specifikacije realnih podataka koje sistem obrađuje.
- ❖ Testiranje je najčešće korišćenja V&V aktivnost.

22

Faze u testiranju



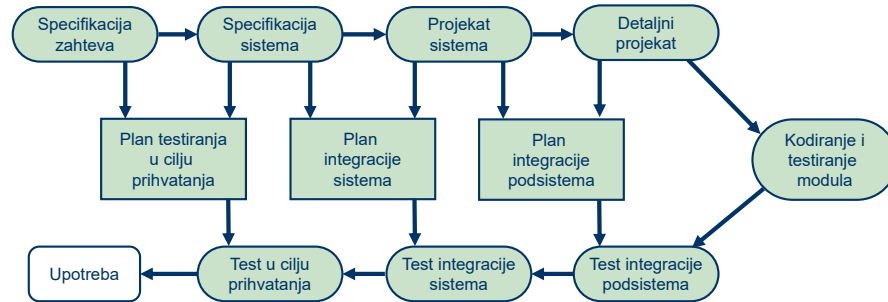
23

Faze u testiranju

- ❖ Razvojno ili testiranje komponenti
 - Pojedinačne komponente se testiraju nezavisno;
 - Komponente mogu biti funkcije ili objekti, kao i koherentne grupe ovih entiteta.
- ❖ Testiranje sistema
 - Testiranje sistema kao celine. Posebno je značajno testiranje bitnih svojstava sistema.
- ❖ Testiranje u cilju prihvatanja
 - Testiranje sa podacima koje obezbeđuju naručioci u cilju provere da li sistem odgovara potrebama korisnika.

24

Test faze u planom-vođenom razvoju softvera



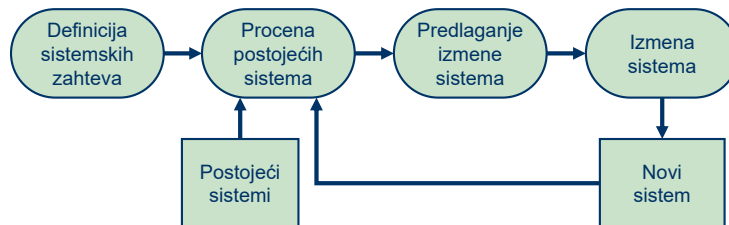
25

Evolucija softvera

- ❖ Softver je podrazumevano fleksibilan i podložan izmenama.
- ❖ Kako se zahtevi menjaju kroz promenu poslovnih okolnosti, softver koji podržava odgovarajući posao mora da evoluira i da se menja.
- ❖ Iako često dolazi do mešanja između novog razvoja i evolucije (održavanja) postojećeg sistema, sama razlika postaje sve manje relevantna pošto imamo sve manje potpuno novih sistema.

26

Evolucija sistema



27

Rezime

- ❖ Softverski proces čine aktivnosti uključene u proizvodnju softverskog sistema. Model softverskog procesa je apstraktna reprezentacija odgovarajućih procesa.
- ❖ Opšti modeli procesa opisuju organizaciju softverskih procesa. Primeri tih opštih modela su: model vodopada, inkrementalni razvoj i razvoj zasnovan na korišćenju gotovih komponenti.

28

Rezime

- ✧ Inženjering zahteva predstavlja proces razvoja specifikacije softvera.
- ✧ Projektovanje i implementacija se bave prevođenjem specifikacije zahteva u izvršni softverski sistem.
- ✧ Validacija softvera je proces provere da li sistem odgovara sopstvenoj specifikaciji, kao i stvarnim potrebama korisnika.
- ✧ Evolucija softvera predstavlja izmenu postojećeg softverskog sistema na osnovu novo postavljenih zahteva. Softver mora da evoluira da bi bio koristan.

29

Softverski procesi

2. čas

30

Kopiranje i menjanje

- ✧ Izmene su neizbežne u svim velikim softverskim projektima.
 - Poslovne promene dovode do pojave novih i promene postojećih zahteva
 - Nove tehnologije otvaraju nove mogućnost u pogledu poboljšane implementacije sistema
 - Promena platforme takođe zahteva izmene u aplikaciji
- ✧ Bez obzira na koji način je promena nastala potrebna je prepravka softvera. Cena prepravke uključuje rad na ponovnoj analizi i specifikaciji zahteva, kao i na implementaciji nove funkcionalnosti.

31

Smanjivanje cene prepravki

- ✧ Izbegavanje izmena, kad softverski proces uključuje aktivnosti koje mogu da predvide moguće izmene pre nego što je za njih potrebno uložiti značajan trud.
 - Na primer, moguće je razviti prototip sistema kako bi se neke najznačajnije funkcionalnosti prikazale naručiocu.
- ✧ Tolerancija na izmene, kad je proces tako prilagođen da se izmene mogu sprovesti po relativno niskoj ceni.
 - Ovo normalno zahteva neki oblik inkrementalnog razvoja. Predložena izmena se može implementirati u inkrementima koji su planirani, a još nisu razvijeni, ili ako je to nemoguće u novom inkrementu.

32

Prototipovanje softvera

- ✧ Prototip je inicijalna verzija sistema koja se koristi u cilju demonstracije koncepata i isprobavanja projektnih opcija.
- ✧ Prototip se može koristiti u:
 - Procesu inženjeringa zahteva kako bi potpomogao prikupljanje i validaciju zahteva;
 - U procesu projektovanja kako bi se istražile moguće opcije i razvio korisnički interfejs;
 - U procesu testiranja kako bi se obezbedili uporedni testovi.

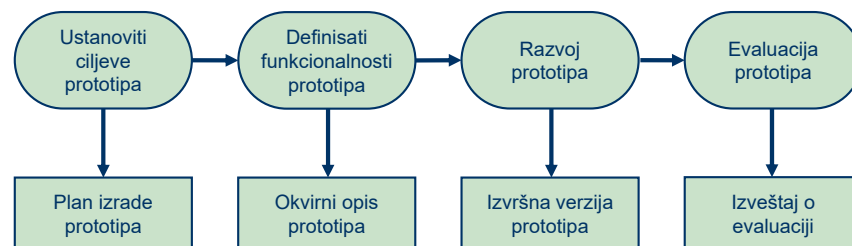
33

Prednosti prototipovanja

- ✧ Unapređena upotrebljivost sistema (*usability*).
- ✧ Bolji odgovor na stvarne korisničke potrebe.
- ✧ Poboljšan kvalitet projekta.
- ✧ Poboljšana sposobnost održavanja (*maintainability*).
- ✧ Smanjen ukupan trud potreban za razvoj softvera.

34

Proces razvoja prototipa



35

Razvoj prototipa

- ✧ Može biti zasnovan na alatima i jezicima za brz razvoj
- ✧ Može da se izostave neke funkcionalnosti
 - Prototip se fokusira na deo proizvoda koji nije najjasnije shvaćen;
 - Provera grešaka i obrada izuzetaka ne mora biti uključena u prototip;
 - Fokus na funkcionalnim zahtevima, a zanemarivanje nefunkcionalnih zahteva (npr. pouzdanost i sigurnost).

36

Prototipi za bacanje

- ❖ Prototipi bi trebalo da se odbace nakon razvoja jer ne predstavljaju dobru osnovu za produkcionu sistem:
 - Nekada nije moguće naštelovati sistem da zadovolji nefunkcionalne zahteve;
 - Prototipovi se ne dokumentuju;
 - Struktura prototipa je najčešće degradirana zbog učestalih izmena;
 - Prototip najverovatnije neće zadovoljiti normalne standarde kvaliteta organizacije.

37

Inkrementalna isporuka

- ❖ Umesto da se sistem isporuči u jednom koraku, imamo podelu razvoja i isporuke u inkremente gde svaki deo isporuke sadrži deo tražene funkcionalnosti.
- ❖ Korisnički zahtevi se uređuju na osnovu prioriteta, tako da se najprioritetniji zahtevi rade u ranim inkrementima.
- ❖ Kada započne razvoj jednog inkrementa zahtevi koji se odnose na njega se zamrzavaju, dok zahtevi za buduće inkremente mogu da nastave da evoluiraju.

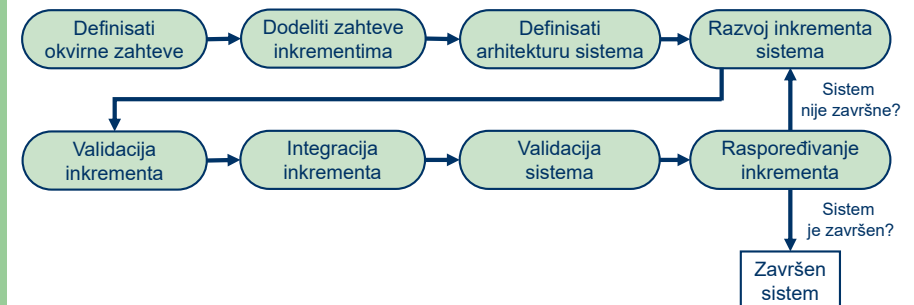
38

Inkrementalni razvoj i isporuka

- ❖ Inkrementalni razvoj
 - Razvoj sistema u inkrementima koji se evaluiraju pre započinjanja razvoja narednog inkrementa;
 - Normalni pristup koji se koristi kod agilnih metoda;
 - Evaluaciju obavlja korisnik/naručilac.
- ❖ Inkrementalna isporuka
 - Isporuka inkrementa krajnjim korisnicima;
 - Realističnija evaluacija praktične upotrebe softvera;
 - Problematična primena kod sistema koji menjaju postojeće sisteme pošto polazni inkrementi nemaju sve funkcionalnosti starog sistema.

39

Inkrementalna isporuka



40

Prednosti inkrementalne isporuke

- ❖ Funkcionalnosti sistema postaju ranije dostupne korisnicima.
- ❖ Rani inkrementi imaju ulogu prototipova kako bi se bolje razumeli zahtevi za kasnije inkremente.
- ❖ Smanjen rizik da projekat kompletno propadne.
- ❖ Funkcionalnosti koje imaju najveći prioritet se rade u startu, pa budu i najviše i najbolje testirane.

41

Problemi inkrementalne isporuke

- ❖ Većina sistema poseduje skup osnovnih konstrukcija koje se koriste u različitim delovima sistema.
 - Kako zahtevi nisu definisani detaljno do početka implementacije inkrementa teško je identifikovati zajedničke konstrukcije koje će koristiti različiti inkreменти.
- ❖ Suština iterativnog procesa je da se specifikacija razvija paralelno sa softverom.
 - Međutim, ovo najčešće dolazi u konflikt sa metodom nabavke kod većine organizacija gde se kompletna specifikacija sistema prilaže uz ugovor o razvoju sistema.

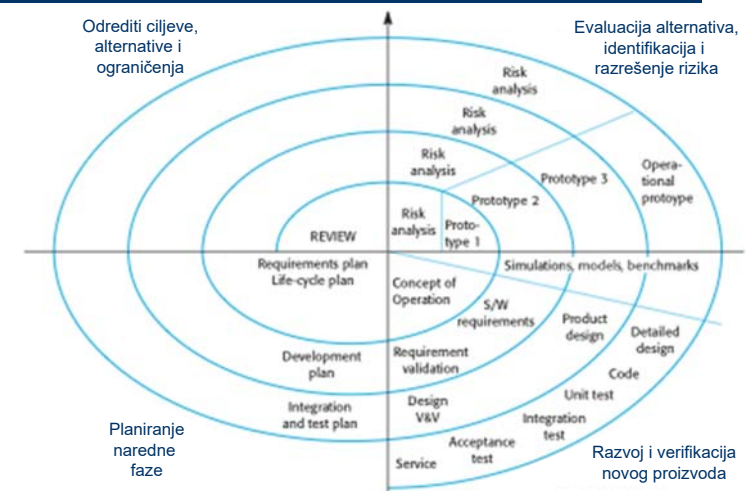
42

Boehm-ov spiralni model

- ❖ Proces je predstavljen spiralom umesto sekvence aktivnosti sa opcionim vraćanjem u nazad.
- ❖ Svaki ciklus u spirali predstavlja jednu fazu procesa.
- ❖ Nema fiksnih faza kao što su specifikacija ili projektovanje – ciklusi se biraju u zavisnosti šta je potrebno.
- ❖ Na moguće rizike se obraća posebna pažnja u toku procesa.

43

Boehm-ov spiralni model softverskog procesa



44

Sektori u spiralnom modelu

- ✧ Postavljanje ciljeva
 - Identifikacija ciljeva za datu fazu.
- ✧ Procena i umanjivanje rizika
 - Vršiti se procena rizika i biraju se odgovarajuće aktivnosti kako bi se rizici smanjili.
- ✧ Razvoj i validacija
 - Bira se razvojni model sistema.
- ✧ Planiranje
 - Vršiti se revizija projekta i planira se naredna faza spirale.

45

Upotreba spiralnog modela

- ✧ Spiralni model je pomogao u razumevanju i prihvatanju ideje o iteracijama u softverskom procesu i ustanovljavanja rizikom vođenog prilaza u razvoju softvera.
- ✧ U praksi je ovaj model retko korišćen.

46

Rational Unified Process (RUP)

- ✧ Moderni proces opšteg tipa zasnovan na UML-u.
- ✧ Povezuje u sebi neka svojstva 3 ranije pominjana opšta modela procesa.
- ✧ Najčešće je opisan iz 3 perspektive
 - Dinamička perspektiva koja prikazuje različite faze u toku vremena;
 - Statička perspektiva koja prikazuje aktivnosti u procesu;
 - Praktična perspektiva koja se ugleda u korišćenju pozitivnih praktičnih iskustava.

47

Faze u RUP-u



48

Faze u RUP-u

- ✧ Započinjanje (*Inception*)
 - Analiza poslovne perspektive.
- ✧ Razrada (*Elaboration*)
 - Razumevanje domena problema i razvoj arhitekture sistema.
- ✧ Izrada (*Construction*)
 - Projektovanje sistema, implementacija i testiranje.
- ✧ Okončanje (*Transition*)
 - Isporuka i instalacija sistema.

49

Iteracije u RUP-u

- ✧ Iteracije u okviru faze
 - Svaka faza je iterativna sa rezultatima koji se razvijaju inkrementalno.
- ✧ Međufazna iteracija
 - Moguće je čitav skup faza ponavljati.

50

Statički tokovi poslova u RUP-u

Tok poslova	Opis
Poslovno modelovanje	Poslovni procesi se modeluju pomoću poslovnih slučajeva korišćenja.
Zahtevi	Identifikuju se akteri koji interaguju sa sistemom i specificiraju se slučajevi korišćenja u cilju modelovanja zahteva.
Analiza i projektovanje	Pravi se projekat specificiranjem arhitekturnih modela, modela komponenti, modela objekata i sekvencnih modela.
Implementacija	Vrši se implementacija komponenti i podsistema. Automatsko generisanje koda na osnovu projektnih modela može ubrzati ovaj proces.

51

Statički tokovi poslova u RUP-u

Tok poslova	Opis
Testiranje	Testiranje je iterativni proces koji se sprovodi paralelno sa implementacijom. Testiranje sistema se sprovodi nakon završetka implementacije.
Raspoređivanje	Pravi se izdanje (release) proizvoda i vrši se distribucija i instaliranje kod korisnika.
Konfiguracija i upravljanje izmenama	Pomoćni tok se bavi upravljanjem izmenama nad sistemom.
Upravljanje projektom	Pomoćni tok koji se bavi upravljanjem procesom razvoja sistema.
Okruženje	Ovaj tok se bavi obezbeđivanjem odgovarajućih softverskih alata timu koji je zadužen za razvoj softvera.

52

Dobra praksa u RUP-u

- ✧ Iterativni razvoj softvera
 - Planiranje i isporuka inkremenata na osnovu prioriteta naručioca.
- ✧ Upravljanje zahtevima
 - Eksplicitno dokumentovanje korisničkih zahteva i praćenje izmena u njima.
- ✧ Korišćenje komponentno zasnovanih arhitekture
 - Organizacija arhitekture sistema kao skupa gotovih komponenti.

53

Dobra praksa u RUP-u

- ✧ Vizuelno modeluj softver
 - Korišćenje grafičkih UML modela za predstavljanje statičkih i dinamičkih pogleda na softver.
- ✧ Provera kvaliteta softvera
 - Osigurati da softver poštuje standarde kvaliteta organizacije.
- ✧ Kontrolisano menjanje softvera
 - Kontrolisano menjanje softvera korišćenjem sistema i alata za upravljanje izmenama.

54

Rezime

- ✧ Procesi treba da uključe aktivnosti koje se bave izmenama. U ovom pogledu moguće je uključiti prototipovanje da bi se izbegle loše odluke u pogledu zahteva i projektovanja.
- ✧ Proces je moguće strukturirati za iterativni razvoj i isporuku, tako da izmene mogu raditi bez ometanja sistema kao celine.
- ✧ RUP je moderni proces opšteg tipa koji karakteriše organizacija u faze (započinjanje, razrada, izrada, okončanje) i aktivnosti (zahtevi, analiza i projektovanje, itd.) koje su nezavisne od faza.

55