****

**Predmet SE201 UVOD U SOFTVERSKO INŽENJERSTVO**

**ISPITNA PITANJA**

1. **Lekcija 1: UVOD U SOFTVERSKO INŽENJERSTVO**
   1. Šta je Softversko inženjerstvo? Zašto je ono važno? Šta je razlika softverskog inženjerstva i računarskih nauka?
   2. Šta je softverski proizvod? Koje dve vrste softverskih proizvoda postoje?
   3. Koji su atributi dobrog softvera?
   4. Šta je softverski proces? Koje su četiri osnovne aktivnosti softverskog procesa?
   5. Koji su ključni izazovi softverskog inženjerstva?
   6. Koji tipovi softverskih aplikacija postoje?
   7. Koje su osnovne postavke softverskog inženjerstva koje su primenljive za sve tipove softverskih sistema?
   8. Koji su troškovi izraženi u softverskom inženjerstvu?
   9. Do kojih promena je došlo u Softverskom inženjerstvu zbog uticaja Interneta?
   10. Zašto dolazi do grešaka u radu softvera?
   11. Koji su standardi prihvatljivog ponašanja softverskih inženjera?
   12. Šta su principi ponašanja? Koji su osam principa ponašanja softver inženjera?
2. **Lekcija 2: SOFTVERSKI PROCESI**
   1. Šta je softverski proces? Koje su osnovne aktivnosti softverskog procesa?
   2. Navedita dve osnovne vrste softverskih procesa i ukratko ih opišite.
   3. Šta je model softverskog procesa? Navedite tri opšta metoda softverskih procesa. Objasnite svaki od njih. Koje su međusobne razlike?
   4. Detaljno opišite model vodopada za razvoj softvera. Šta je proces projektovanja sistema? Šta je projektovanje softvera? Šta je implementacija softvera? Šta je testiranje jedinica? Šta je operativni rad i održavanje? Kada se koristi model vodopada?
   5. Šta je inkrementalni razvoj softvera? Zašto se koristi? Šta je inkrement? Koje su prednosti inkrementalnog razvoja u odnosu na model vodopada? Kako se primenjuje inkrementalni razvoj softvera? Koji su problemi u primeni?
   6. Zašto je inkrementalni razvoj najefektniji pristup u razvoju poslovnih softverskih sistema? Zašto je ovaj model manje pogodan za razvoj sistema u realnom vremenu?
   7. Predložite model opšteg softverskog procesa za razvoj sledećih sistema:
      1. Rezervacija avionskih karata i hotela
      2. Sistema za prodaju proizvoda preko veba
      3. Upravljanje sistemom podzemne železnice u gradu
   8. Šta je softversko inženjerstvo za višestruku upotrebu? Šta su softverske komponente? Koje tipove komponenata imamo?Opišite proces razvoja softvera baziranog na ponovno upotrebljivim komponentama. Opišite ukratko svaku aktivnost?
   9. Koje su prednosti korišćenja razvoja softvera baziranog na upotrebi komponenata? Koji su nedostaci korišćenja?
   10. Opišite glavne aktivnosti procesa razvoja softvera i dobijene rezultate ovih aktivnosti. Upotrebom dijagrama, pokažite relacije između izlaza iz ovih aktivnosti.
   11. Šta je specifikacija softvera? Opišite proces inženjeringa zahteva. Koje su četiri glavne faze procesa inženjeringa zahteva? Opišite svaku od ovih faza.
   12. Čime se bavi aktivnost softverskog procesa koja se bavi projektovanjem i implementacijom softvera. Nacrtajte model procesa projektovanja i implementacije softvera. Objasnite aktivnosti navedene u modelu.
   13. Šta je validacija softvera? Nacrtajte dijagram procesa testiranja. Koje su tri osnovne faze testiranja? Opišite svaku od ovih faza.
   14. Kako se vrši testiranje u slučaju primene ekstremnog programiranja. Kako se vrši testiranja u slučaju planskog razvoja softvera? Nacrtajte faze testiranja u okviru modela procesa planskog razvoja softvera. Šta je alfa testiranje? Šta je beta testiranje?
   15. Šta je evolucija softvera? Zašto dolazi do promena zahteva? Koka se mogu smanjiti troškovi evolucije softvera?
   16. Zašto se koristi prototip softvera? Kada se koristi prototip? Nacrtajte model procesa razvoja prototipa. Objasnite faze razvoja prototipa datih u modelu procesa razvoja prototipa. Šta prototip treba da sadrži, a šta ne mora da sadrži? U kom obliku se predstavljaju prototipovi?
   17. Šta je inkrementalna isporuka softvera? Nacrtajte model inkrementalnog razvoja softvera. Koji su nedostaci inkrementalnog razvoja softvera? Kada inkrementalni pristup nije dobro rešenje?
   18. Šta je spiralni model razvoja softvera? Nacrtajte model spiralnog modela razvoja? Opišite sve sekcije u spirali tazvoja. Koji su rizici primene spiralnog modela razvoja softvera?
   19. Objasni zašto je spiralni model prilagodljiv model koji može da podrži neophodne promene i zašto toleriše aktivnosti promena? Objasnite zašto se u praksi često koristi?
   20. Opišite glavne aktivnosti procesa razvoja softvera i dobijene rezultate ovih aktivnosti. Upotrebom dijagrama, pokažite relacije između izlaza iz ovih aktivnosti.
   21. Šta je RUP? Koje perspektive prikaza softverskog procesa podržava. Koje su četiri osnovne faze razvoja softvera po RUP-u? Objasnite svaku od ovih faza. Opišite radne tokove RUP-a. Za koje slučajeve je RUP pogodan, a za koje nije? Koja je razlika između faza i radnih tokova RUP-a?
3. **Lekcije 3 : MODELIRANJE STRUKTURE SISTEMA PRIMENOM UML** 
   1. Šta je modelovanje sistema? Šta je model? Zašto služe modeli? Koje vrsti modela se koriste u softverskom inženjerstvu? Zašto? Koje se perspektive modela koriste? Navedite i ukratko opišite glavne UML dijagrame. Koja su tri načina upotrebe grafičkih simbola pri modelovanju?
   2. Šta je konteksni model sistema?Šta on uzima u obzir? Dajte jedan primer. Koji su najčešći UML modeli interakcije? Za šta se koriste i kako se koriste UML slučajevi korišćenja? Opišite slučaj korišćenja sistema po Vašem izboru.
   3. Šta su sekvencijalni dijagrami? Dajte jedan primer. Kako se određuje stepen apstrakcije sekvencijalnog dijagrama.
   4. Strukturni modeli – navedite UML dijagram strukture. Zašto se oni koriste? Objasnite dijagram klasa. Dajte primer. Šta je generalizacija? Kako se obeležava u dijagramu klasa? Zašto se generalizacija koristi? D ajte primer generalizacije u vidu UML dijagrama klasa. Šta je agregacija? Kako se obeležava u UML?
4. **Lekcija 4: Modeliranje ponašanja sistema u UML**
   1. Šta su modeli ponašanja u UML? Opišite modele koji su vođeni podacima (dijagram aktivnosti, sekvencijalni dijagram…). Šta je modelovanje vođeno događajima? Opišite UML dijagrame stanja. Dajte primer. Koji su problemi kod modelovanja stanja?
   2. Šta je modelima vođeno inženjerstvo? Šta je modelima vođena arhitektura sistema? Zašto se koristi modelima vođeno inženjerstvo? Koji su argumenti protiv primene modelima vođenog inženjerstva? Navedite tri različita apstraktna modela i definišite ih. Prikažite dijagrame transformacije u slučaju modelom vođenim transformacijama. Šta je to izvršni UML? Šta je OCL?
   3. Opišite odgovarajućim UML modelima sve faze razvoja softverskog sistema za sledeće primere:
      1. Razvoj softverskog sistema za izradu rasporeda časova, ili
      2. Razvoj softverskog sistema za prikaz i naručivanje jela u restoranu (svaki sto ima touch screen računar sa prikazom jelovnika i izborom jela i pića.
      3. Razvoj sistema za odobrenje putnih naloga i za izveštavanje i obračun po obavljenom putu.
   4. Definišite UML slučajeve korišćenja, prateće sekvencijalne dijagrame i scenarije za softverski sistem dat sa sledećim opisom: *Sistem upravlja procesom realizacije službenih putovanja zaposlenih u firmi ABC. Sistem treba da obezbedi odobrenje za službeni put, izvore finansiranja puta, rezervacije karata, hotela i dr., izveštaj lica koje je išlo na službeni put, isplatu dnevnica, plaćanje računa za karte, hotel i dr., izradu konačnog finansijskog obračuna, i arhiviranje celokupne dokumentacije vezane za službeni put.*
   5. Nacrtajte sledeće UML dijagrame: a) Slučajevi upotrebe, b) Sekvencijalni c)Dijagrama klasa, d) Dijagram stanja za slučaj razvoja softverskog sistema koji treba da omogući naručivanje pice (ima ih 4 vrste) preko odgovarajućeg veb sajta, sa isporukom kući.
   6. Isključivo primenom detaljnog sekvencijalnog dijagrama, odredite atribute i metode neophodnih klasa za sledeći slučaj korišćenja sistema za rezervaciju hotelskog smeštaja: Korisnik želi da vidi u izabranom hotelu sve vrste hotelskih soba, njihovu cenu prenoćišta sa doručkom i raspoloživot u određenom terminu.
5. **Lekcija 5: INŽENJERSTVO ZAHTEVA**
   1. Šta su zahtevi za softver? Čime se bavi inženjerstvo zahteva? Kakvih vrsta zahteva ima? Objasnite ih detaljnije. Šta su funkcionalni zahtevi? Dajte primere. Šta su nefunkcionalni zahtevi. Dajte primere. Kako se pišu zahtevi? Šta je kompletnost, a šta konsistentnost zahteva?
   2. Šta su funkcionalni zahtevi? Koji su specifični problemi definisanja funkcionalnih zahteva kod velikih sistema?
   3. Šta su nefunkcionalni zahtevi? Zašto su oni važni? Navedite osnovne grupe nefunkcionalnih zahteva i opišite svaki. Koji se problemi javljaju kod utvrđivanja nefunkcionalnih zahteva? Koja je metrika nefunkcionalnih zahteva? Koje su teškoće u njenoj primeni?
   4. Šta je dokument sa zahtevima? Kako se ovaj dokument koristi kod različitih pristupa razvoju softvera? Ko su korisnici tog dokumenta i zašto? Kako se određuje stepen detaljnosti dokumenta sa specifikacijom zahteva? Koja je struktura dokumenta sa specifikacijom zahteva.
   5. Kako se specificiraju zahtevi? Kako treba da budu napisani? Da li postoje veza između arhitekture softverskog sistema i sistemskih zahteva? Koje su mogući načini izražavanja sistemskih zahteva (notacija)? Koje su preporuke za specificiranje zahteva običnim, govornim jezikom?
   6. Kako se specificiraju zahtevi primenom strukturnog jezika? Da li se može koristiti i programski jezik? Kako se struktuiraju sistemski zahtevi? Dajte jedan primer. Dajte tipičan sadržaj standardizovanog formulara za specifikaciju zahteva. Zašto se primenjuje i strukturiran način definisanja zahteva?
   7. Koje osnovne aktivnosti proces inženjeringa zahteva? Dajte spiralni pogled na proces inženjerstva zahteva i objasnite ga. Opišite metod slučajeva upotrebe u UML za definisanje zahteva.
   8. Zašto softverski inženjeri treba da razgovaraju sa korisnicima ili kupcima softvera. Šta su akteri softverskog sistema? Koji su ostali akteri sistema, sa kojima treba razgovarati u procesu prikupljanja zahteva? Nacrtajte model prikupljanja i analize sistema, i objasnite ga, tj. njegove aktivnosti. Koje se poteškoće javljaju pri radu sa akterima sistema? Kako olakšati prikupljanje zahteva? Pored razgovora sa akterima, koji su drugi izvori informacija značajnih za definisanje zahteva?
   9. Zašto služe intervjui u procesu prikupljanja zahteva, i koje dve vrste intervjua postoje? Za šta su intervjui dobri, a za šta nisu? Koje karakteristike treba da ima lice koje obavlja intervju (intrevjuer)?
   10. Šta su scenariji i čemu služe u procesu prikupljanja i specificiranja zahteva? Koji su osnovni elementi jednog scenarija? U kojim formama se može opisati scenario? Šta je UML slučaj korišćenja? Kako se on opisuje? Dajte jedan primer. Kako se scenariji koriste u vezi sa UML slučajevima upotrebe? Da li se sekvencijalni dijagram može koristiti za grafičko predstavljanje scenarija slučaja upotrebe? Dajte jedan primer.
   11. Šta je etnografija? Kako se ona koristi za otkrivanje zahteva? Kako se etnografija može povezati sa korišćenjem prototipova softverskih sistema? Kada je posebno preporučljivo koristiti etnografiju?
   12. Šta je validacija zahteva? Kako validacija utiče na troškove razvoja sistema? Kako se vrši validacija (provera zahteva) dokumenta koji specificira zahteve? Koje se tehnike validacije zahteva primenjuju?
   13. Čime se bavi upravljanje zahtevima? Koji je glavni izazov? Zašto dolazi do promena zahteva? Šta je evolucija zahteva? Zašto se javljaju novi zahtevi i posle puštanja sistema? Šta je upravljanje zahtevima? Koje odluke treba doneti da bi se planski upravljalo zahtevima? Kakva se automatizovana podrška može dati aktivnostima upravljanja zahtevima? Kako upravljati promenama zahteva? U čemu je prednost primene formalizovanog procesa upravljanja zahtevima? Koje su faze procesa upravljanja promenama zahteva? Do kakvog se problema može doći ubacivanjem stalno novih zahteva? Koja je specifičnost promene zahteva kod agilnog razvoja softvera?
6. **Lekcija 6: Analiza i modelovanje softverskog sistema**
   1. Šta je model softverskog sistema? Kada se koriste modeli softverskog sistema? Navedite najčešće korišćene UML dijagrame i navedite njihovu svrhu.
   2. Šta obezbeđuje model analize sistema? Koja se polazna dokumenhta koriste za analizu?
   3. Koje su tri klase analize sistema? Objasni njihovu ulogu.
   4. Šta predstavlja klasa Entity? Kako se analizom nalaze klase Entity? Opiši iterativni postupak određivanja klase Entity.
   5. Šta je klasa tipa Boundry? Kako se utvrđuje klasa tipa Boundary analizom slučajeva korišćenja? Navedite primer.
   6. Šta klasa tipa Control? Kako se utvrđuje klasa tipa Control analizom slučajeva korišćenja? Navedite primer.
   7. Dajte primer jednog slučaja korišćenja opisanof klasama analize sistema.
   8. Šta je inženjerstvo sa modelima? Šta je OCL i čemu on služi? Dajte jedan primer priimene OCL.
   9. Šta je VOPC dijagram (View Of Participating Classes)? Dajte primer jednog VOPC dijagrama.
7. **Lekcija 7: PROJEKTOVANJE ARHITEKTURE SOFTVERSKOG SISTEMA**
   1. Šta je projekat (dizajn) arhitekture? Šta je model arhitekture sistema? Kada se definiše arhitektura sistema? Zašto je arhitektura važna? Šta omogućava arhitektura sistema? Kako se pravi model arhitektura? Dajte jedan primer. Za koje svrhe se koristi model arhitekture sistema?
   2. Objasni zašto je potrebno da se projektuje arhitektura sistema pre nego što je završena specifikacija zahteva
   3. Objasni proces projektovanja arhitekture. Na koja pitanja treba naći odgovore u toku procesa projektovanja arhitekture? Šta su arhitektonski šabloni? Šta oni sadrže? O kojim nefunkcionalnim zahtevima se mora voditi računa prilikom izboru arhitektonskog šablona? Kako se ocenjuje arhitektura sistema?
   4. Objasni zašto se mogu javite konflikte situacije pri projektovanju arhitekture u kojima su zahtevi za raspoloživost i bezbednost najvažniji nefunkcionalni zahtevi. Dajte jedan konkretan primer.
   5. Šta je arhitektonski pogled? Koji su fundamentalni (osnovni) arhitektonski pogledi? Objasnite ukratko svaki od njih. Pored fundamentalnih pogleda, koristi se i koncepcijski pogled. Šta je to? Da li je detaljna specifikacija arhitekture potrebna?
   6. Šta je arhitektonski šablon? Objasni MVC šablon (model-kontroler-pogled). Prikaži arhitekturu veb aplikacije urađenu prema MVC šablonu.
   7. Objasnite zašto upotrebljavate nekoliko šablona arhitekture kada projektujete arhitekturi velikog sistema. Pored onoga što je rečenu u tekstu predavanja o šablonima, koje su dodatne informacije koje mislite da su od koristi pri projektovanju velikih sistema?
   8. Šta je arhitektonski šablon? Šta je slojevita arhitektura? Dajte opis šablona slojevite arhitekture. Prikažite grafički primer slojevite arhitekturu softverskog sistema. Šta softverski slojevi omogućavaju? Dajte primer slojevite arhitekture nekog sistema.
   9. Šta je arhitektonski šablon? Šta je arhitektura sa skladištem podataka? Ko koristi ovu arhitekturu? Dajte opis šablona arhitekture sa zajedničkim skladištem podataka. Dajte grafički prikaz jedne arhitekture sa zajedničkim skladištem podataka. Koja svojstva ima ova arhitektura? Šta su pasivna, a šta aktivna skladišta podartaka?
   10. Šta je arhitektonski šablon? Šta je model cevi i filtra? Šta je „cev“, a šta je „filter“? Opišite šablon arhitekture sa cevima i filtrima. Dajte prime i grafički prikažite primer jedne arhitekture sa cevima i filtrima. Za koje slučajeve je ova arhitektura pogodna, a za koje nije?
   11. Šta su aplikacioni sistemi-aplikacije? Šta su arhitekture aplikacija, šta sadrže? Kako odlučujete da li kupiti ili napravite softver? Kako koristiti modele arhitekture standardnih softverskih proizvoda?
   12. Šta su sistemi za transakcionu obradu? Koja je struktura sistema za transakcionu obradu? Prikažite arhitekturu softvera ATM sistema (bankomata).
   13. Šta su informacioni sistemi? Prikažite opšti model informacionih sistema. Dajte primer nekog informacionog sistema. Kako se upravlja podacima kod informacionih sistema?
   14. Šta su sistemi za obradu jezika? Kako izgleda arhitektura sistema za obradu jezika? Kakva je arhitektura kompajlera, tj. prevodioca programskih jezika? Dajte prikaz arhitekture sistema za prevođenje programskog jezika koja sadrži skladište
   15. Upotrebom osnovnog modela višeslojne arhitektura nekog informacionog sistema, predloži komponente koje bi mogle da bude deo informacionog sistema koji dozvoljava korisnicima da vide informaciju o dolazećim i dolazećim letovima na jednom aerodromu.
   16. Definišite arhitekturu sistema definisan sledećim opis om: *Sistem upravlja procesom realizacije službenih putovanja zaposlenih u firmi ABC. Sistem treba da obezbedi odobrenje za službeni put, izvore finansiranja puta, rezervacije karata, hotela i dr., izveštaj lica koje je išlo na službeni put, isplatu dnevnica, plaćanje računa za karte, hotel i dr., izradu konačnog finansijskog obračuna, i arhiviranje celokupne dokumentacije vezane za službeni put.* (povezan zadatak: 5.8)
8. **Lekcija 8: PROJEKTOVANJE SOFTVERA I ŠABLONI KREIRANJA OBJEKATA**
   1. Šta sU objektno-orijentisani sistemi? Šta je stanje objekta? Koje su osnovne aktivnosti procesa projektovanja? Zašto je važno razumevanje spoljnjeg okruženja u kome treba da radi softverski sistem? Šta je model konteksta sistema? Dajte neki primer modela konteksta sistema.
   2. Šta je model interakcije? Šta je UML dijagram korišćenja (UML use case). Šta je akter? Nacrtajte dijagram korišćenja nekog sistema i objasnite svaki od slučajeva korišćenja koji ste koristili u dijagramu, u vidu formalnog tekstualnog opisa. Šta su scenariji? Navedite jedan scenario za jedan slučaj korišćenja sistema iz vašeg dijagrama.
   3. Na početku projektovanja softverskog sistema, polazite od prethodno definisane arhitekture sistema. Zašto? Koje informacije vam ona daje,a relevantne su za vaš dalji rad na projektovanju sistema. Kako vama dijagrami slučajeva korišćenja i prateće tekstualne informacije, npr. scenariji, pomažu da utvrdite osnovne objekte i operacije koje sistem treba da ima? Dajte primer.
9. **Lekcija 9: ŠABLONI PROJEKTOVANJA STRUKTURE SOFTVERA**
   1. Kako se vrši detaljno projektovanje klasa? Šta je UML model klase? Kako određujete njene metode i atribute? Dajte neki vaš primer. Dajte projektno rešenje klase za sledeće objekte (definišite atribute i operacije/metode): telefon, štampač personalnog računara i bankovni račun.
   2. Šta su modeli projektovanja? Šta je nivo apstrakcije modela projektovanja? Navedite strukturne UML modele. Navedite UML dinamičke modele. Definišite sve metode i atribute klasa koje koristite.
   3. Definišite argumente u zaglavlju svakog metoda. Sve dijagrame, klase i metode uradite za sledeći slučaj korišćenja sistema: *Sistem treba da uključi sistem za grejanje vaše kuće svakog jutra u 6 sati, a da vam uključi radio pored vašeg kreveta u 7 sati.*
10. **Lekcija 10: ŠABONI PROJEKTOVANJA PONAŠANJA SISTEMA**
    1. Šta su UML sekvencijalni dijagrami? Koja je veza između UML dijagrama korišćenja i sekvencijalnih dijagrama? Dajte primer dva povezana sekvencijalna dijagrama. Prvi se odnosi na visok nivo apstrakcije sistema, a drugi – na niski, detaljni nivo. Čemu služi sekvencijalni model?
    2. Šta je UML model stanja? Šta je stanje objekta? Kako model menja stanje? Zašto se koriste modeli stanja? Dajte neki primer model stanja i objasnite navedena stanja. Dajte dijagram stanja lifta.
    3. Šta je interfejs? Koja je uloga interfejsa? Koja je veza interfejsa i klasa? Da li jedna klasa može imati više interfejsa? Da li jedan interfejs može da bude povezan sa više klasa? Šta su zaglavlja(signature operacija/metoda? Šta one sadrže? Šta je povratna vrednost metoda? Šta je semantika interfejsa? Šta su servisi, šta oni rade?
    4. Šta je šablon? Šta je šablon projektovanja? Koje informacije sadrži opis jednog šablona za projektovanje. Opišite šablon „posmatrač“ (engl. Observer). Kada se koristi ovaj šablon? Dajte primer primene ovog šablona.
    5. Koja je korist od primene šablona projektovanja? Koji problemi mogu nastati u primeni šablona? Kako se ovi problemi rešavaju? Kako se nalazite pogodan šablon?
11. **Lekcija 11: IMPLEMENTACIJA SOFTVERA**
    1. Šta implementacija sistema? Da li je to programiranje, ili rad sa drugim aspektima razvoja softvera? Koji? Objasnite svaki od ovih aspekata.
    2. Ponovna upotreba softvera: Šta je to? Koja je korist od ovoga? Koji su nivoi primene ponovljivosti softvera? Objasniti primenu na svakom nivou. Koji su troškovi primene upotrebljenog softvera, odn. ranije razvijenog softvera/softverskog proizvoda?
    3. Upravljanje konfiguracijom softvera: Šta je konfiguracija softvera? Koja je korist od upravljanja konfiguracijom? Koji su izazovi povezani sa konfiguracijama softvera? Koje su tri osnovne aktivnosti upravljanja konfiguracijom? Opišite upravljanje konfiguracijom za svaku od ovih aktivnosti. Navedite neki softverski alat za upravljanje konfiguracijom softvera?
    4. Navedite specifičnosti razvoja sledećih softverskih sistema: a) ugrađeni sistemi, b) posrednički softver (middleware).
    5. Koji se softverski alati koriste u okviru razvojnih platformi? Čemu služi server za razvoj softvera/razvojni server)? Šta su to integrisane razvojne platforme (IDE)? O čemu se mora voditi računa pri primeni razvojnih platformi?
    6. Razvoj softvera sa otvorenim izvornim kodom: Šta je to? Šta je karakteristično za ovakav način razvoja softvera? Ko nosi odgovornost za razvoj ovog softvera? Šta treba da imate u vidu kada odlučujete da li da koristite softver otvorenog koda ili komercijalni softver? Koji je poslovni model primene softvera sa otvorenim izvornim kodom?
    7. Licence za korišćenje softvera otvorenog koda: Ko je vlasnik softvera? Ko ima prava da ga menja i koristi? Opište prava i obaveze koje predviđaju licence namenjene sistemima sa otvorenim kodom: a) GNU opšta javna licenca b) GNU ograničena javna licenca c) Berkli licenca standarde distribucije. Koji mogući problema prate korišćenje softvera sa otvorenim kodom? Koje preporuke bi dali vašem direktoru koji razmišlja da razvoj novog softverskog proizvoda zasnuje na korišćenju komponenti koji su otvorenog koda.
    8. Definišite dijagrame klasa i sekvencijalne dijagrama za softverski sistem dat sledećim opisom: *Sistem upravlja procesom realizacije službenih putovanja zaposlenih u firmi ABC. Sistem treba da obezbedi odobrenje za službeni put, izvore finansiranja puta, rezervacije karata, hotela i dr., izveštaj lica koje je išlo na službeni put, isplatu dnevnica, plaćanje računa za karte, hotel i dr., izradu konačnog finansijskog obračuna, i arhiviranje celokupne dokumentacije vezane za službeni put.* (povezan zadatak: 5.8 i 6.16)
12. **Lekcija 12: TESTIRANJE SOFTVERA**
    1. Šta je testiranje softvera? Koji su ciljevi testiranja? Nacrtajte model testiranja. Šta pokazuje testiranje softvera? Koja je razlika između verifikacije i validacije softvera? Zašto služe verifikacija i validacija?
    2. Koji faktori utiču na nivo poverenja korisnika u softver? Šta je inspekcija i recenzija softvera? Koje su prednosti inspekcije u odnosu na testiranje softvera?
    3. Nacrtajte model tradicionalnog procesa testiranja softvera. Objasnite svaku fazu procesa testiranja. Koja je razlika ručnog i automatskog testiranja? Objasnite zašto testiranje može samo da utvrdi prisustvo grešaka, a ne i njihovo odsustvo.
    4. Šta je razvojno testiranje softvera? Koji nivoi (granulacije) testiranja u razvoju se koriste? Dajte kratak opis svakog nivoa testiranja softvera u fazi razvoja.
    5. Testiranje jedinice: Šta je to? Šta je testiranje klase, šta obuhvata? Kako se testiraju operacije klasa u njihovoj hijerarhiji, tj. da li se ista operacija klase mora da testira i u njnoj pod- klasi ? Kako se biraju slučajevi za testiranje jedinica? Koje se strategije testiranja mogu izabrati? Opišite svaku od njih. Šta je regresiono testiranje? Objasnite kako upotreba automatskih testovi i platforme za atestiranje, kao što je Junit, uprošćavaju regresiono testiranje?
    6. Testiranje komponenata: Šta se testira kod softverskih komponenti? Koji tipovi grešaka se mogu javiti kod testiranja interfejsa? Koje kategorije grešaka postoje kod interfejsa? Koja su opšta pravila za testiranje komponenata?
    7. Testiranje sistema: Šta obuhvata testiranje sistema? Koja je razlika testiranja sistema i testiranje komponenti? Šta se očekuje of testiranja sistema? Šta je testiranje slučajeva korišćenja? Kako odrediti slučajeve korišćenja za testiranje? Koja svojstva sistema se testiraju? Da li se vrši i automatizovani testiranje sistema?
    8. Šta je razvoj vođen testovima? Dajte grafički prikaz procesa testiranja funkcije softvera u fazi razvoja. Kako se vrši automatsko testiranje inkremenata sa JUnit? Koje povoljnosti nudi razvoj softvera koji je vođen testovima? Koji su problemi ručnog testiranja? Koje su prednosti automatskog testiranja? Kada se koristi razvoj softvera vođen testovima? Šta treba da odredi testiranje sistema? Kada se postižu najbolji rezultati primene razvoja vođenog testovima?
    9. Napišite scenario koji bi mogao da se upotrebi za projektovanje testova za stanicu za prikupljanje podataka o vremenskim prilikama.
    10. Testiranje softvera za isporuku: Šta je to? Ko vrši testiranje sistema spremnog za isporuku? Zašto? Šta je cilj testiranja softvera za isporuku. Kako se vrši testiranje sistema za isporuku? Koliko testova je potrebno izvršiti?
    11. Testiranje softvera za isporuku: Šta je scenario testiranja? Šta on mora da obezbedi? Koliko se zahteva proverava jednim scenarijom? Kako se proveravaju performanse? Šta je operativni profil? Šta je „stres testiranje“? Zašto je ono važno i šta pomaže?
    12. Korisničko testiranje: Šta je korisničko testiranje? Zašto je ono potrebno? Koja je razlika između alfa beta testiranja i testa prihvatanja softvera? Nacrtajte dijagram procesa testiranja prihvatanja softvera od strane naručioca. Kako se vrši test prihvatanja kod primene agilnih metoda razvoja softvera? Koji rizici prate testove prihvatanja?
    13. Koje su prednosti uključenja korisnika u testiranje softvera za isporuku u ranim fazama procesa testiranja? Da li ima i nedostataka u uključenju korisnika u razvoj testova?
    14. Predložite test testiranja za sistem sa sledećim opisom: *Sistem upravlja procesom realizacije službenih putovanja zaposlenih u firmi ABC. Sistem treba da obezbedi odobrenje za službeni put, izvore finansiranja puta, rezervacije karata, hotela i dr., izveštaj lica koje je išlo na službeni put, isplatu dnevnica, plaćanje računa za karte, hotel i dr., izradu konačnog finansijskog obračuna, i arhiviranje celokupne dokumentacije vezane za službeni put.* (povezan zadatak: 5.8, 6.16 i 7.18
13. **Lekcija 13: EVOLUCIJA SOFTVERA**
    1. Šta je evolucija softvera? Kako izgleda spiralni model razvoja i evolucije softvera? Ko održava softver razvijen po narudžbini?
    2. Koja je razlika između evolucije softvera i njegovog servisiranja? Šta pokreće proces evolucije softvera? Koji su razlozi za to? Objasnite ciklični karakter evolucije softvera. Kada se prihvataju predložene promene? Kada se publikuje (objavljuje) novo izdanje softvera? Koje su specifičnosti evolucije softvera kod primene agilnih metoda razvoja softvera?
    3. Nacrtajte proces evolucije softvera i objasnite njegove faze/aktivnosti. Zašto u njemu ima iteracija? Šta bi moglo da se menja, u idealnom slučaju, u softveru, za vreme njegove implementacije? Zbog čega se softver menja i posle početka njegove primene? Koje mogu biti negativne posledice menjanja softvera u toku njegove primene? Koja su dve vrste problema mogu javiti kada različiti timovi razvijaju softver i vrše njegovu evoluciju?
    4. Navedite pet zakonitosti evolucije softvera. Ukratko ih objasnite.
    5. Navedite i ukratko opišite Lemanove zakone evolucije softvera.
    6. Šta je održavanje softvera? Koji tipovi održavanja softvera postoje? Kakvi su troškovi održavanja u odnosu na troškove razvoja softvera i šta izaziva najveće troškove održavanja? Zašto su troškovi održavanja ugrađenih sistema i četiri puta veći nego troškovi razvoja softvera? Kako se troškovi održavanja mogu smanjivati? Koji softver brže stari: razvijen agilnim metodama ili planski razvijen softver? Zašto?
    7. Zašto je skuplja nova funkcionalnost u održavanja nego u razvoju? Koji su razlozi za to? Kako se mogu predviđati troškovi održavanja? Kako oceniti složenost odnosa između sistema i okruženja? Koje su preporuke za smanjivanje troškova održavanja?
    8. Kakva je metrika procesa potrebna da bi se ocenila sposobnost održavanja sistema? Objasnite posebno svaki od načina za ocenu sposobnosti održavanja softverskog sistema.
    9. Reinženjering softvera: Zašto softver stari? Šta je reinženjering softvera? Koja je korist od primene reinženjeringa softvera? Nacrtajte dijagram procesa reinženjeringa starog Objasnite svaku aktivnost tog procesa. softvera. Da li u praksi moramo uvek da sprovedemo sve navedene aktivnosti? Zašto?
    10. Šta je restrukturiranje softvera? Koja je razlika restrukturiranja i reinženjeringa? Koja je specifičnost primene restrukturiranja kod agilnih metoda razvoja softvera? Kada treba primeniti restrukturiranje koda? Šta je i kada se primenjuje restrukturiranje projektnog rešenja?
    11. Šta raditi sa starim softverom? Koje su varijante? Kada koju primeniti?Kako postaviti strategiju evolucije softvera u vašoj firmi? Kakvu klasterizaciju možete primeniti kod starih softverskih sistema? Šta raditi sa svakim od karakterističnih klastera?
    12. Kako utvrditi poslovnu vrednost nekog softverskog sistema? Koje faktore uzimate u obzir? Objasnite svaki od faktora.
    13. Treba da donesete odluku da li da zamenite stari softver sa novim. Jadan od parametara odlučivanja je i ocena okruženja u kome softver sada radi. Koje faktore koristite za ocenu okruženja? Koja tipična pitanja postavljate za svaki od faktora?
    14. Treba da donesete odluku da li da zamenite stari softver sa novim. Jadan od parametara odlučivanja je i ocena kvaliteta softvera, tj. aplikacije koju ste do sada koristili. Koje faktore koristite da bi ocenili kvaliteta aplikacije? Koja tipična pitanja postavljate za svaki od faktora?
    15. Koji su vam podaci potrebni da bi dali ocenu dosadašnjeg softverskog sistema. Ukratko objasnite zašto vam je potreban svaki od navedenih podataka. Koji su ostali uticajni faktori koji dovode do promene softverskog sistema ili do zadržavanja starog softvera.
14. **Lekcija 14: AGILNI RAZVOJ SOFTVERA**
    1. Šta je to agilni metod razvoja? Zašto je došlo da razvoja agilnih metoda razvoja softvera? Koja su svojstva agilnih metoda? Šta je to specifično kod agilnih metoda razvoja softvera? Zašto je brzi razvoj i isporuka novih sistema često važnije za poslovanje nego detaljna funkcionalnost ovih sistema?
    2. Šta je karakteristično za agilne metode razvoja softvera? Z akoje vrednosne stavove sa zalaže tyv. Agilni manifest? Koji su principi agilnih metoda? Šta je to Sprint ciklus?
    3. Kada se primenjuju agilne metode? Koje su poteškoće u primeni agilnih metoda? Kako se mogu primeniti agilne metode u održavanju i koje su prisutne teškoće?
    4. Objasnite kako principi na kojima se zasnivaju agilni metodi ubrzavaju razvoj softvera?
    5. Kada nećete preporučiti primenu agilnog metoda razvoja softverskog sistema i zašto?
    6. Šta je ekstremno programiranje? Nacrtajte i objasnite ciklus ekstremnog programiranja.
    7. Koji principa agilnog programiranja se primenjuju pri ekstremnom programiranju?
    8. Navedite i objasnite principe ekstremnog programiranja. Otkud naziv „ekstremno“ programiranje? Šta je tu „ekstremno“?
    9. Šta su i čemu služe karte sa pričama? Objasnite njihovu primenu.
    10. Šta je problem inkrementalnog programiranja? Kako se on otklanja u slučaju primene ekstremnog programiranja?
    11. Kako se vrši testiranje softvera u uslovima primene ekstremnog programiranja pri razvoju softvera? Kada i kako se definiše test? Dajte primer testa za neki zadatak u razvoju softvera.
    12. Zašto se vrši automatizacija testova kod ekstremnog programiranja? Kako tetsovi prate novu funkcionalnost koju donosi svaka nova verzija softvera? Koji se problemi ovde mogu da jave?
    13. Šta je to programiranje u parovima? Koju povoljnost ta praksa donosi? Kakva je produktivnost takvog rada? Navedite četiri razloga za ubrzavanje produktivnosti programera kada rade u paru za oko 50% u odnosu kada sva programera rade individualno.
    14. Uporedite planom vođeni razvoj i agilni razvoj softvera. Kako se pri oba pristupa realizuju iteracije u procesu razvoja? Šta je to hibridni pristup u razvoju softvera? Kada se on primenjuje?
    15. Kako se može primeniti agilne metode razvoja softvera i u slučaju razvoja velikih softverskih sistema? Koje su specifičnosti razvoja velikih sistema? Koje su perspektive primene agilnih metoda kod razvoja velikih sistema? Koje su poteškoće u primeni kod velikih sistema?
    16. Obično se navodi da je jedan od problema uključivanje korisnika u razvojni tim je što mogu da izgube kontakt sa potrebama svojih kolega, korisnika sistema. Preporučite tri načina za izbegavanje ovog problema i diskutujte prednosti i nedostatke svakog od pristupa.
15. **Lekcija 15: Upravljanje projektovanja razvojem softvera**
    1. Koja su tri glavna ograničenja pri razvojusoftvera? Dajte neki primer.
    2. Šta je cilj menadžera projekta? Navedite njogove odgovornosti.
    3. Koje kategorije rizika postoje pri razvoju softvera? Navedite primere.
    4. Nacrtajte proces upravljanja rizikom pri razvoju softvera.
    5. Navedite vrste rizika i navedite po jedan primer za svaki od njih.
    6. Kako se mogu oceniti verovatnoža rizika. Kako se mogu oceniti efekti riyika.
    7. Šta je planiranje rizika. Koje se strategija u planiranju riyika najčešće koriste? Dajte primere.
    8. Šta je monitoring rizika. Koji su indikatori razika. Navedite ih po tipovima rizika.
    9. Koji su kritični faktori u upravljanu ljudima, tj. članovima razvojnog tima projekta? Objasnite svaki o kritičnih faktora.
    10. Koje potrebe ljudi utiču na njihovu motivaciju?
    11. Zašto je z Koja tri faktora utiču na rad tima?a razvoj projekta potrebno primeniti timski rad? Koje su koristi od kohezije tima?
    12. Navedite pitanja čiji odgovori određuju organizaciju grupe.
    13. Koja su tri nivoa planiranja projekt? Koj tri glvna prametra utiču na troškobe projekt razvija.
    14. Koji su uticajnu faktorna cenu softvera?
    15. Navedite tipičan plan projekta razvoja softvera? Nacrtajte dijagrm aktivnosti procesa planiranja projekta razvoja softvera.
    16. Koja je dodatna dokumnetacij potrebna pri planiranju projekta?
    17. Šta je termin plan projekta? Šta on definiše? Dajte dijagram izrade termin plana.
    18. Koja dva osnovna tipa planiranja se koriste pri planiranju agilnog razvoja softvera. Objasnite ih.
    19. Kako se procenjuje potreban rad na razvoju softvera?
    20. Navedite dijagram akcija procesa planiranja agilnog racrtajte COCOMO modelII azvoja