**Softversko inzinjerstvo**

**I kolokvijum**

**Napomena: Ovo je podsetnik za ucenje, ne ocekuj da ovo priucis I dobijes 10 ;) #11**

**Darko Stevanović**

1. *1. Šta je softver i čime se bavi softversko inžinjerstvo?*

**Softver** je: // 3

* 1. Racunarski program
  2. Pridruzena dokumentacija
  3. Konfiguracioni podaci

**Softversko inžinjerstvo** je inžinjerska disciplina koja se bavi (*TAM*): // 3

* 1. Teorijom
  2. Metodama ==========> za profesionalni razvoj softvera
  3. Alatima

Bavi se i efikasnim razvojem sw i obuhvata sve aspekte proizvodnje sw.

**SW sistem** se sastoji od:

1. Programi
2. Konf.fajlovi

+++

1. Sistemska dokumentacija
2. Korisnička dokumentacija

+++

1. Veb sajtova

Tipovi **SW proizvoda**: // 2

* 1. Generički - za sve
  2. Ugovorni - za posebne potrebe

1. *13. Po čemu se razlikuje sw od informatike (nauke o računarstvu)?*

Informatika se bavi Teorijom i

Osnovama računarstva.

SWE se bavi

praktičnom stranom razvoja i

isporuke korisnog sw.

Teorija nauke o računarstvu je trenutno dobro razvijena i obezbeđuje solidnu osnovu za swe.

1. *18. Po čemu se SWE razlikuje od ostalih vrsta inžinjerstva?*

SWE vs Sistemski inzinjering

Sistemski inžinjering uključuje: hardverski inžinjering

Softverski inžinjering

Procesni inžinjering

Sistemski inžinjeri su uključeni u: Aktivnosti specificiranja

Arhitekturnog projektovanja

Integracija

Isporuke sistema

Sistemski stariji od SWE.

1. *5. Navesti i opsiati atribute dobrog softvera // PUSE*

Skripta.

1. *22. Šta su nepoželjne karakteristike*

Sigurnost i bezbednost

1. *6. Nacrtati dijagram koji iustruje proces sistemskog inžinjerstva*

Skripta.

1. *14. Grafički ilustrovati i objasniti proces nabavke sistema.*

Skripta.

Specifikacija I arhitekturno projektovanje sw => Nabavka sistema

COTS - Commercial of the Shelf, komercijalno dostupne komponente => Modifikacija zahteva

Period rada na ugovoru

1. *7. Šta je sw proces? Navesti i kratko opisati zajedničke aktivnosti za različite sw procese?*

Sw proces je Struktuiran

Skup ==========> *SSA*

Aktivnosti

Neophodan za razvoj sw sistema.

Zajedničke aktivnosti: // 4

7.1) Specifikacija - šta radi sistem

7.2) Projektovanje i implementacija - definisanje sistema i njegova implementacija

7.3) Validacija - provera

7.4) Evolucija - promena sw kao odgovor na promenu zahteva

1. *8. Razvoj sw po modelu vodopada (dijagram, prednosti i problemi)*

*Skripta.*

1. *2. Inkrementalni razvoj sw (dijagram, prednosti, problemi)*

**Prednosti**: // 4

2.1) Smanjena cena izmene u toku razvoja

2.2) Olakšano dobijanje povratne informacije od korisnika u toku razvoja

2.3) Brža isporuka korisnicima

2.4) Korisnici vide proizvod ranije u odnosu na Waterfall

**Problemi**: // 3

2.5) Proces nije vidljiv

2.6) Degradacija strukture

Specifikacija *(SRV)*

Razvoj ==========> se preklapaju

Validacija

Vođen: // 2

2.8) planom - procesi kod kojih se sve aktivnosti unapred planiraju, te se i progres meri na osnovu tog plana

2.9) agilan - planiranje je inkrementalno, pa je lakše promeniti proces nakon promene zahteva naručioca

1. *17. Navedite 4 generička modela sw procesa i opišite 4 zajedničke aktivnosti ovih modela*
2. Model vodopada - plan
3. Inkrementalni razvoj - plan / agilan
4. Razvoj zasnovan na korišćenju gotovih komponenti - plan / agilan
5. Spiralni model - /

Faze: a) specifikacija

b) projektovanje i implementacija

c) validacija

d) evolucija

1. *19. Objasniti metod Prototipa za razvoj softvera*
2. *20. Objasniti SWE zasnovano na komponentama*

Planom vođen, ili agilan

Faze:

1. Analiza komponenti
2. Modifikacija zahteva
3. Projektovanje sistema korišćenjem gotovih komponenti
4. Razvoj i integracija

Tipovi SW komponenti:

1. Veb servisi
2. Kolekcije objekata
3. Nezavisni sw sistemi
4. *12. Navesti i kratko opisati najpoznatije metode inžinjeringa zahteva:*

Inžinjering zahteva predstavlja proces razvoja specifikacije sw.

12.1) Studija izvodljivosti - tehnički i finansijski izvodljiv razvoj sistema

12.2) Prikupljanje i analiza zahteva - šta zainteresovane strane žele

12.3) Specifikacija zahteva - detaljno definisanje zahteva

12.4) Validacija zahteva - provera zahteva

1. *21./4. Navesti i objasniti 6 principa dobre prakse po RUP-u*
2. Iterativni razvoj sw
3. Upravljanje zahtevima

Zahtev klijenta => potrebe I funkcije sistema => detaljna specifikacija funkcionalnih I nefunkcionalnih zahteva => test procedure, projekat, korisnička dokumentacija

1. Komponenta arhitektura sw-a
2. Vizuelno modeliranje
3. Kontinualna verifikacija kvaliteta
4. Upravljanje promenama
5. *8. Navesti I kratko opisati faze u RUP-u. Objasniti odnos faza I iteracija u RUP-u.*

RUP - Rational Unified Process - moderni proces opšteg tipa zasnovan na UML-u

Svaka faza može imati proizvoljan broj iteracija i svaka iteracija (osim početne) treba da rezultira izvršnom verzijom koja se može testirati.

8.1) Početna faza

a) Analiza problema

b) Razumevanje potreba potencijanih korisnika

Rezultat: *Vizija sistema*

8.2) Faza elaboracije

a) Izrada plana projekta

b) Organizacija i ekipni rad

c) Detaljna definicija zahteva

d) Definisanje strukture sistema

Rezultat: *Plan projekta, Use - case specifikacija, Arhitekturni projekat sistema*

8.3) Faza izrade

a) Realizacija sistema

b) Testiranje

Rezultat: *Plan testiranja, Test specifikacija, Detaljni projekat sistema, SW proizvod*

8.4) Faza isporuke

a) Finalizacija sw sistema

b) Alfa (beta) testiranje

c) Izrada korisničke dokumentacije (uputstva)

d) Obuka korisnika

e) Uvođenje sistema kod korisnika

Rezultat: *Test izveštaji, Korisničko uputstvo, Instalacija sistema*

1. *3. Grafički ilustrovati i kratko opisati 4 + 1 model sistema*

3.1) **Logička arhitektura** opisuje:

a) najvažnije klase u sistemu

b) organizacija u pakete I podsisteme

c) paketi i podsistemi u nivoe

Dijagrami klasa

3.2) **Procesna arhitektura** - procesi i niti

Dijagrami klasa

3.3) **Implementacioni model**

Dijagrami komponenti

3.4) **Fizički model** - fizički čvorovi i njihov razmeštaj

Dijagrami razmeštaja

1. *9. Navesti osnovne principe agilnog razvoja sw izražene kroz Agile Manifesto*

Više vrede: Nego:

Pojedinci i interakcije | procesi i alati

Programska podrška | dokumentacija

Saradnja sa klijentima | ugovori

Odgovor na promene | praćenje plana

Principi:

9.1) Brza isporuka

9.3) Česta isporuka

9.4) Ispravan sw

9.2) Mogućnost promene zahteva

9.5) Bliska saradnja

9.6) lice u lice

9.10) promenljive okolnosti

9.7) motivisani pojedinci

9.8) jednostavnost

9.9) samoorganizovani timovi

1. *15. Grafički ilustrovati i opisati SCRUM proces*

Skripta.

Kroz niz sprintova pravimo funcktionalnosti pri čemu one sa najbitnije imaju najviši prioritet.

1. *23. Navesti I kratko opisati faze u SCRUM-u*

Product backlog -> Sprint backlog

-> Zadaci (1 - 16)h

-> Highlevel design

1. *24. Navesti koji sve sastanci po SCRUM-u postoje I kratko ih opsati*
2. Planiranje sprinta
3. Dnevni SCRUM sastanak
   * 1. 3 parametra
     2. 3 pitanja
     3. Informisanje medju jednakima
4. Pregled sprinta
5. Restrospektiva sprinta
6. *16. Objasniti spiralni model procesa specifikacije zahteva*

6 / 20

Proces upravljanja zahtevima se sastoji od medjusobno nerazdvojnih podprocesa, koji se ne mogu zasebno I sekvencijalno odvijati:

Sektori u sprialnom modelu:

1. Postavljanje ciljeva
2. Procena i umanjivanje rizika
3. Razvoj i validacija
4. Planiranje naredne faze spirale
5. *10. Šta predstavlja verifikacija, a šta validacija*

Verifikacija pokazuje da sistem odgovara: Sopstvenoj specifikaciji

Stvarnim potrebama korisnika

Validacija pokazuje da sistem odgovara: Zahtevima naručioca

1. *Šta predstavlja upravljanje zahtevima?*

Osnovi princip po RUP-u.

Zahtev klijenta =>

potrebe I funkcije sistema

=> detaljna specifikacija funkcionalnih I nefunkcionalnih zahteva

=> test procedure, projekat, korisnička, dokumentacija

Napomena: neophodna procedura u slučaju promene zahteva korisnika.