

KI206 - TESTOVI posao

TEST 2

1. Šta je softverski proces?

- **Softverski proces** je skup povezanih aktivnosti koji vodi proizvodnji softvertskog proizvoda. Ove aktivnosti mogu dovesti do razvoja potpuno novog softvera ili do usavršavanja nekog postojećeg.

2. Kako možemo podleiti softverske procese?

- Softverski procesi se mogu podeliti na dva tipa:
 - = Procese vođene planom
 - = Agilne procese

3. Kako možemo definisati procese vođene planom a kako agilne procese?

- **Procesi vođeni planom** su procesi kod kojih su sve aktivnosti unapred planirane I anpredak u razvoju softvera se određuje stepenom ostvarivanja tog plana.
- **Agilni procesi** su procesi kod kojih se planiranje radi postupno (inkrementalno) kako bi se procesi lakše prilagodili promenljivim zahtevima korisnika.

4. Koja tri opšta modela softverskih procesa se najčešće navode?

- Najčešće se navode:
 - = Model vodopada,
 - = Inkrementalni (postepeni) razvoj,
 - = Softversko inženjerstvo zasnovano na višestrukoj upotrebljivosti.

5. Navesti prednosti i nedostatke razvoja softvera upotrebomkomponenata?

- **Prednosti** su da ovaj model razvoja samnjuje količinu softvera koji treba da bude razvijen i smanjuje troškove i rizike, što vodi ka bržaoj isporuci softvera.
- **Nedostatak** što su potrebni kompromisi zahteva i ovo može voditi ka sistemu koji neće u potpunosti ispuniti stvarne potrebe korisnika, jer se upotrebljavaju unapred definisane komponente. Takođe, gubi se i kontrola nad evolucijeom softvera sa upotrebom nižih verzija komponenata.

6. Navesti četiri glavne faze procesa inženjeringa zahteva?

- **Glavne faze su:**
 - = Studija izvodljivosti,
 - = Izvođenje i analiza zahteva,
 - = Specifikacija zahteva
 - = Validacija zahteva.

7. Zbog čega je inženjering zahteva jako bitna faza u procesu razvoja softvera?

- Zato što greške u ovoj fazi neizbežno vode ka kasnijim problemima u proejktovanju i implementaciji sistema.

8. Kako razvijati softver u uslovima stalnih promena?

- U uslovima stalnih promena, primanjuju se dva pristupa u razvoju:
 - = Izrada prototipa softvera, radi provere zahteva.
 - = Inkrementalni razvoj softvera.

9. Šta je inkrementalna isporuka softvera? Navesti jedan primer upotrebe inkrementalne isporuke softvera?

- **Inkrementalna isporuka** softvera je isporuka u kojoj se naručiocu softvera isporučuju inkrementi sistema radi kometarisanja i eksperimentisanja. Primer je Android aplikacija (seti se Theme.io-a).

10. Objasnite zašto je inkrementalni razvoj najefikasniji pristup za razvoju poslovnih softverskih sistema?

- Zato što omogućava izbegavanje prevremenih i čestih promena i daje toleranciju promena. A time se izbegava prerano opredeljivanje da se nešto u sistemu menja, dok se to ne proveri na inkreментu. Takođe, umesto da promene obuhvate ceo sistem, one se odnose samo na inkrement što je znatno jeftinije.

11. Navesti primer upotrebe spiralnog modela razvoja?

- Primer razvoja sistema za centralizaciju dokumenata na teritoriji jedne države.

12. Objasnite faze RUP-a?

- RUP model se bavi analizom rizika i podržava razvoj koji primenjuje slučajeva korišćenja. **Faze** su:

- 1) **Početak** – cilj ovve faze je postavljanje poslovnog scenarija sistema t.j. određivanje šta sistem treba da radi. Utvrđuju se spoljni akteri (ljudi i sistemi) koji su u interakciji sa sistemom.
- 2) **Razrađivanje** – (elaboracija) cilj je razvoj razumevanja domena problema, postavljanje arhitektonskog okvira sistema, razvoj plana projekta i utvrđivanje ključnih rizika projekta.
- 3) **Konstrukcija** – obuhvata projektovanje sistema, programiranje i testiranje.
- 4) **Tranzicija** – je konačna faza RUP-a koja se bavi prenosom sistema iz razvojnog okruženja u korisničko okruženje i stavlja ga u rad u stvarnom okruženju.

13. Šta je inkrementalna isporuka softvera? ***

- Pristup razvoju softvera koji omogućava da se kupcima isporučuju razvijeni delovi sistema, koje onda korisnik testira, ali ne pušta u operativnu upotrebu.

14. Kada inkrementalni pristup nije dobro rešenje? (Označite tri odgovora) ***

- Nije dobro rešenje u slučaju:
 - = Razvoja kritičnih sistema,
 - = Razvoja ugrađenih sistema koji zavise od hardvera,
 - = Kada se sistem razvija na više lokacija.

15. Koji su nedostaci inkrementalnog razvoja softvera? (3) ***

- **Nedostaci** su:
 - = Teško je utvrditi osnovne funkcije koje će biti realizovane od strane svih inkremenata.
 - = Korisnici često nisu voljni da eksperimentišu sa nekompletnim novim sistemom.
 - = Sve do poslednjeg inkrementa ne postoji kompletna specifikacija sistema, što zahteva nove forme ugovora koje često ne odgovaraju nekim organizacijama.

16. Šta je Beta testiranje? ***

- **Beta testiranje** je testiranje koje se vrši kod grupe odabranih korisnika sistema, tj. u njihovom radnom okruženju.

17. Koja je razlika između faza i radnih tokova RUP-a? ***

- **Faze** su dinamičke i imaju svoje ciljeve. **Radni tokovi** su statički i predstavljaju tehničke aktivnosti koje nisu pvezane sa pojedinačnom fazom, već se mogu upotrebiti za vreme celog razvoja radi ostvarivanja ciljeva svake faze.

18. Šta je Alfa testiranje? ***

- Testiranje koje vrši razvojni tim, sam ili sa naručiocem softvera.

19. Šta od navedenog NE spada u četiri osnovne faze razvoja softvera po RUP-u? (Označite 2 odgovora) ***

- Programiranje
- Integracija

20. Zašto je inkrementalni razvoj najefektivniji pristup u razvoju poslovnih softverskih sistema? ***

- Kod poslovnih sistema se često traže promene u softveru tokom njegovog razvoja. U tom slučaju, inkrementalni razvoj je jeftiniji i lakši za rad.

21. Kako se vrše testiranja u slučaju planskog razvoja softvera? ***

- Testiranje se vrše u skladu sa planovima testiranja, koje priprema nezavistan tim na osnovu specifikacije i projekta sistema.

22. Koje su dve osnovne vrste softverskih procesa? ***

- Softverski procesi se mogu podeliti na:
 - = Procese vođene planom
 - = Agilne procese

TEST 3

1. Koja tehnika prikupljanja zahteva je po vašem mišljenju najefikasnija?

- Najefikasnija je Tehnika **Intervijua**. Zato što se uz pomoć ove tehnike prikupi najveći broj zahteva i to direktno od aktera u sistemu. Dobijeni odgovori daju priliku za postavljanje dodatnih pitanja, čime se utvrđuju i pribavljaju nove informacije. Dobri su za razumevanje kako će se sistem ponašati u realnim scenarijima upotrebe i šta sve korisnici očekuju od njega.

2.

TEST 4

1. Koja arhitektura je po vašem mišljenju najzastupljenija u web sistemima?

- Servisno-orjentisana arhitektura.

2.

TEST 5

1. Koja je razlika između dijagrama komponenata i dijagrama klasa? Šta se može videti na dijagramu komponenti a nije prikazano kroz dijagram klasa sistema?

- **Dijagram komponenata**, predstavlja jedan fizički pogled na sistem. Njegova svrha je da pokaže zavisnost između softvera i drugih softverskih komponenata u sistemu.

- **Dijagram klasa**, prikazuje skup klasa, interfejsa i njihovih relacija, tj. opisuje strukturu sistema.

- **Razlika** je u tome što Dijagram klasa prikazuje logičku strukturu apstrakcija (tj. softv. kompon.) a Dijagram komponenata prikazuje fizičku organizaciju i zavisnosti između skupa komponenata. Dijagram komponenti omogućava prikaz fizičke komponente koja već postoji u računaru (za razliku od klasa koje su samo logičke apstrakcije).

2.

TEST 6

1. Objasniti preslikavanje sekvencijalnog modela u dijagram klasa. Na osnovu kog UML modela je moguće generisati programski kod?

- Nakon kreiranja sekvencijalnog dijagrama, identifikovani objekti postaju klase a poruke postaju metode u dijagramu klasa. Na osnovu klasnog dijagrama koji je prethodno nastao od sekvencijalnog dijagrama vrši se generisanje Java programskog koda (tj. Java klasa).

2.

TEST 7

1. Da li testiranje jedinice ubrzava ili usporava proces testiranja softvera? Šta možemo saznati iz dobijenih rezultata testiranja jedinice?

- Iz dobijenih rezultata (jediničnih testova) možemo utvrditi da određene metode izračunavaju tačno ono što je korisnik tražio. Tako da sa većom pouzdanošću možemo reći da sistem radi ono što treba da radi. Testiranje jedinice usporava proces testiranja pa i ukupno razvoja softvera.

2.

TEST 8

1. Šta podrazumeva Evolucija softvera? Ko predlaže promene koje iniciraju evoluciju softvera?

- **Evolucija softvera** je stalna promena softvera posle njegovog inicijalnog razvoja i isporuke naručiocu ili tržištu. Kompanija koja je razvila softver najčešće predaje odgovornost za dalji razvoj i modifikacije softvera kompaniji koja ga je naručila i koja mora da ima odgovarajući kadar za takav razvoj. Takođe, postoji mogućnost i da kompanija angažuje neku drugu organizaciju da podržava evolutivan razvoj njenog softvera.

2.

TEST 9

1. U kom slučaju je potrebno odabrati agilni razvoj pre nego planom vođen razvoj softvera? Navedi konkretan primer.

- **Agilne metode** razvoja softvera se koriste:

- = Kada je potrebno da primenimo inkrementalnu isporuku softvera kupcu i kada od kušca očekujemo brz odgovor.
- = Kada se sistem razvija sa malim timom koji ima članove koji mogu međusobno neformalno da komuniciraju.
- = Kada raspolažemo dobrim alatima za održavanje verzija projektnog rešenja softvera.
- = Kada se u toku procesa razvoja softvera upotrebljava neformalno znanje, sa glavnim fokusom na znanje programera.

2. Šta je to programiranje u parovima? ***

- Dva programera rade za istom radnom jedinicom.

3. Da li kulturološki i istorijski razlozi mogu da utiču na primenu agilnih metodologija? ***

- Da

4. Kako testovi prate novu funkcionalnost koju donosi svaka nova verzija softvera? ***

- Pokreće se izvršavanje i novih i starih testova.

5. Kod agilnih metoda procesi pripreme specifikacije, projektovanja i implementacije su jsano definisani i ne preklapaju se? ***

- Ne

6. Kakav je odnos produktivnosti i kvaliteta rada u paru? ***

- Manja produktivnost ali veći kvalitet koda.

7. Šta je to hibridni pristup u razvoju softvera? ***

- Kombinovanje agilnog i planski vođenog pristupa.

8. Uključivanje svih aktera u razvoj sistema predstavlja problem kod velikih sistema? ***

- Da

9. Za uspešnu primenu agilnih metoda neophodno je da članovi tima imaju vrlo dobre veštine? ***

- Da

10. Koje su perspektive primene agilnih metoda kod razvoja velikih sistema? ***

- Scaling up i Scaling out

11. Kada se definiše test softvera kod ekstremnog programiranja? ***

- Pre pisanja koda.