

R271 – Informacioni sistemi

1. Objasniti razliku između informacije i podataka. Navesti poželjne karakteristike informacija.

Klasifikacija složenosti informacija:

- Podatak
 - Primer: 45
 - Nema semantiku
- Informacija
 - Primer: Petar Petrović ima broj cipela 45
 - Ima semantiku i podrazumeva razumevanje odnosa
 - Podatak stavljen u neki kontekst tj. čini deo šireg konteksta. U prethodnom primeru nam informacije govori u broju cipela Petra Petrovića. Povezani podaci: Petar (ime), Petrović (prezime), 45 (broj cipela).
- Znanje
 - Primer: Prosečan broj cipela muškaraca u Srbiji je 42
 - Podrazumeva razumevanje obrazaca
 - Složene analize nad informacijama na osnovu kojih se izvode neki zaključci. Ovde se primenjuju metode istraživanja podataka.
- Mudrost
 - Podrazumeva razumevanje principa (nešto čemu težimu)
 - Potpuno razumevanje stvari izvučenih iz obrazaca

Poželjne karakteristike informacije:

- Tačne i precizne
 - Informacija nema nikakvu vrednost ako nije tačna
 - Informacija nije upotrebljiva i ako nije dovoljno precizna
 - Povezano sa poverljivosti
- Potpune
 - Potpune u kontekstu kome odgovaraju
 - Suprostavljeno sa ekonomičnosti
- Ekonomične
 - Za proizvodjenje, prenos, održavanje
 - Suprostavljeno sa potpunosti
- Fleksibilne
 - Upotrebljive u različitim kontekstima
- Bezbedne
 - Zastićene od neautorizovanih korisnika

2. Šta je sistem?

Definicija: "Sistem je neka kombinacija sastavnih delova koja predstavlja složenu ili jedinstvenu celinu". Problem je što formalne definicije nisu uvek prikladne jer:

- Granice sistema često nisu sasvim jasne:
 - Uglavnom postoje komunikacije sa drugim sistemima ili sa širim sistemom, pa je teško reći da je sistem izolovan.
 - Tada se postavlja pitanje kako definisati granicu našeg sistema u odnosu na te ostale.
 - Granice se često prinudno (i oštro) postavljaju kako bismo mogli da projektujemo taj sistem.
- Praktično svaki sistem predstavlja sastavni deo neko šireg sistema (povezano sa prethodnim).
- Praktično svaki sistem obuhvata neke podsisteme ili tesno sarađuje sa drugim sistemima (povezano sa prethodnim).

Odlike sistema:

- Postoje u nekom **okruženju** koje je šire od samog sistema. U odnosu na to okruženje, postoje nekakve **granice** koje odvajaju sistem. Kroz te granice postoji komunikacije koje definišemo kao **ulaze i izlaze** i sistema. Tu komunikaciju definišemo preko **interfejsa**.
- Sistem obuhvata neke **procese** (kako se poslovi rade) i može da se sastoji od **podsistema**, gde svaki podsistem ima svoju odgovarajuću ulogu. Tu takođe moraju da postoje i **kontrolni mehanizmi** (periodične provere stanja).

Karakteristike sistema:

- Jednostavnost/složenost (zavisi od broja komponenti i njihovih odnosa)
- Otvorenost/zatvorenost (da li komunicira sa drugim sistemima)
- Stabilnost/nestabilnost (koliko se često sistem menja)
- Prilagodljivost/neprilagodljivost (sposobnost menjanja u zavisnosti od promena u okruženju)
- Trajnost/privremenost (trajanje sistema kao celine)

Informacioni sistem nazivamo različite vrste sistema čiji je fokus obrada informacija.

3. Koji su osnovni aspekti informacionih sistema?

Svaki IS može da se posmatra iz tri osnovna ugla (aspekta):

- Poslovni ugao
- Ugao strukture i ponašanja
- Funkcionalni ugao

4. Objasniti poslovni aspekt informacionih sistema.

Svaki informacioni sistem se pravi sa nekim ciljem tj. radi prikupljanje nekih vrsta informacija za neku obradu. Očekujemo da informacioni sistem doprinese radu neke organizacije.

Definicije:

- IS se razvijen da bi podržao procese, upravljanje i odlučivanje u organizaciji...
 - Ovde je fokus na nameni
- IS je sistem koji posreduje u komunikaciji između svojih korisnika.
 - Ovde je fokus na komunikaciji.
 - U nekim organizacijama su bitniji procesi, a u nekim je bitnija komunikacija.
 - IS se prilagođava potrebama.

Pitanja:

- Šta je cilj IS-a?
- Zašto se IS pravi?

5. Objasniti strukturni aspekt informacionih sistema.

Posmatranje IS iz strukturnog ugla je primereno razvoju i održavanju (šta treba da napravimo).

Prvenstveno se posmatraju:

- Struktura
- Ponašanje pojedinačnih komponenti
- Međusobni odnosi komponenti
- Ponašanje IS kao celine

Definicije:

- IS je sistem koji čine ljudi (korisnici), podaci, procesi i interfejsi.

Pitanja:

- Od čega se IS sastoji?
- Kako se IS implementira?

6. Objasniti funkcionalni aspekt informacionih sistema.

Posmatranje IS iz funkcionalnog ugla je primereno konceptualnom modeliranju:

- Od interesa su, pre svega, funkcije koje IS ispunjava
- Na ovom nivou apstrakcije nije bitno zašto i kako se nešto implementira

Definicije:

- IS je sistem koji prikuplja, skladišti, obrađuje i isporučuje informacije...

Pitanja:

- Koje zadatke IS mora da izvršava?

7. Objasniti socijalnu dimenziju informacionih sistema.

Informacioni sistem ima dve značajne dimenzije: tehničku i socijalnu. Informacioni sistem dobija celovitost i pun smisao tek u socijalnom kontekstu kome pripada.

Svaki put kada se informacioni sistem uvodi u neko preduzeće, prave se manje ili veće promene rada tog preduzeća.

- Ako neke procese u IS automatizujemo, gde je prethodno te procese izvršavala ručno neka osoba, onda je ta osoba nezadovoljna uvođenjem IS-a (jer je IS direktno ugrožava).

- Povećanje monitoringa i kontrole: Odgovara ljudima na višim pozicijama, ali ne i na nižim pozicijama.

8. Šta su formalizovani i neformalizovani elementi informacionog sistema?

Formalizovani sistem:

- Plansko prikupljanje i izdavanje informacija
- Predefinisana pravila za rukovanje informacija
- Formalizovana razmena informacija
- Obično se za neke ili za sve operacije rukovanja informacijama upotrebljavaju savremene IT tehnologije

Neformalizovani sistem:

- Odsustvo planskog prikupljanja i izdavanja informacija
- Intuitivno rukovanje informacija (ali ne i predefinisano)
- Primenjeno samo u okruženjima sa malom količinom informacija (kao i malim brojem subjekata, predmeta i postupaka)
- Neformalna razmena podataka
- Ne upotrebljavaju se savremene IT

Primer formalizovanog elementa sa IT podrškom: Popunjavanje formulara.

Primer neformalizovanog elementa bez IT podrške: Pisanje izveštaja.

Primer neformalizovanog elementa sa IT podrškom: Mejl sistem.

9. Šta su informacione tehnologije?

“**Informacione tehnologije**” je savremeni izraz koji označava kombinaciju računarskih i telekomunikacionih tehnologija koje se upotrebljavaju u prikupljanju, obradi, skladištenju, razmeni i izdavanju informacija. U širem smislu obuhvataju sve tehnologije koje se tiču rada sa informacijama.

Informacioni sistemi ne moraju nužno da koriste savremene IT i IT ne predstavlja osnovni aspekt IS. Da bi IS koristio IT, on mora da bude formalizovan (što nije uvek trivijalno).

10. Navesti i objasniti bar jednu definiciju pojma „informacioni sistem“.

“Informacioni sistem je sistem koji prikuplja, skladišti i isporučuje informacije relevantne za poslovnu organizaciju, tako da su informacije dostupne i upotrebljive svima koji ih koriste... (pretežno funkcionalni aspekt posmatranja, sa manje izražajnim poslovnim delom za poslednji deo “... upotrebljive svim koji ih koriste...”).

“Informacioni sistem je uređeni sistem koji prikuplja, skladišti, obrađuje i isporučuje informacije o stanju domena...” (funkcionalni aspekt posmatranja)

“Informacioni sistem je uređeni sistem koji čine ljudi, podaci, procesi i interfejsi koji međusobno sarađuju na podršci i unapređivanju svakodnevnih operacija u poslovanju i omogućavaju rešavanje problema i donošenje odluka upravi i korisnicima” (strukturni aspekt posmatranja, sa malim prelazom na poslovni aspekt posmatranja)

11. Navesti bar pet najčešćih uloga informacionih sistema?

- Informacioni sistem se uglavnom realizuje radi povećanja profita nekog preduzeća.
- Uloga informacionog sistema se obično posmatra iz poslovnog ugla. Uloga IS u poslovnoj organizaciji može predstavljati osnovu za klasifikovanje IS.
- Jedan IS uglavnom ima više uloga

Najčešće uloge:

- Upravljanje resursima (Primer: kupovina robe, naručivanje, praćenje stanja, ...)
- Obrada transakcija (Primer: Banke - praćenje promena umesto stanja)
- Podrška proizvodnji
- Podrška odlučivanju (pomoć pri donošenju strateških odluka)
- Podrška planiranju
- Informisanje
- Upravljanje dokumentima i administrativnim poslovima
- ...

12. Navesti bar pet vrsta informacionih sistema?

Vrste IS:

- Poslovni (opšti) sistemi
 - podržava tok faktura, robe i kadrova
- Proizvodni
 - specifičan vid poslovnih sistema
 - Dodatno se prati tok proizvodnje
 - Koliko kojih delova je neophodno za formiranje automobila?
- Zdravstveni
 - Omogućava lekarima da vide dosije pacijenta
 - Veliki protok informacija
 - Visok stepen osetljivosti podataka
- Obrazovni
 - Škola, fakulteti, ...
- Naučni
 - Vrste informacija obično nisu prostog tipa
 - Primer: Fotografije
- Geografski
- Veb-zasnovani
- Sistem za obradu transakcija

- Transakcija = razmena nekih dobara:
 - podataka
 - novca
 - proizvoda
- Uglavnom se koriste relacije baze podataka
- Sistem za podršku odlučivanju
- Upravljački
 - Redovne informacije o redovnim aktivnostima
 - Kontrola i efikasnost poslovanja
 - Transakcioni podaci su osnovni izvor informacija
 - Izveštaji: redovni, zahtevani, izuzetni
- ...

13. Navesti i objasniti osnovne funkcije IS?

Osnovne funkcije svakog IS su:

- Memorijska funkcija
 - Podrazumeva održavanje slike stanja domena
 - Odvija se na dva osnovna načina:
 - Po zahtevu (eksplicitno): Kada se stanje domena promeni, korisnik na odgovarajući način informiše sistem o nastaloj promeni.
 - Autonomno (implicitno): Sistem autonomno, bez eksplicitnih akcija korisnika, prepoznaje i evidentira nastale promene stanja domena. Primeri okidača: Periodično, stigao mejl, ...
- Informativna funkcija
 - Podrazumeva pružanje informacija o stanju domena
 - Odvija se na dva osnovna načina:
 - Po zahtevu: Kada je korisniku potrebna neka informacija, on je eksplicitno zahteva.
 - Autonomno: Sistem autonomno, bez eksplicitnih akcija korisnika, obaveštava odgovarajuće korisnike o trenutnom stanju domena ili nastalim promenama. Primeri okidača: Promena rasporeda, ...
 - Može biti osnovna (isporučivanje osnovnih podataka o stanju) ili složena (isporučivanje rezultata analize podataka).
- Aktivna funkcija
 - Aktivna funkcija se odnosi na izvođenje akcija koje menjaju stanje domena
 - Odvija se na dva osnovna načina:
 - Po zahtevu: Korisnik eksplicitno nalaže sistemu da izvede potrebnu akciju.
 - Autonomno: Sistem autonomno, bez eksplicitnih akcija korisnika, a na osnovu odgovarajućih promena stanja domena, izvodi određene akcije. Primer: Prenos novca u banci (može da bude po zahtevu ili automatski).

14. Navesti i objasniti osnovne strukturne elemente IS?

Osnovni strukturni elementi IS su:

- Subjekti (ljudi)
- Podaci
- Procesi
- Interfejsi

15. Navesti i objasniti vrste subjekata IS?

U IS se često koristi pojam **stakeholder** koji može u ovom kontekstu da se prevede kao ulagač u sistem koji očekuje neku dobit. Svi subjekti koji se u narednom koraku navode su **stakeholder-i** (sem poslednje stavke, jer nisu direktno vezani).

Vrste subjekata u IS su:

- Vlasnici
 - Postavljaju ciljeve
 - Odgovorni za finansiranje razvoja i održavanja IS-a
- Korisnici
 - Osobe koje redovno neposredno koriste IS
- Projektanti
 - Osobe koje prevode poslovne zahteve i uslove sistema u tehnička rešenja
 - Projektuju baze podataka, ulazne i izlazne procese, aplikativne i korisničke interfejse, računarske mreže, programe i ostale potrebne vidove IT.
- Implementatori
 - Osobe koje proizvode komponente IS na osnovu projektnih specifikacija.
- Analitičari sistema
 - Osobe koje upravljaju razvojem i održavanjem IS-a kroz održavanje komunikacije sa svim ostalim subjektima.
- Proizvođači IT i konsultanti

16. Šta je razvojna metodologija?

Metodologija razvoja informacionih sistema je skup procedura, tehnika, alata i dokumentacionih tehnika koji pomažu razvijaoima sistema u njihovim naporima da izgrade nov IS. Pored toga što metodologija definiše ovaj skup, ona mora da definiše i njihov redosled primenjivanja (redosled primenjivanja tehnika i alata). Metodologija nam govori u kom trenutku koju tehniku (ili alat) da koristimo. Metodologija predlaže i načine planiranja, praćenja i procenjivanje uspešnosti procesa razvoja.

Metodologija je bazirana filozofijom. Filozofija nam govori šta su prioriteti u razvoju.

Primeri: Zadovoljiti korisnika, osnovni cilj razvoja je efikasnost, ... Od same premise zavisi šta je rezultat razvoja.

17. Navesti primere definicija razvojnih metodologija.

Definicije:

- “Metodologija je preporučen skup filozofija, faza, procedura, pravila, tehnika, alata, dokumentacije, upravljanje i obuke za razvijaoce informacionih sistema.”
- “Metodologija je skup principa i metoda, koji se u konkretnoj situaciji svodi na metode koje su najbolje prilagođene toj konkretnoj situaciji.”

18. Šta čini metodologiju razvoja IS?

Svaka metodologija se odlikuje:

- Idejom vodiljom (filozofija)
- ciljevima
- fazama životnog ciklusa
- tehnikama koje se primenjuje u određenim fazama
- alatima koji se upotrebljavaju u tehnikama
- ulogama subjekata u razvoju

19. Šta predstavlja filozofija metodologije razvoja IS?

Predstavlja osnovnu ideju ili skup ideja kojima se rukovodi metodologija. Metodologija se gradi u cilju ostvarivanja ili praćenja osnovne ideja.

U zavisnosti od toga koji softver pravimo, pa i koji informacioni sistem pravimo, mogu da nam odgovaraju različite metodologije. Neke metodologije više odgovaraju socijalnom aspektu, a neke strukturnom aspektu.

Jednostavni primeri:

- Dobro rešenje je sistem koji pruža najveću iskorišćenost računara.
- Dobro rešenje je sistem koji obezbeđuje najbolju dokumentaciju.
- Dobro rešenje je sistem koji je najjeftiniji u eksploataciji.
- Dobro rešenje je sistem koji se dopada vlasnicima
- Uvek težiti jednostavnosti.

20. Šta predstavljaju ciljevi metodologija razvoja IS?

- Različitost filozofija ima za posledicu da metodologije mogu da imaju i različite ciljeve.
- Cilj metodologije usmerava pažnju razvojnog tima na odgovarajuće aspekte sistema i utiče na izbor tehnika i alata.

21. Šta predstavljaju faze razvoja jedne metodologije razvoja IS?

Prilikom nastanka metodologija od 60-tih do 80-tih godina se smatralo da ako se dobro definiše redosled aktivnosti, onda se obično garantuje kvalitet rezultata. Ovo uglavnom nije tačno za

razvoj softvera, ali u slučaju IS i dalje ima smisla zbog prirode problema deliti razvoj na faze. Pri korišćenju modernih metodologija se teži ka projektovanju samo arhitekture, bez zalaženja u detalje. Problem kod IS je što to nije dovoljno. Jedan od glavnih elemenata informacionih sistema je baza podataka, čija struktura se veoma teško menja ako baza nije prazna. Zbog toga je dobro projektovanje strukture veoma neophodno.

- Svaka metodologija deli proces razvoja na neke faze kroz koje mora da se prođe pri razvijanju IS.
- Sadržaj i oblik faza, kao i njihov propisan tok, predstavljaju posledicu filozofije i cilja metodologije.
- Faze razvoja značajno utiču na mogućnosti primene metodologije pri rešavanju konkretnih problema.

22. Šta predstavljaju i čemu služe tehnike metodologije razvoja IS?

Tehnika je način obavljanja nekog posla. Jedan isti posao se često može uraditi primenom različitih tehnika. Različite metodologije predlažu različite tehnike (to ne znači da se ne koriste i tehnike drugih metodologije, već postoji jasna razlika u skupu tehnika nad kojim je fokus). Tehnike obuhvataju: načine istraživanja, načine predstavljanja, načine projektovanja, ...

23. Šta su proizvodi metodologije razvoja IS?

Svaka primena neke tehnike nam daje neki proizvod. Proizvod (ili artefakt) je svaki rezultat rada koji nastane tokom razvoja IS. Primeri:

- Baza podataka (proizvod modeliranja baze podataka)
- formular
- upustvo za upotrebu
- konceptualni model sistema
- dijagram toka podataka
- dijagram slučajeva upotrebe
- ...

24. Šta je životni ciklus IS?

Životni ciklus informacionog sistema predstavlja skup razvojnih faza, njihov sadržaj i redosled odvijanja.

25. Objasniti životni ciklus IS po modelu vodopada. Navesti osnovne faze.

Poznat i po imenima: konvencionalna analiza sistema ili kaskadni model.

Osnovna ideja ovog životnog ciklusa je da čitav razvoj podelimo na faze u predefinisanim redosledu tako da izlaz jedne faze predstavlja ulaz u drugu fazu. Nije moguće preći na sledeću fazu dok se prethodna ne završi potpuno i nije moguće vraćanje unazad. Spisak faza se razlikuje od metodologije do metodologije:

- Studija izvodljivosti
- Istraživanje sistema
- Analiza sistema
- Projektovanje sistema
- Implementacija
- Održavanje

26. Navesti ulogu, ishod i sadržaj faze životnog ciklusa IS „studija izvodljivosti“?

Studija izvodljivosti podrazumeva sledeće tri faze (istraživanje sistema, analiza sistema i projektovanje sistema) „na brzinu“. Ideja je da neki tim za neki kraći period (mesec dana) analizira problem i da okvir za projekat. Na osnovu tog rezultata se odlučuju da li se ulazi u projekat ili se ne ulazi u projekat. Alternativa je da se napravi još jedna iteracija studije izvodljivosti, ali sa drugim timom.

Studija izvodljivosti obuhvata osnovnu analizu:

- postojećeg stanja
 - uočavanje slabosti i problema
- željenog novog stanja
 - potrebna unapređenja
- Iz različitih aspekata:
 - pravnih
 - organizacionih
 - socijalnih
 - tehničkih
 - ekonomskih

Kao rezultat studije se dobija predlog idejnog rešenja. Realizacija projekata ne mora nužno da se izvrši na baš taj način, ali to predstavlja neku osnovu.

27. Navesti ulogu, ishod i sadržaj faze životnog ciklusa IS „istraživanje sistema“?

Podrazumeva detaljno sagledavanje postojećeg stanja i potrebnih promena (ne govorimo o novom stanju, ali znamo želje za promenama). Rezultat je iscrpna slika stanja.

Obuhvata iscrpno posmatanje sistema:

- funkcionalni zahtevi postavljeni pred postojeći sistem
- funkcionalni zahtevi koji se postavljaju pred novi sistem

- dodatni zahtevi i uslovi
- vrste i obim obuhvaćenih podataka
- posebni slučajevi

28. Navesti ulogu, ishod i sadržaj faze životnog ciklusa IS „analiza sistema“?

Analiza sistema pretpostavlja da imamo izdvojene sve postojeće informacije. Na osnovu tih informacija je jasno šta je tekući sistem i šta se traži da se popravi (koji su funkcionalni i nefunkcionalni zahtevi).

Uloga:

- Razumevanje postojećeg sistema
- Razumevanje problema
- Razumevanje prostora za unapređenje
- Oblikovanje zahteva

Ishod:

- Postavljeni zahtevi

Analiza sistema obuhvata analizu prikupljenih podataka o sistemu:

- Zašto problem postoji?
- Zašto se primenjuju baš postojeći postupci?
 - Da li postoje alternative?
 - Zašto nisu primenjene?
- Kako se menja obim podataka u vremenu?

29. Navesti ulogu, ishod i sadržaj faze životnog ciklusa IS „projektovanje sistema“?

Uloga projektovanja sistema je priprema za implementaciju:

- Strukturno definisanje sistema
- Određivanje delova sistema i njihovih odgovornosti
- Određivanje veza i komunikacija između delova sistema
- Određivanje izgleda informacija koje se čuvaju
- Određivanje poslovnih pravila koja moraju da se odvojaju
- ...

Proizvod je iscrpna dokumentacija o svim aspektima predstojeće implementacije sistema:

- ulazni podaci
- izlazni podaci
- tok procesa u sistemu
- strukture i organizacije podataka
- sigurnosni aspekti i procedure

- plan implementacije
- plan testiranja
- plan tranzicije

30. Navesti ulogu, ishod i sadržaj faze životnog ciklusa IS „implementacija“?

Implementacija podrazumeva izgradnju novog (unapređenog) sistema. Proizvod je nov napravljen sistem u skladu sa definisanim zahtevima.

Implementacija obuhvata:

- Pravljenje svih elemenata novog sistema
 - softver
 - hardver
 - organizacioni uslovi za primenu
- Kontrola kvaliteta
 - testiranje
 - stalne provere korektnosti
- Edukacija budućih korisnika
- Pripremu dokumentacije
 - razvojne
 - korisničke
- Implementacija sigurnosnih mehanizama
- Tranzicija sa starog na novi sistem

31. Navesti ulogu, ishod i sadržaj faze životnog ciklusa IS „održavanje“?

Nije greška reći da održavanje predstavlja iterativno ponavljanje svih faza osim prve faze. Uloga održavanje je očuvavanje saglasnosti sistema sa okruženjem, a proizvod je sistem prilagođen okruženju. Glavni cilj održavanje je produžavanje veka rada informacionog sistema.

Prilikom uvođenja novog informacionog sistema, često se on ne uvodi potpuno odjednom, nego delimično sa nekim delovima koji kasne. Zbog toga se dešava da se tek kasnije uvide neki bagovi. Jedan od ciljeva održavanje jeste i ispravka ovih bagova.

32. Objasniti dobre strane životnog ciklusa IS po modelu vodopada.

Model vodopada je vrlo dobar za manje projekte (do 6 meseci, evetualno do godinu dana), gde se ne očekuje da dođe do značajnih promena tokom vremena.

Dobre strane modela vodopada:

- Sistematičnost
- Proveren pristup
- Standardizovanost
- Planski pristup problemu razvoja
- Specijalizacija po fazama olakšava upravljanje poslovima i kadrovima

33. Objasniti slabosti životnog ciklusa IS po modelu vodopada.

Osnovna ideja modela vodopada je da nema vraćanja unazad. Posledica ovog pristupa je da je model veoma problematičan kad je neophodno vraćanje unazad.

Primer: Ukoliko smo unajmili honorarno tim analitičara za prvu fazu projekta, onda je pitanje da li će opet biti dostupni. Ako nisu dostupni, onda mora da se unajmi neki drugi tim koji mora prvo da analizira već urađeno (što je veoma skupo). Ukoliko imamo tim analitičara, onda oni trenutno rade na nekom drugom projektu. U tom slučaju moramo da ih povučemo sa tog projekta kako bi radili na našem projektu.

Druga posledica nevraćanja je razvijanje projekta koji će morati da se menja tokom održavanja, tako da se troše dodatni resursi za nešto što neće biti primenljivo prilikom završetka implementacije.

Slabosti:

- Mogu se “promašiti” potrebe vlasnika
 - potrebe se neposredno posmatraju samo u prvoj fazi
- Nestabilnost
 - U prvim fazama se najviše pažnje posvećuje procesima, a procesi su promenljivi
- Nefleksibilnost
 - Mnogo toga se “zacrtava” u ranim fazama i teško se naknadno menja
- Nezadovoljstvo korisnika
- Neusklađenost dokumentacije
 - Dokumentacija često počiva na ranim fazama, a rezultat se kasnije menja
- Nedostatak kontrole
 - Zbog velike raslojenosti je veoma teško unapred proceniti obim pojedinih faza
- Nekompletnost sistema
 - Neki specijalni slučajevi se ne uzimaju na vreme
- Dugo se čeka na gotov sistem
- Idealizacija
 - Preterano komplikovanje nekih detalja
- Pretpostavka belog papira
 - Koncept je prilagođen razvoju novih sistema, ali sistemi se često ne prave od nule

34. Kada nije rizično primeniti u praksi životni ciklus IS po modelu vodopada?

Model vodopada nije rizično primeniti na manje projekte gde rukovodioci tima imaju iskustva u sličnim projektima. Model vodopada je zapravo dobar model za projekte gde je verovatnoća da pođe nešto po zlu mala.

35. Koje su karakteristike spiralnih životnih ciklusa?

Slabosti modela vodopada se pokušavaju rešiti drugim oblicima životnog ciklusa. Primer tih oblika je spiralni životni ciklus. Ideja je da se faze projekta ponavljaju od početka. Nakon prve iteracije se često ne javljaju baš sve faze iz osnovne iteracije.

36. Koje vrste tehnika postoje u okviru razvojnih metodologija IS?

Prema predmetu bavljenja i konceptu na kome počivaju, tehnike se dele na:

- Tehnike opisivanja procesa
- Tehnike opisivanja podataka
- Objektno-orijentisane tehnike
- Holističke tehnike
- Tehnike upravljanja projektom
- Organizacione tehnike
- Kadrovske tehnike

37. Objasniti ulogu tehnika za opisivanje procesa. Navesti bar 5.

Tehnike za opisivanje procesa imaju veliki značaj u razumevanju i modeliranju sistema

- Procesi predstavljaju jednu od osnovnih komponenti sistema
- Modeliranje procesa omogućuje da se bolje razumeju odnosi subjekata i objekata u sistemu
- Ipak, procesi nisu stabilni i kroz životni vek sistema trpe relativno česte promene.

38. Navesti bar 7 tehnika za opisivanje procesa.

Tehnike za opisivanje procesa:

- Tehnike za opisivanje poslovne logike / pravila odlučivanja
 - Drvo odlučivanja
 - Tabele odlučivanja
 - Strukturiran tekst
 - Dijagram akcija
- Tehnike za praćenje projekta
 - Strukturni pregled
- Tehnike za opisivanje dekompozicije / sktrukture

- Dijagram toka podataka
- Matrice
- Dijagrami strukture
- Životni ciklus entiteta
- Dijagram BPMN

39. Objasniti funkciju i osnovne elemente dijagrama toka podataka?

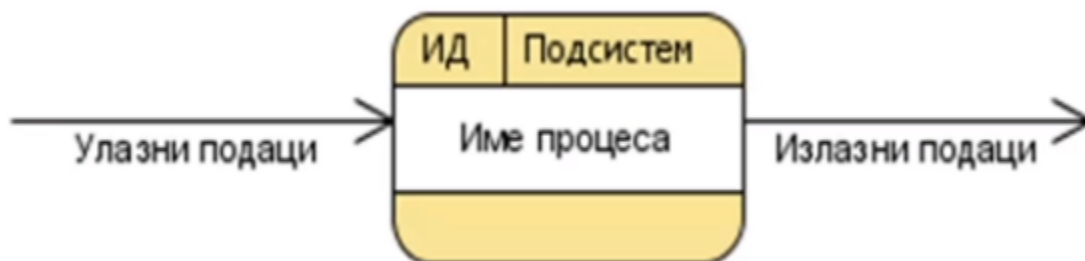
Dijagram toka podataka - DTP (end. dataflow diagram - DFD) je jedna od osnovnih tehnika strukturnih metodologija, ali upotrebljava se i u drugim metodologijama:

- Modeliranje poslovne strukture
- Funkcionalna dekompozicija sistema
- Modularno predstavljanje okruženja
- Koncizno grafičko predstavljanje
 - razumljivo svim subjektima

40. Navesti, nacrtati i objasniti osnovne elemente dijagrama toka podataka.

Osnovni elementi:

- Tok podataka (strelice)
- procesi (zaobljeni pravougaonici)



- складишта података (отворени pravouganoci koji sa desne strane nemaju ivicu)
 - datoteke
 - baze podataka
 - fizička складишта
 - ...



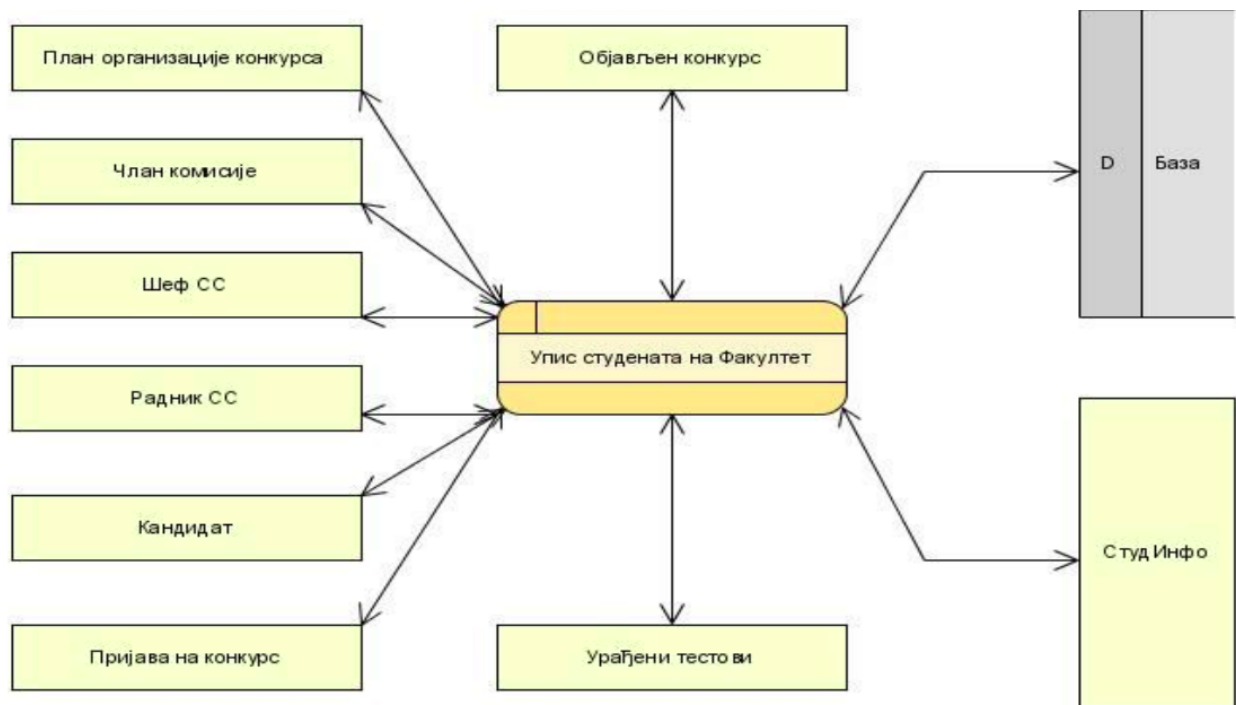
- izvori i ishodi



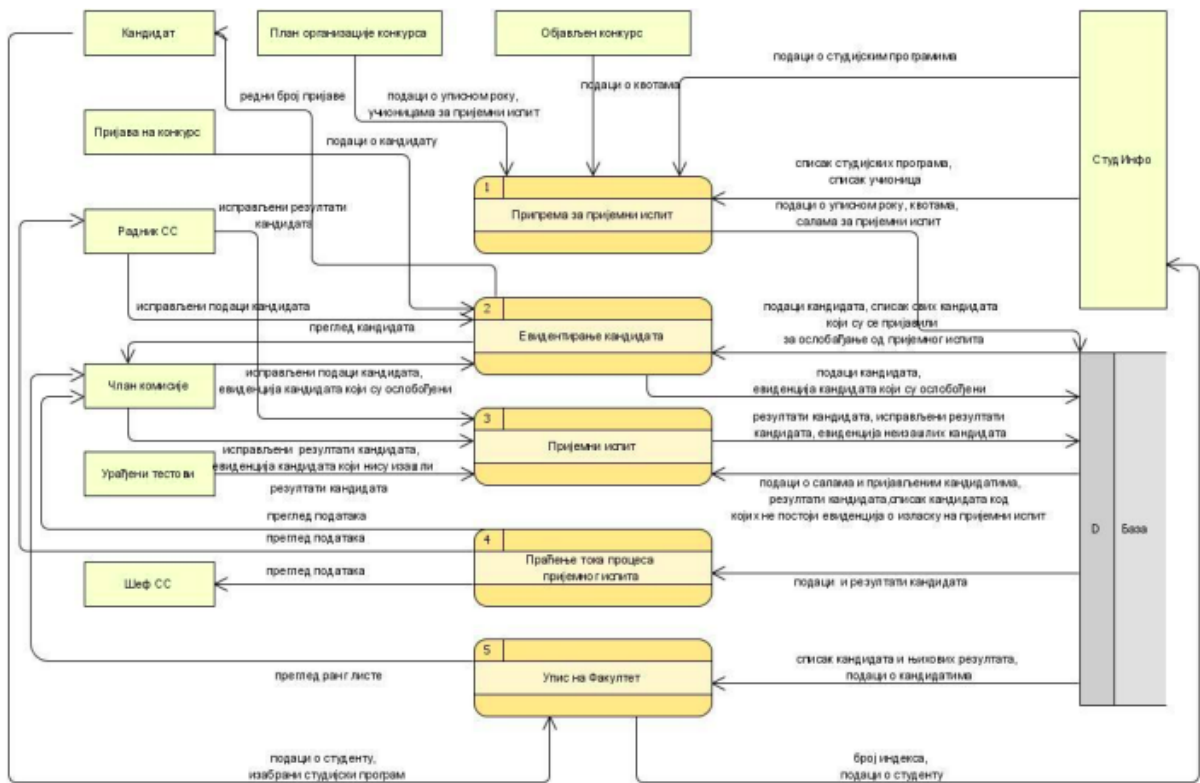
DTP se često prikazuju na više nivoa apstrakcije počev od nultog nivoa koji je na najvišem nivou apstrakcije, pa ostalim, višim nivoima u kojima se više ulazi u detalje. Dijagram nivoa 0 imamo samo jedan, a dijagrame nivoa 1 imamo otprilike isto koliko i procesa na nultom nivou (zavisi koliko razbijamo procese). U nekim literaturama se dijagram nultog nivoa naziva dijagram konteksta, a u drugim je na najvišem nivou apstrakcije dijagram konteksta, a dalje kreće od nultog nivoa.

41. Nacrtati primer dijagrama toka podataka u kome se pojavljuju svi osnovni elementi ove vrste dijagrama.

Dijagram konteksta:



Dijagram nultog nivoa (prvog nivoa ako je dijagram konteksta isto što i dijagram nultog nivoa):



42. Šta je dijagram konteksta? Koji su njegovi osnovni elementi?

Pogledati pitanja 40 i 41.

43. Šta obuhvata dijagram konteksta?

Pogledati pitanja 40 i 41.

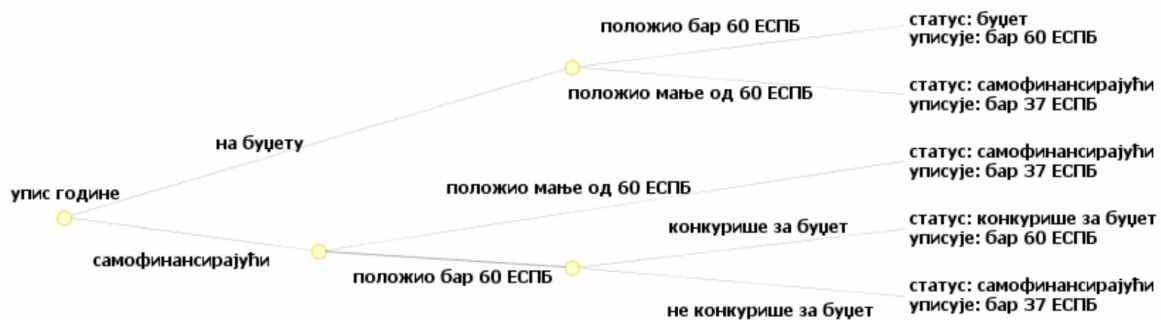
44. Nacrtati primer dijagrama konteksta.

Pogledati pitanja 40 i 41.

45. Šta je drvo odlučivanja? Kada se koristi i čemu služi?

- Drvo odlučivanja je tehnika koja se upotrebljava za modeliranje načina odlučivanja.
 - Relativno uopšteno
 - Ali pre svega pri izboru procesa koji se primenjuje
- Način odlučivanja se predstavlja u obliku drveta
 - U korenu drveta je opšti slučaj
 - Svakim grananjem se opšti slučaj deli na više specifičnih
 - U listovima su poznate odluke
 - Opisi procesa koji se primenjuju
 - Obično se grananje predstavlja udesno

46. Napraviti primer drveta odlučivanja.



47. Šta je tabela odlučivanja? Kada se koristi i čemu služi?

Tehnika koja se upotrebljava za modeliranje načina odlučivanja. Nadopunjuje se sa drvetom odlučivanja tj. predstavlja precizniju i kompaktniju tehniku.

Odlučivanje se predstavlja tabelom

- u gornjem delu se predstavljaju uslovi
 - vrste odgovaraju pojedinačnim uslovima
- u donjem delu se predstavljaju akcije
 - vrste odgovaraju akcijama
- kolone tabele odgovaraju skupovima uslova
 - u preseku kolone i vrsta koja odgovara uslovu se označava da li je uslov ispunjen
 - u preseku kolone i vrsta koja odgovara akciji se označava da li se akcija primenjuje

Tabela odlučivanja je kompaktnija, jer je lakše praviti tabelu u nekom alatu (excel), nego crtati stabla.

48. Napraviti primer tabele odlučivanja.

Статус је буџет	ДА	ДА	не	не	не
Статус је самофинансирање	не	не	ДА	ДА	ДА
Положено бар 60 ЕСПБ	ДА	не	ДА	ДА	не
Положено мање од 60 ЕСПБ	не	ДА	не	не	ДА
Конкурише за буџет	-	-	ДА	не	-
Уписује буџет	Х				
Уписује "конкурише за буџет"			Х		
Уписује самофинансирање		Х		Х	Х
Мора да упише бар 60 ЕСПБ	Х		Х		
Мора да упише бар 37 ЕСПБ		Х		Х	Х

49. Šta je strukturiran tekst? Kada se koristi i čemu služi?

Tehnika koja se upotrebljava za modeliranje načina odlučivanja i opisivanje toka procesa. Opisivanje se vrši pomoću struktuiranog teksta koji liči meta program.

50. Napraviti primer strukturiranog teksta.

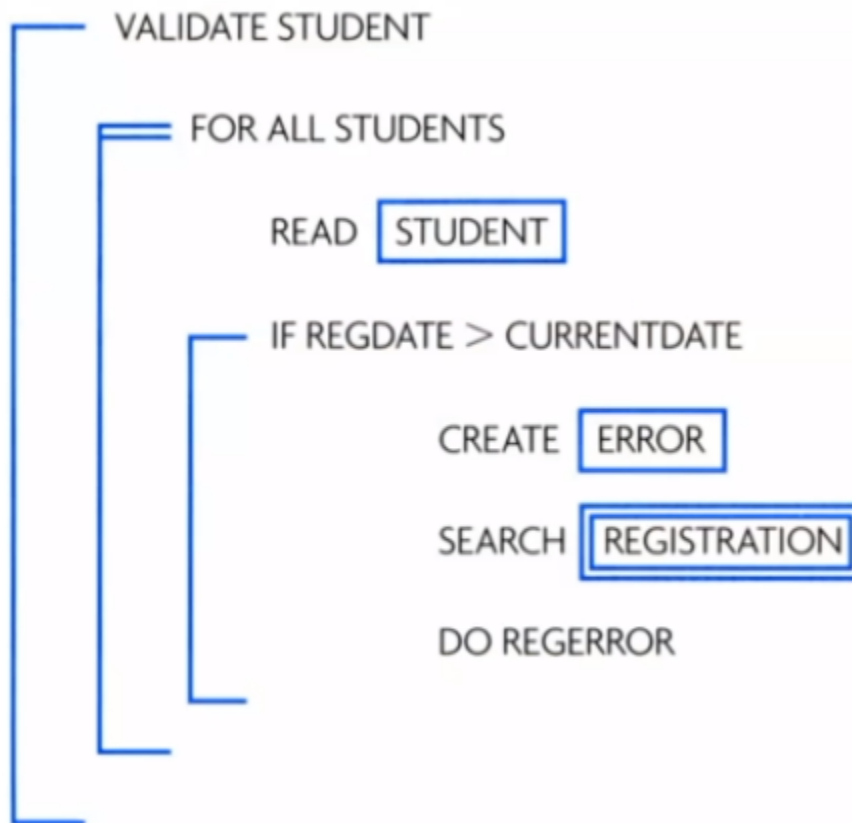
ако је студент на буџету
и положио је бар 60 ЕСПБ
онда уписује годину на буџету, са бар 60 ЕСПБ
иначе, **ако** је положио мање од 60 ЕСПБ
или из било ког разлога не конкурише за буџет
онда уписује годину као СФ, са бар 37 ЕСПБ
иначе добија статус “конкурише за буџет” и уписује бар
60 ЕСПБ

51. Šta je dijagram akcija? Kada se koristi i čemu služi?

Tehnika dijagrama akcija je isto što i struktuiran tekst, ali nudi vizualizaciju. Koristi se na više nivoa: za pregledno predstavljanje i za detaljno predstavljanje. Osnovni element dijagrama je blok (kao zgrade sa leve strane struktuiranog teksta) koji odgovara bloku u programskom jeziku ili strukturnom tekstu.

Dijagrami akciju predstavljaju, ukratko rečeno, uopšteni način za pisanje algoritama.

52. Napraviti primer dijagrama akcija.



53. Navesti bar 3 vrste matrica koje se upotrebljavaju pri opisivanju procesa.

Matrice se upotrebljavaju za modeliranje i predstavljanje odnosa u veoma različitim slučajevima. Neke od primena obuhvataju predstavljanje:

- odnosa poslova i funkcija
- odnosa entiteta (podataka) i funkcija
- odnosa funkcija i subjekata
- odnosa entiteta (podataka) i subjekata
- ...

54. Šta je matrica poslova i funkcija? Kada se koristi i čemu služi?

Namena matrice poslova i funkcija je:

- Dokumentovanje odnosa poslova i funkcija
- Prepoznavanje funkcija sa lokalnim i širim značajem
- Prepoznavanje atomičkih funkcija

Kolone predstavljaju poslove, a vrste funkcije (ili obrnuto)

- Poslovi obuhvataju postupke ili događaje koji se dešavaju u okviru domena
- Funkcije predstavljaju pojedinačne operacije
- Svakom poslu mora da odgovara barem jedna funkcija
- Broj i veličina tablice zavisi od veličine i složenosti sistema

55. Napraviti primer matrice poslova i funkcija.

Посао	Упис на факултет	Упис године	Исписивање са факултета
Функција			
Пријављивање кандидата	X		
Уписивање на факултет	X		
Промена статуса		X	X
Избор предмета	X	X	
Уписивање године	X	X	
Одређивање школарине	X	X	
Проверавање дуговања		X	X
Евидентирање уплате	X	X	X

56. Šta je matrica entiteta i funkcija? Kada se koristi i čemu služi?

Namena matrice entiteta i funkcija je dokumentovanje odnosa entiteta i funkcija, i prepoznavanje entiteta sa lokalnim i širim značenjem. Nad entitetima se vrše CRUD (PČMB) operacije:

- Pravljenje (eng. create) - funkcija pravi novi entitet
- Čitanje (eng. read) - funkcija čita postojeći entitet
- Menjanje (eng. update) - funkcija menja postojeći entitet
- Brisanje (eng. delete) - funkcija briše postojeći entitet

Isto kao kod matrice entiteta i funkcija, broj tablica zavisi od veličine i složenosti sistema.

57. Napraviti primer matrice entiteta i funkcija.

Ентитет	Пријава на конкурс	Индекс	Пријава за упис године	Уписана година	Уписан курс	Фин. обавеза
Функција						
Пријављивање кандидата	П					
Уписивање на факултет	Ч	П				
Промена статуса		ЧМ				
Избор предмета			П			
Уписивање године			ЧМ	П	П	
Одређивање школарине		Ч		ЧМ	Ч	
Проверавање дуговања		Ч				Ч
Евидентирање уплате		Ч				ЧМ

58. Šta je dijagram strukture? Kada se koristi i čemu služi?

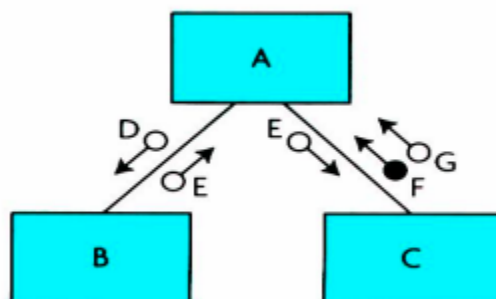
Predstavljaju funkcionalnu dekompoziciju problema (hijerarhijska organizacija).

Struktura se predstavlja grafički:

- Pravougaonicima se predstavljaju procesi ili moduli
- Linijama se povezuju procesi/moduli
- Uz linije se navode podaci koji se prenose

Danas se uglavnom koriste dijagrami komponenti umesto ovog dijagrama.

59. Napraviti primer dijagrama strukture.



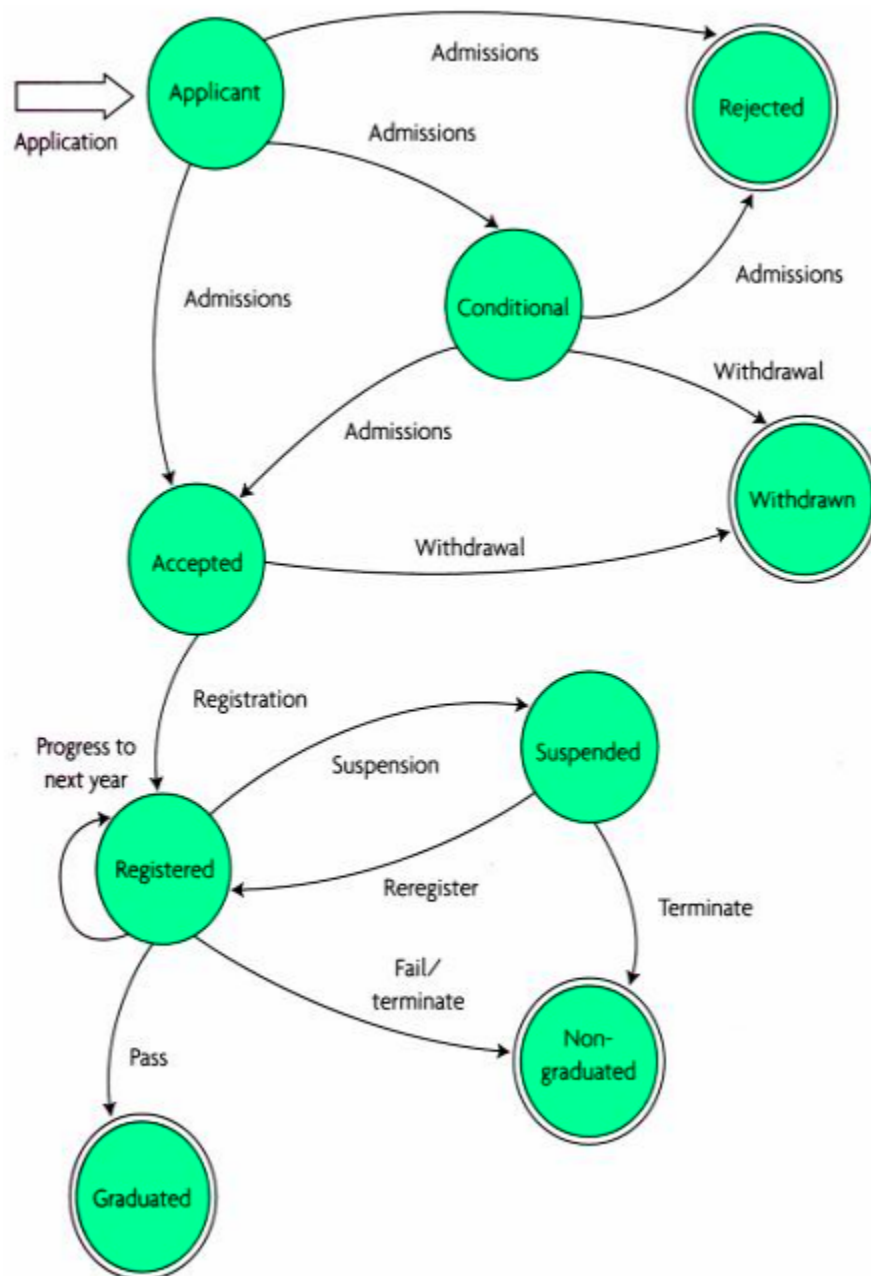
60. Šta je životni ciklus entiteta? Kada se koristi i čemu služi?

Tehnika koja odgovara UML dijagramu stanja. Koristi se za opisivanje stanja kroz koja prolazi neki entitet

Osnovni elementi:

- Stanja predstavljaju krugove
- Strelice predstavljaju uslovne prelaske
- Početno stanje je označeno velikom strelicom sa nazivom procesa koji pravi objekat
- Podebljani krugovi su završna stanja

61. Napraviti primer životnog ciklusa entiteta.



62. Šta je BPMN? Kada je nastala i gde se koristi?

BPMN (Business Process Modeling Notation) je napredna notacija za predstavljanje poslovnih procesa. Postoje dve osnovne vrste dijagrama koje odgovaraju ovoj notaciji:

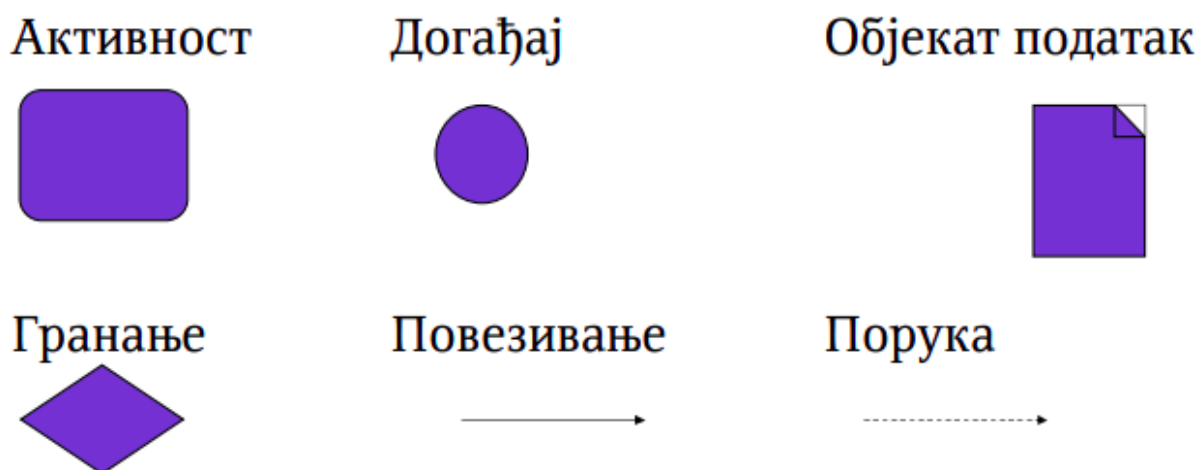
- dijagram procesa
- dijagram saradnje

Razvijena je kada je ustanovljeno da odgovarajuća tehnika u UML ne postoji. Predstavlja savremeniju alternativu za DTP, u smislu da omogućava više informacija, veću sadržajnost i ima vremenski tok (što je jedna od glavnih mana DTP dijagrama).

63. Koje vrste dijagrama poslovnih procesa postoje u okviru BPMN? Po čemu se razlikuju?

Postoje dijagrami procesa i dijagrami saradnje. Razlika je u tome što u dijagramu saradnje grupišemo celine po subjektima koji izvode te procese (swimlanes).

64. Koji su osnovni elementi dijagrama procesa BPMN? Nacrtati ih i objasniti.

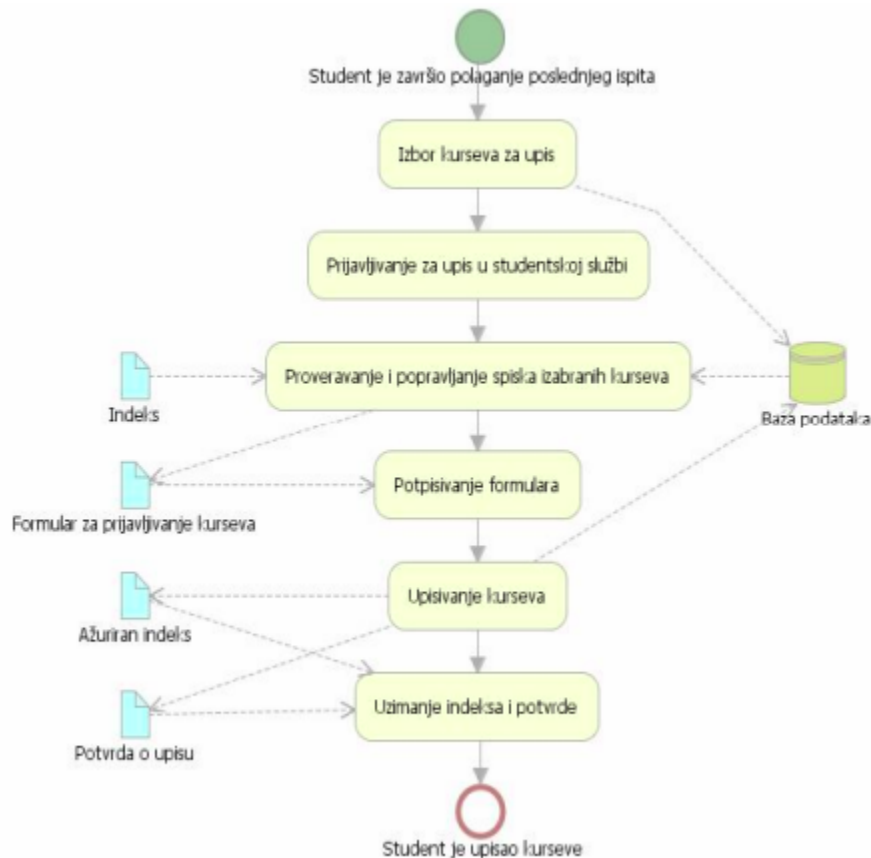


Elementi:

- Aktivnost (zaobljeni pravougaonik)
- Povezivanje (puna strelica)
 - tok informacija (tok procesa)
 - iz jedne aktivnosti prelazimo na drugu aktivnost
- Grananje (romb)
- Objekat podatak (ikona datoteke)
 - Iz aktivnosti može da protekne neki podatak. U tom slučaju isprekidanom strelicom (porukom) se povezuje aktivnost i objekat podatak.
 - Aktivnost može da koristi neki podatak. U tom slučaju isprekidanom strelicom (porukom) se povezuje objekat podatak i aktivnost.
- Poruka (isprekidana linija)
- Događaj (krug)
 - Mogu da predstavljaju početak i kraj
 - Primeri: Stigao je mejl, tajmer, ...

- Obični krugovi se odnose na događaje koji započinju nešto
- Podebljani krugovi se odnose na događaje koji završavaju nešto
- Krugovi sa duplim linijama se odnose na događaje koji se dešavaju “u sred nečega”

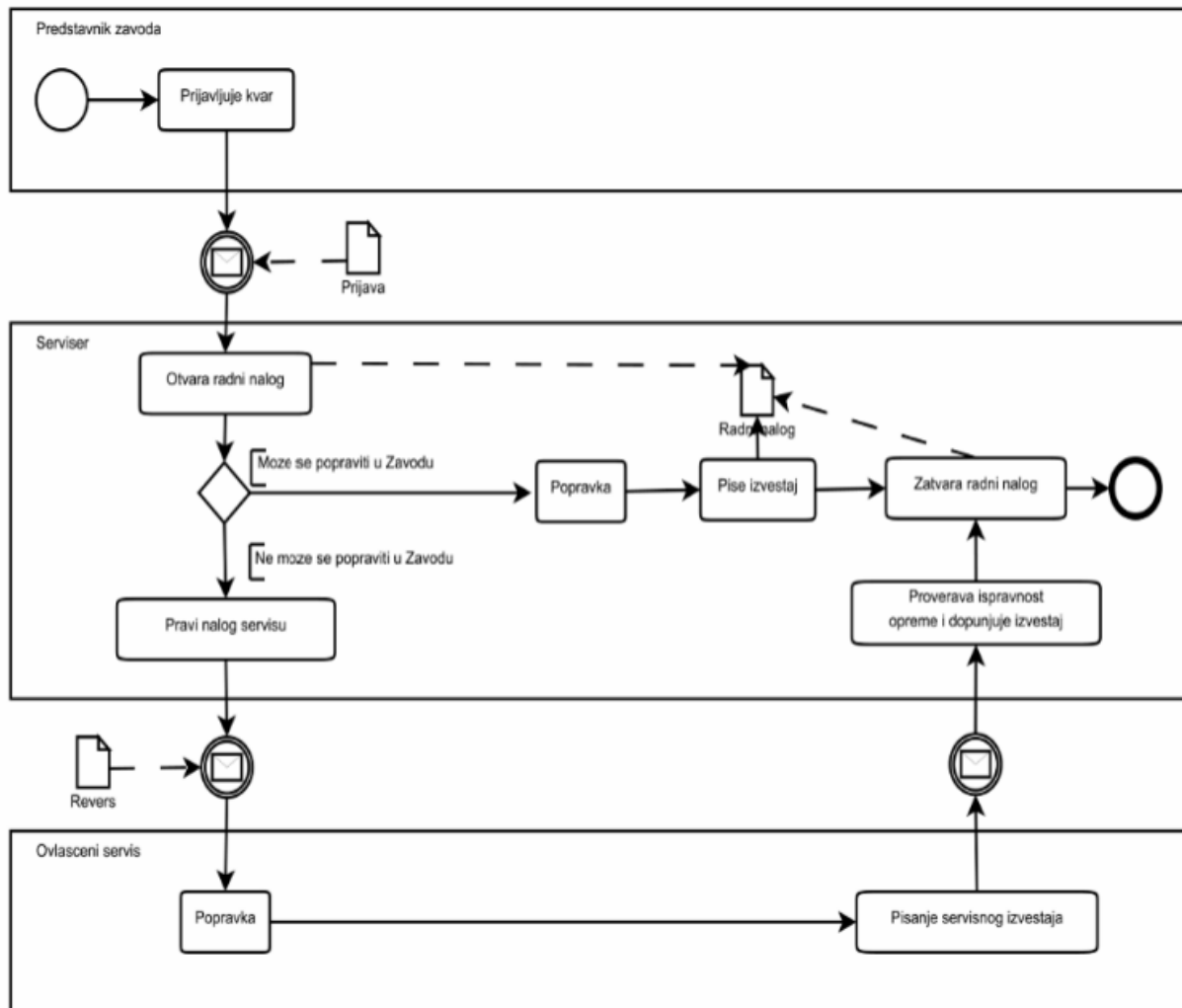
65. Nacrtati primer dijagrama procesa BPMN.



66. Koji su osnovni elementi dijagrama saradnje BPMN? Nacrtati ih i objasniti.

Svi osnovni elementi kao i kod dijagrama procesa BPMN zajedno sa grupisanim aktivnostima po akterima.

67. Nacrtati primer dijagrama saradnje BPMN.



68. Objasniti ulogu tehnika za opisivanje podataka. Navesti osnovne.

Tehnike za opisivanje podataka se koriste da modeliraju:

- strukturu podataka
- međusobne odnose podataka

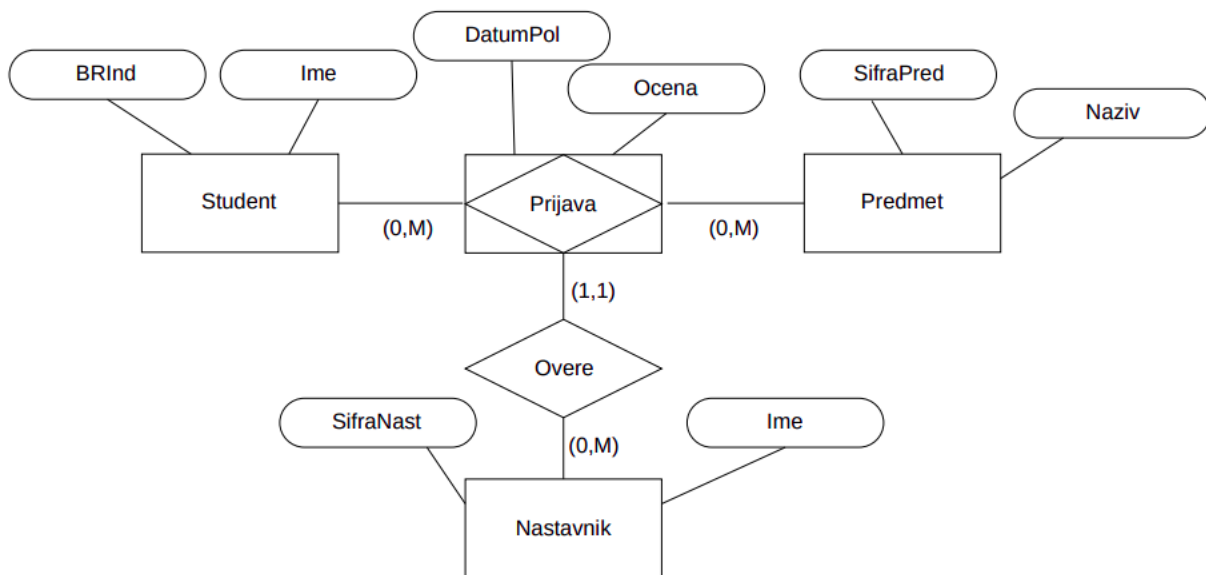
Najvažnije tehnike su:

- Dijagrami entiteta i odnosa
- Dijagrami tabela
- Dijagrami klasa podataka

69. Šta je dijagram entiteta i odnosa? Koji su njegovi osnovni elementi?

Dijagram entiteta i odnosa (ER dijagram) je tehnika za opisivanje podataka (modeliranje podataka). Ova tehnika je prvenstveno namenjena za konceptualno modeliranje (nije pogodna za niže nivoe apstrakcije) i relativno se lako prevodi u logički model tabela. Ova tehnika je slična tehnici dijagrama klasa podataka.

70. Nacrtati primer dijagrama entiteta i odnosa.



71. Šta je dijagram klasa entiteta?

Dijagram klasa entiteta je tehnika za opisivanje podataka (modeliranje podataka).

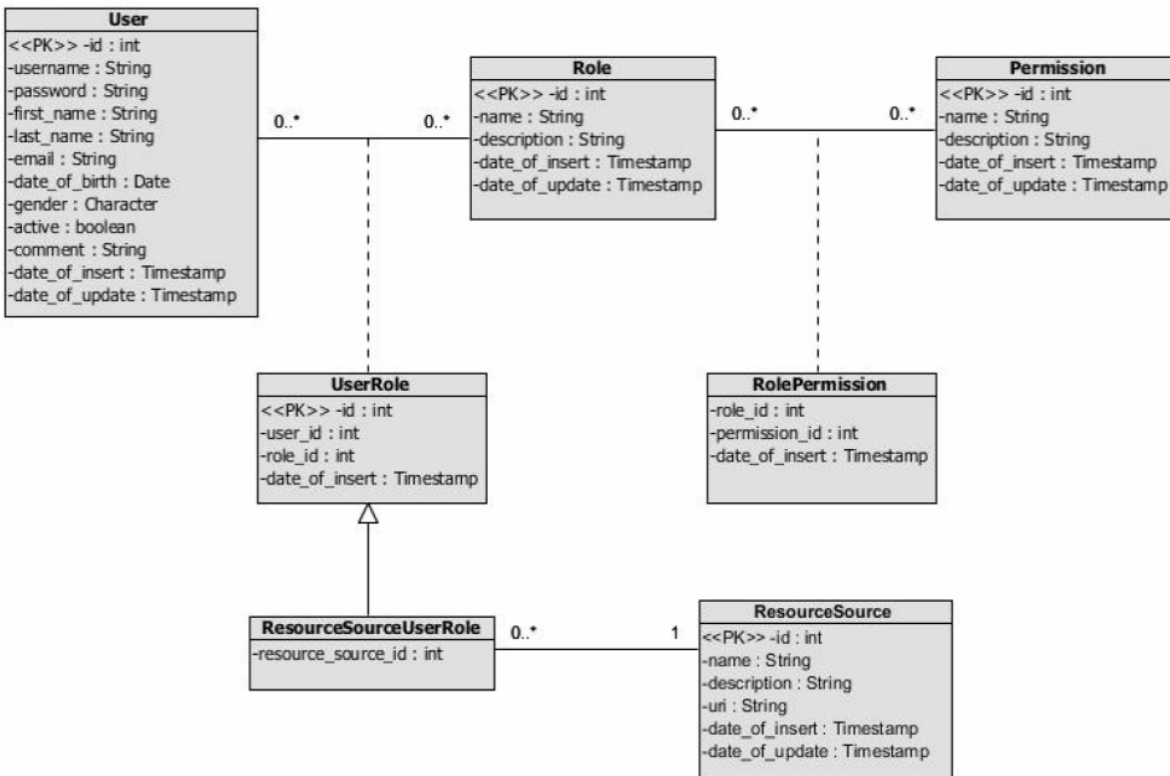
Upotrebljavaju se često umesto ER dijagrama i čini širu tehniku od ER dijagrama. Ideja je da se u odnosu na klasičan dijagram klasa izostave metode, enkapsulacija, privatni i javni delovi, ... a da se akcenat stavi na strukturu klasa (podaci) i njihove veze.

- Takođe se lako prevodi na relacioni model.
- Koristi se najviše za konceptualni model podataka.

Postoje tri tipa dijagrama klasa:

- Dijagram klasa podataka (zovu se i dijagrami klasa domena i dijagrami klasa entita)
 - Izostavljaju se metode, enkapsulacija, privatni i javni delovi... (ponašanje)
 - Akcenat je na strukturi podataka i njihovim vezama
- Dijagram klasa ponašanja
 - Izostavljaju se enkapsulirani delovi i prikazuju se samo javni metodi
- Dijagram klasa implementacije
 - Sadrži sve

72. Nacrtati primer dijagrama klasa entiteta.



73. Objasniti razliku između dijagrama entiteta i odnosa i dijagrama klasa entiteta.

- Ako pravimo logički ili fizički model
 - za relacionu bazu podataka, onda je najbolje da se koristi dijagram klasa podataka
 - za nerelacionu bazu podataka, onda je neophodna neka sopstvena tehnika ili tehnika prilagođena toj bazi podataka
- Ako pravimo konceptualni model, onda koristimo EER dijagrame ili dijagrame klasa entiteta

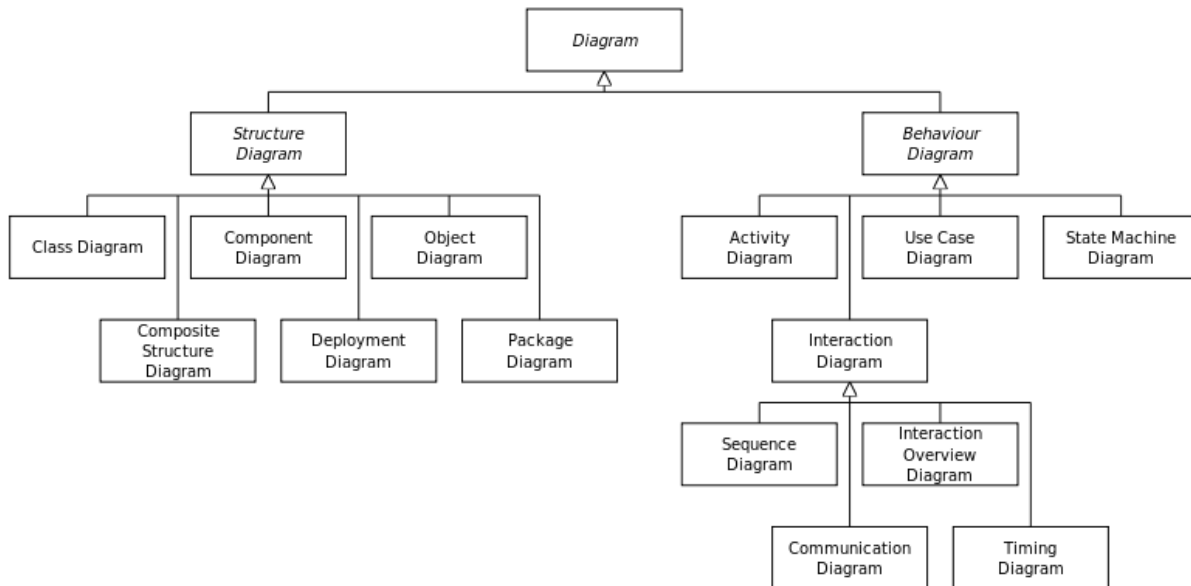
Razlika između EER i dijagrama klasa entiteta:

- Dijagrami klasa entiteta se dobro uklapaju u kontekst OO metodologija i tehnika.
- Dijagrami klasa entiteta su često pregledaniji i lakši za upotrebu.
- Obe tehnike su pogodne za modeliranje na konceptualnom nivou.

74. Šta je UML? Kada se upotrebljava? Čemu služi?

UML (Unified Modeling Language) je nastao objedinjavanjem više različitih pristupa OO projektovanju i odgovarajućih jezika modeliranja. Predstavlja skup tehnika.

75. Koje vrste dijagrama postoje u UML-u? Kako se klasifikuju?



Klase i vrste dijagrama:

- Strukturni dijagrami
 - Dijagrami klasa
 - Dijagrami komponenti
 - Dijagrami objekata
 - Dijagrami složene strukture
 - Dijagrami isporučivanja
 - Dijagrami paketa
- Dijagrami ponašanja
 - Dijagrami aktivnosti
 - Dijagrami slučajeva upotrebe
 - Dijagrami stanja mašine
- Dijagrami interakcija (podklasa dijagrama interakcije)
 - Dijagrami sekvenci
 - Dijagrami komunikacije
 - Dijagrami interakcije
 - Dijagrami vremena

76. Navesti bar 5 vrsta strukturnih dijagrama UML-a i sasvim kratko objasniti kada se koji upotrebljava.

Pogledati pitanje 75.

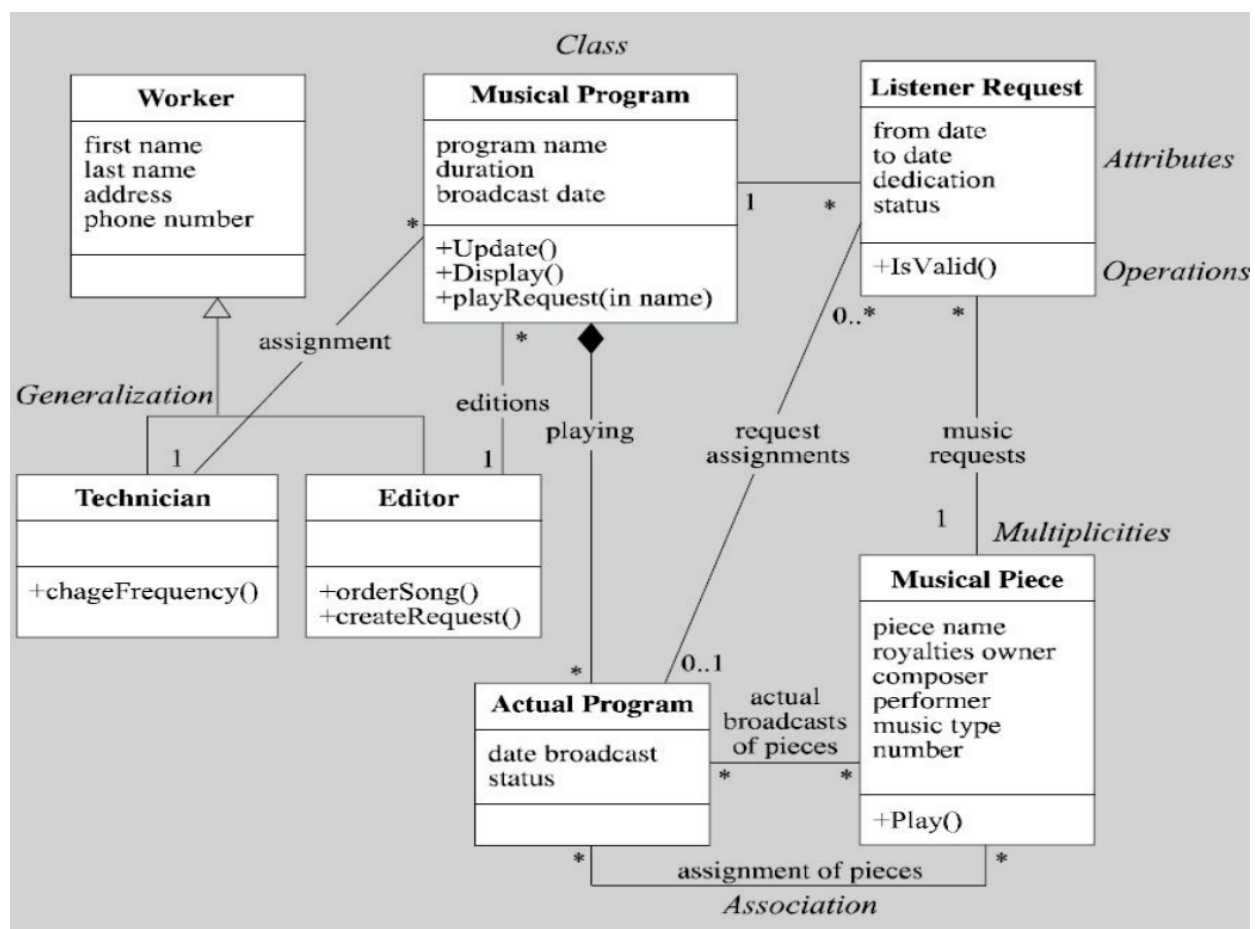
77. Šta je dijagram klasa? Navesti i objasniti osnovne elemente dijagrama klasa (UML).

Opisuje elemente statičkog modela, kao što su klase, njihov sadržaj i međusobne odnose.

Sadrži:

- Nazive klasa
- Atribute klasa
- Metode klasa
- Specijalizaciju i generalizaciju
- odnose
 - asocijaciju
 - agregaciju
 - kompoziciju

78. Navesti primer dijagrama klasa koji sadrži sve glavne elemente (UML).



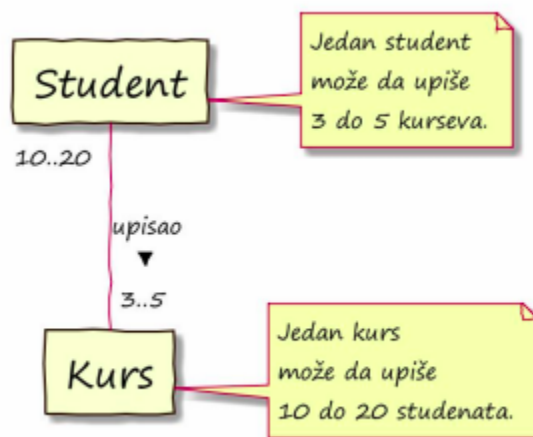
79. Navesti vrste asocijacija (dijagram klasa), objasniti i ilustrovati primerom svaku od vrsta (UML).

Ako je neophodno da dve klase komuniciraju između sebe, onda postoji veza između njih. Ovo nazivamo asocijacija. Vrste asocijacija kod dijagrama klasa:

- asocijacija
 - Postoji veza između klase A i B
 - Primer: Kurs i student
- kompozicija
 - Klasa A sadrži klasu B i klasa B ne može da postoji bez klase A
 - Primer: Direktorijum sadrži datoteke
- agregacija
 - Klasa A sadrži klasu B, ali klasa B može da postoji bez klase A
 - Primer: Pištolj i meci (igrica)

80. Kako se označava kardinalnost na dijagramu klasa (UML)? Nacrtati primer.

Kardinalnost se označava granicama X..Y i mogu da budu nula-jedan, jedan-jedan, jedan-više, više-više. Primer:



81. Koje elemente ima jedna klasa na dijagramu klasa (UML)?

Jedna klasa ima naziv, metode i atribute. Metode i atributi imaju oznake da li su javni ili privatni.

82. Šta je, čemu služi i kada se upotrebljava dijagram komponenti (UML)?

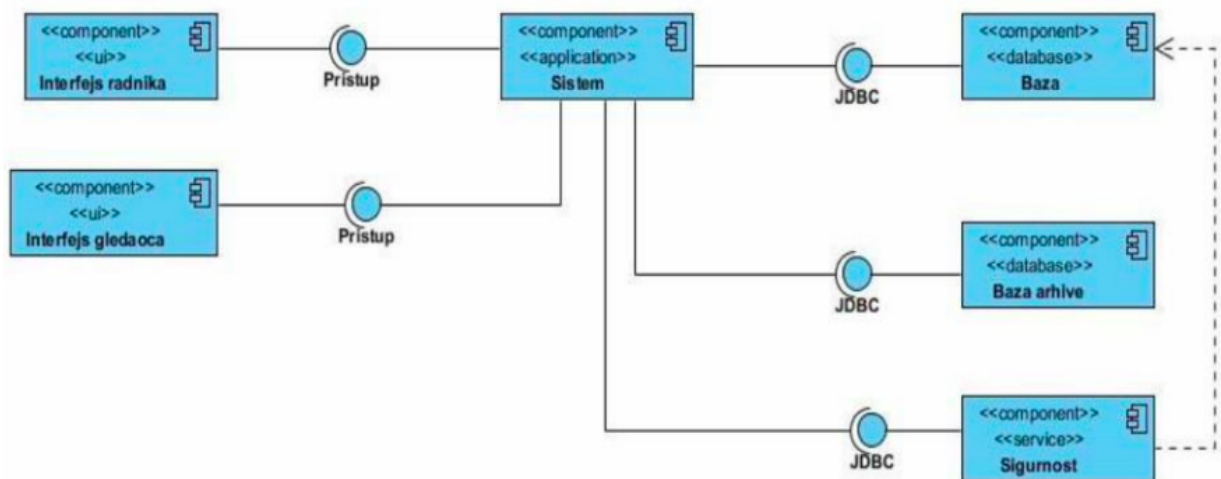
Dijagram komponenti ilustruje komponente koje čine aplikaciju, sistem ili organizaciju. Koristi se u funkcionalnoj dekompoziciji sistema. Sadrži:

- nazive komponenti
 - Komponente imaju svoju strukturu i način funkcionisanja, ali se na ovom nivou enkapsulira
- njihove međusobne veze
 - Uglavnom se misli na interne veze komponenti, a ne veze sa korisnikom
 - Nije greška ako se posmatraju i veze sa korisnikom
 - Ne smeju se zanemarivati veze komponenti sa spoljnim komponentama (spoljnim sistemima)
- javne interfejsse
 - Interfejsse ka drugim komponentama

83. Koji su osnovni elementi dijagrama komponenti (UML)?

Osnovni elementi dijagrama komponenti:

- Komponente
 - Pravougaonik sa simbolom u gornjem desnom uglu
- Veze sa krugovima u sredini
 - Krug - interfejsi
 - Okvir kruga - korišćenje interfejsa



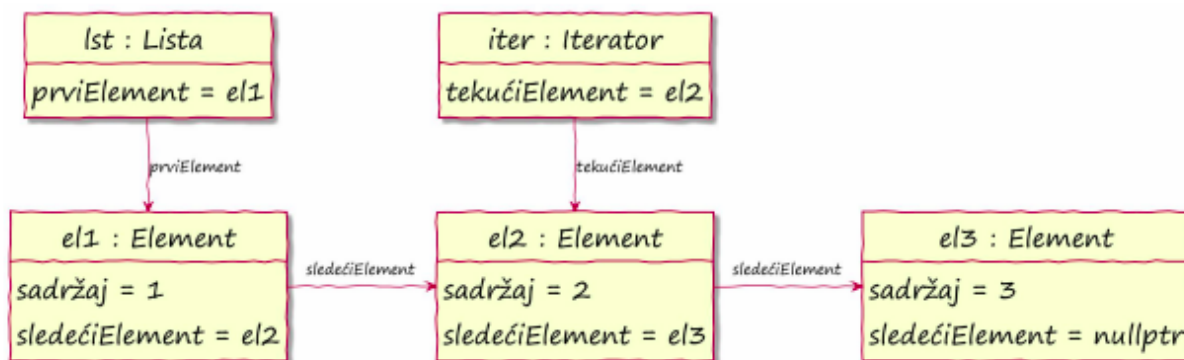
84. Koje su sličnosti i razlike između dijagrama klasa i dijagrama komponenti (UML)?

- Dijagram komponenti je na višem nivou apstrakcije od dijagrama klasa. Dijagram komponenti sadrži jednu ili više klasa koje sa međusobnim vezama čine tu komponentu
- Dijagram komponenti ne prikazuje detaljno internu strukturu i interfejsse.
- Oba dijagrama prikazuju veze između komponenti (klasa).

85. Šta je, čemu služi i kada se upotrebljava dijagram objekata (UML)?

Predstavlja objekte i njihove veze u jednom trenutku vremena. Koristi se kao dopuna dijagrama klasa za opisivanje dinamičkih sistema. Sadrži:

- Nazive klasa
- Nazive objekata
- Imena i vrednosti atributa
- Odnose



86. Koji su osnovni elementi dijagrama objekata (UML)?

Pogledati prethodno pitanje.

87. Koje su razlike između dijagrama klasa i dijagrama objekata (UML)?

Dijagram objekata predstavlja primer veze između objekata klasa u nekom fiksnom trenutku, dok dijagram klasa daje opšte veze između klasa. Dijagram objekata ima konkretne vrednosti za atribute, dok dijagram klasa ima samo nazive. Na jednom dijagramu možemo da imamo više objekata jedne klase. Ideja dijagrama objekata je da bude dopuna za dijagram klasa.

88. Šta je, čemu služi i kada se upotrebljava dijagram isporučivanja (UML)?

Dijagram isporučivanja je dijagram koji predstavlja fizičke elemente arhitekture sistema. Sadrži:

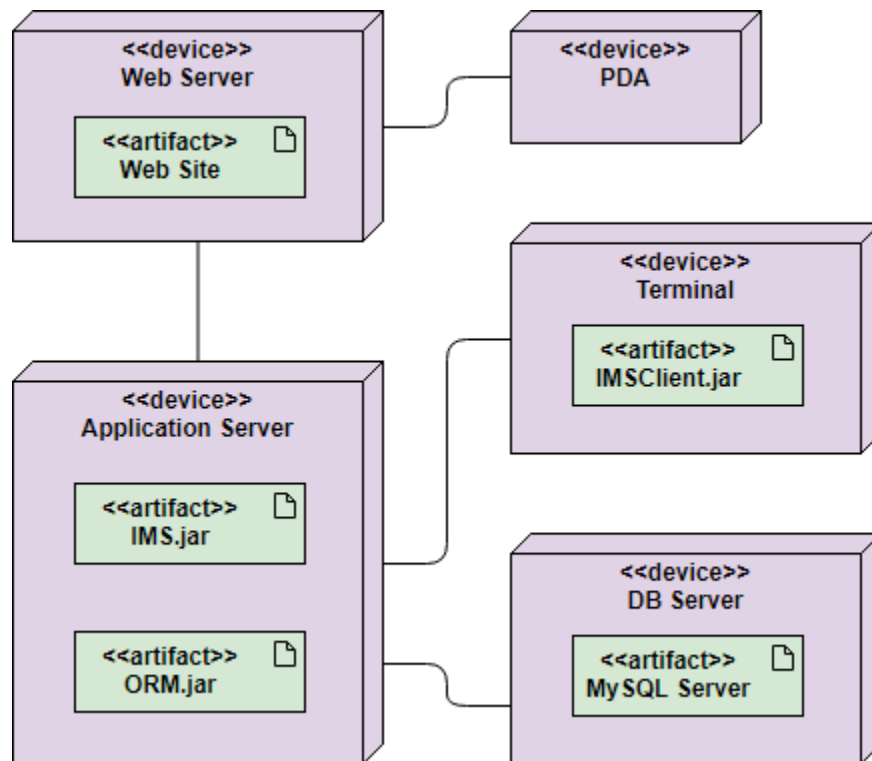
- čvorove (serve)re
- softverske ili hardverske podsisteme
- međusobne veze podsistema
- može da ilustruje i zastupljenost komponenti u podsistemima

Dijagram nam daje informaciju koja softverska komponenta radi na kojoj hardverskoj komponenti.

89. Šta je, čemu služi i koji su osnovni elementi dijagrama isporučivanja (UML)?

Pogledati pitanje 88.

90. Navesti primer dijagrama isporučivanja (UML).

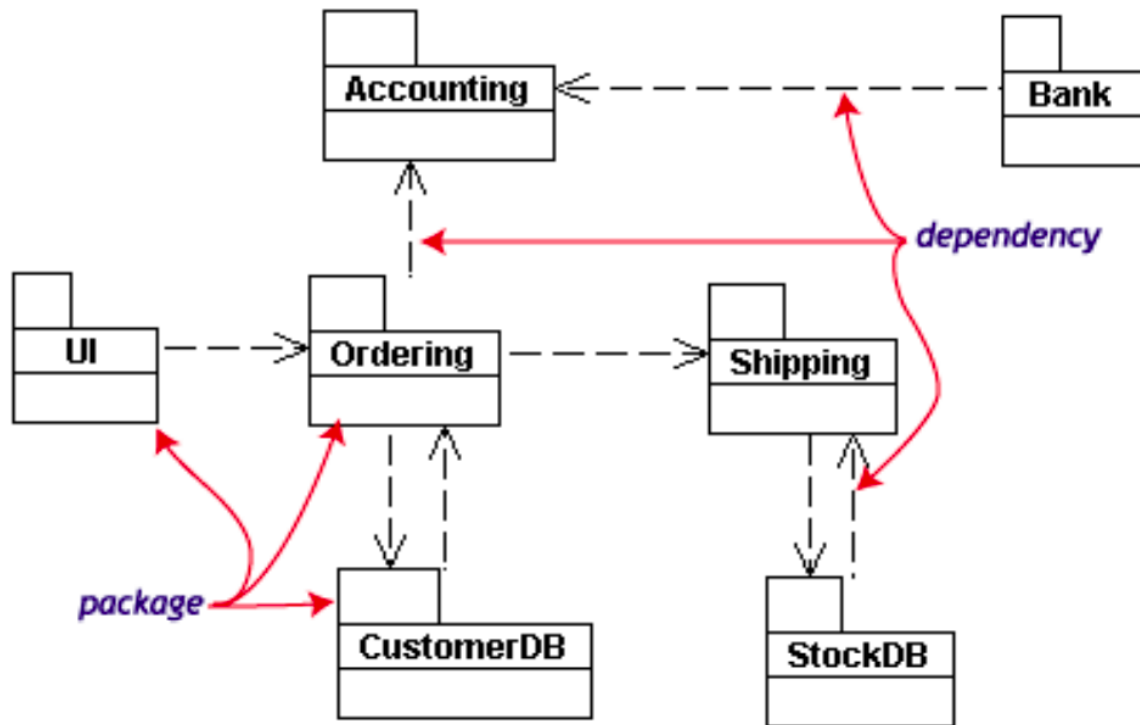


91. Šta je, čemu služi i koji su osnovni elementi dijagrama paketa (UML)?

Dijagram paketa ilustruje kako su elementi logičkog modela organizovani u pakete, kao i međuzavisnosti paketa. Sadrži:

- Nazive i granice entiteta
- Klase u paketima
- Međusobne odnose klasa
- Međusobne zavisnosti paketa
- Može da se koristi i u domenu slučajeva upotrebe

92. Navesti primer dijagrama paketa (UML).



93. Navesti dijagrame ponašanja UML-a.

- Dijagram aktivnosti
- Dijagram stanja
- Dijagram slučajeve upotrebe

94. Šta je dijagram aktivnosti? Navesti i objasniti osnovne elemente dijagrama aktivnosti (UML).

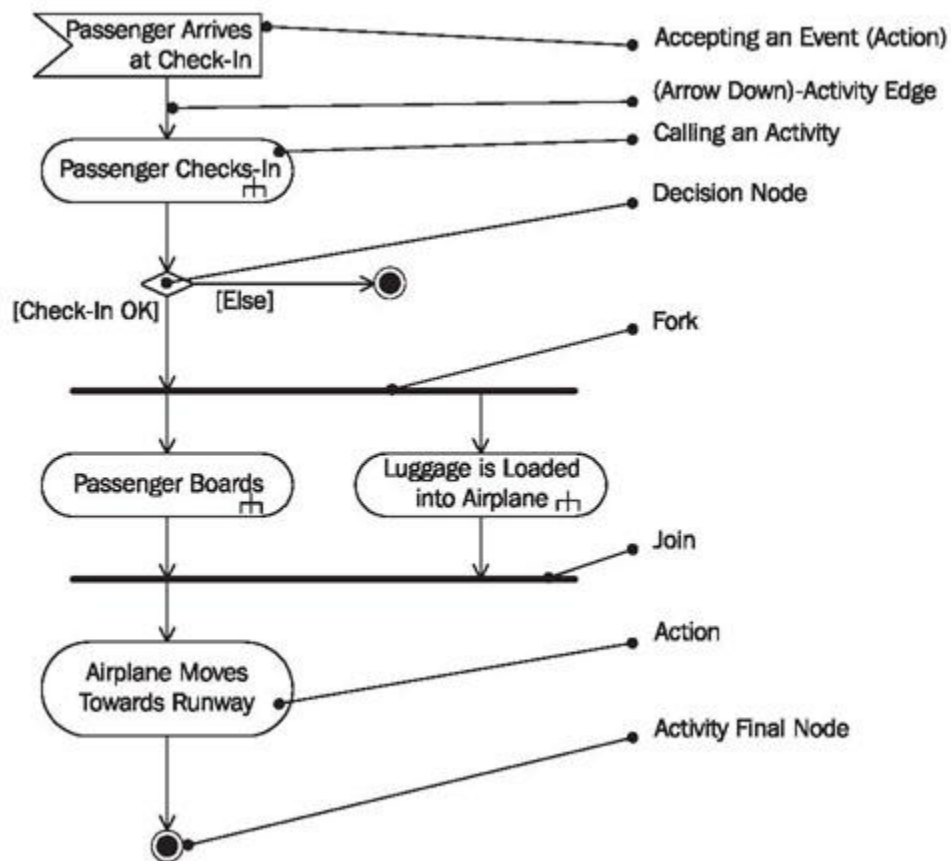
Predstavlja poslovne procese višeg nivoa, tokove podataka i eventualno složene logičke elemente sistema. Sadrži:

- procese
- tokove podataka
- čvorišta i grananja
- uslovne tačke
- početke i završne tačke
- može da sadrži i "linije autora"

95. Koje su sličnosti i razlike između dijagrama aktivnosti (UML) i dijagrama procesa (BPMN)?

BPMN je pogodniji za modeliranje poslovnih procesa u odnosu na dijagram aktivnosti i sadrži nešto bogatiji skup elemenata.

96. Navesti primer dijagrama aktivnosti (UML) koji sadrži sve glavne elemente.

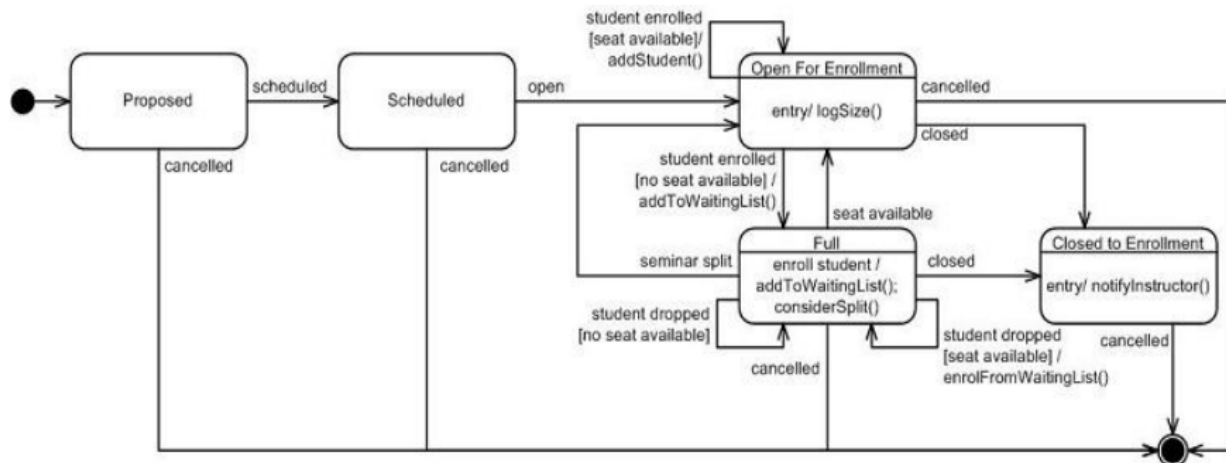


97. Šta je dijagram stanja? Navesti i objasniti osnovne elemente dijagrama stanja (UML).

Dijagram stanja opisuje kako se stanje objekta menja u zavisnosti od interakcija u koje objekat ulazi. Sadrži:

- Početna stanja
- Aktivna stanja
- Završna stanja
- Promene stanja

98. Nacrtati primer dijagrama stanja (UML) koji sadrži sve glavne elemente.



99. Šta je slučaj upotrebe? Kada se upotrebljava i čemu služi?
Kako se opisuje?

Dijagram slučajeva upotrebe je UML tehnika. Predstavlja:

- Skup različitih načina (scenarija) na koje akter može da interaguje sa sistemom;
- U kojima ciljevima naš sistem pomaže akterima da ih ostvare;
- Opseg sistem.

Prilikom crtanja dijagrama slučajeva upotrebe, ne ulazi se u mnogo detalja (pogotovo ne tehničke stvari tj. kako se nešto implementira), već se daje opis sistema i usluga koje on pruža na visokom nivou.

Dijagram slučajeva upotrebe je idealan za:

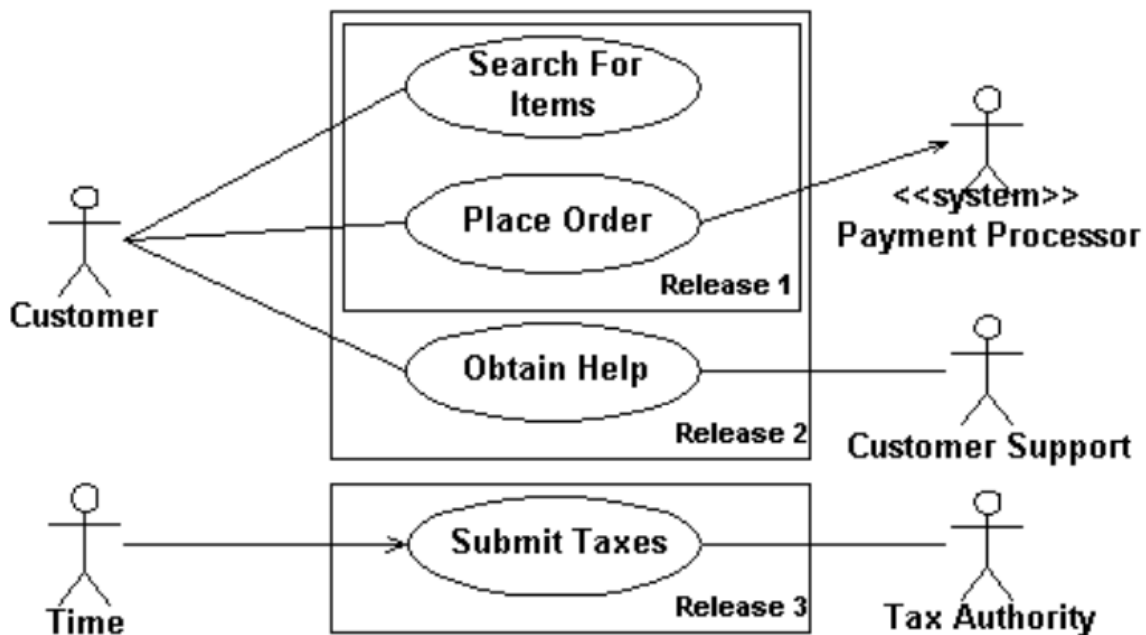
- Prikazivanja ciljeva interakcija sistema i korisnika
- Definisanje funkcionalnih zahteva sistema
- Definisanje konteksta sistema

100. Šta je dijagram slučajeva upotrebe (UML)? Koji su njegovi osnovni elementi?

Osnovni elementi dijagrama slučajeva upotrebe su:

- Slučajevi upotrebe
- Akteri
- Paketi
- Podsystemi

- Međusobni odnosi



101. Šta obuhvata dokumentacija jednog slučaja upotrebe?

Dijagram slučajeva upotrebe predstavlja odnose slučajeva upotrebe i aktera. Pojedini slučajevi upotrebe se opisuju tekstualno. Dobar opis slučaja upotrebe sadrži dovoljno informacija da se može sagledati kada, kako, ko i sa kojim podacima učestvuje u slučaju upotrebe.

Opis slučaja upotrebe može sadržati različite elemente, ali obično ga čine:

- naziv
- akteri
- kratak opis
- preduslovi
- postuslovi
- opis toka slučaja upotrebe
- opis posebnih slučajeva
- dijagrami koji tačnije opisuju slučaj upotrebe
 - dijagrami aktivnosti, dijagrami interakcije, ...

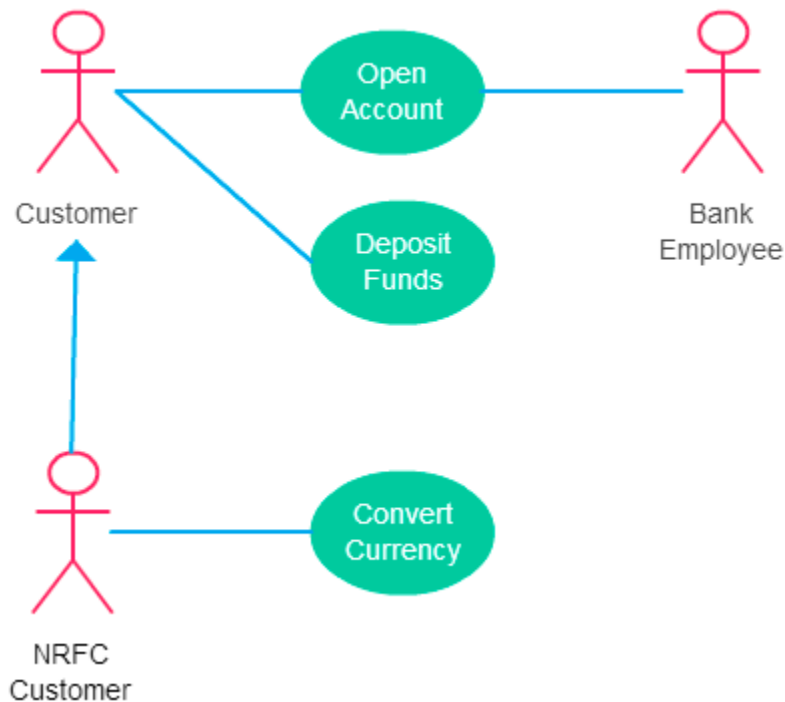
Opciono, dijagrami slučaja upotrebe mogu da sadrže i:

- klasifikacija
 - po vrsti, nivou, ...
- dodatni zahtevi
 - ciljne performanse, ...
- najvažniji rizici u izvođenju

- potencijalno otvorena pitanja
- alternative
- moguća unapređenja
- objašnjenje cilja koji ostvaruje slučajem (ako nije očigledno)
- različiti prilozi

102. Koji su mogući odnosi između aktera (UML)? Kako se predstavljaju? Nacrtati primere.

Između dva aktore je moguća **generalizacija** tj. **nasleđivanje**. Ako A nasleđuje B, onda A učestvuje u svim slučajevima upotrebe kao i B.

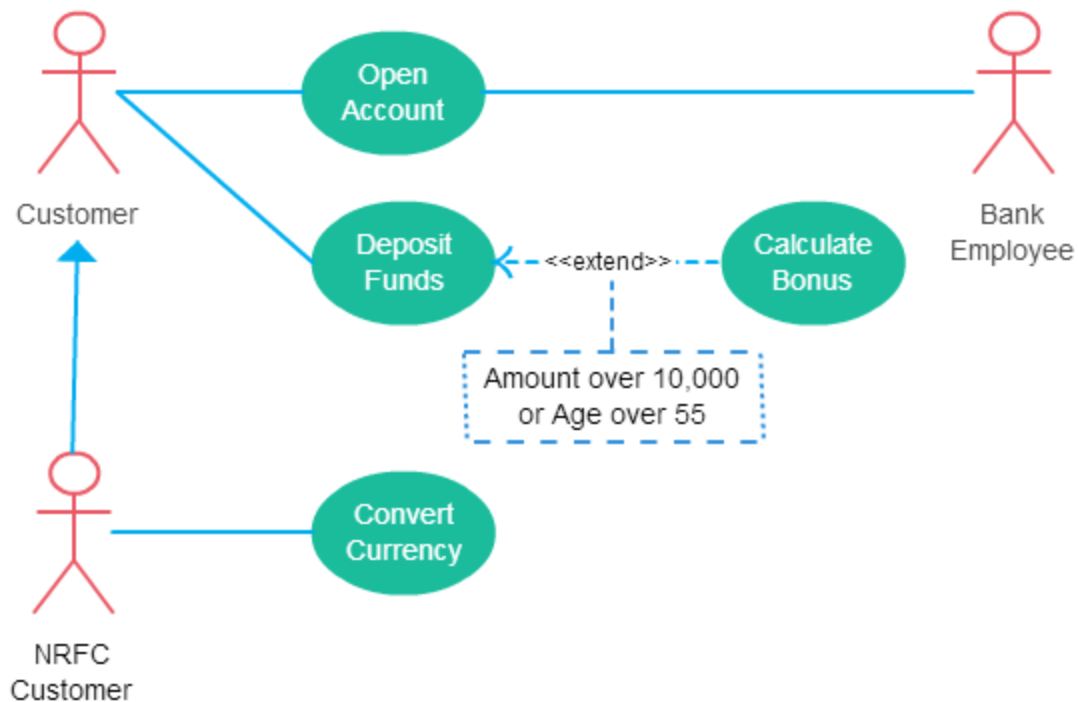


“Customer” učestvuje u “Open Account” i “Deposit Funds”, a “NRFC Customer” učestvuje u “Open Account”, “Deposit Funds” i “Convert Currency”.

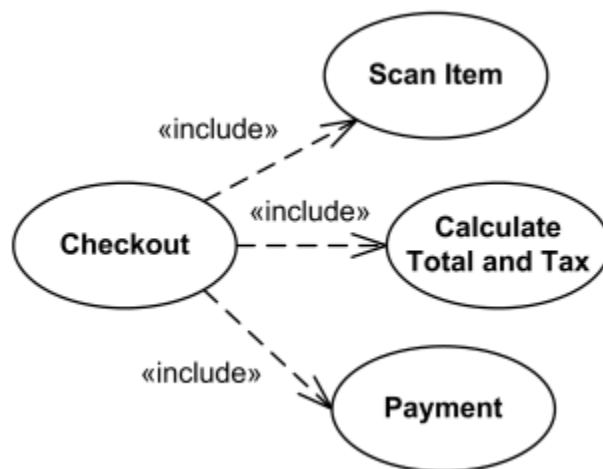
103. Koji su mogući odnosi između aktera (UML)? Kako se predstavljaju? Nacrtati primere.

Između dva slučaja upotrebe su moguće veze:

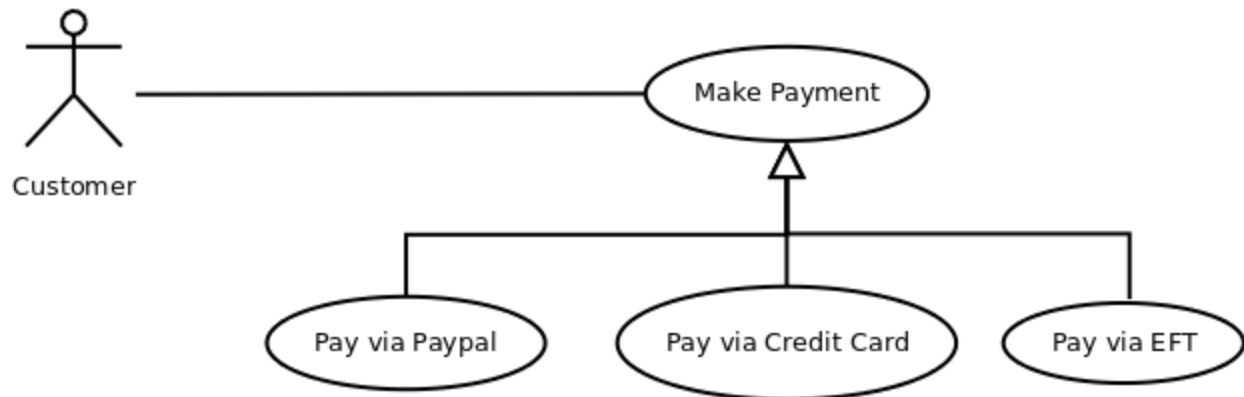
- proširivanje (extend)
 - “A extend B” znači da može opciono da uključi i B, ali ne mora



- obuhvatanje (include)
 - “A include B” znači da je A nepotpun slučaj upotrebe i mora da uključi B



- generalizacija
 - Možemo da definišemo apstrakne (nepotpune) slučajeve upotrebe koji imaju svoj specijalizacija. Primer: Podešavanje (generalizacija), Podešavanje zvuka (specijalizacija), podešavanje tastature (specijalizacija), ...



104. Koji dijagrami spadaju u dijagrame interakcije (UML)?
Navesti i kratko objasniti namenu.

Dijagrami interakcije:

- Dijagram komunikacije
- Dijagram interakcija
- Dijagram sekvence
- Dijagram vremena

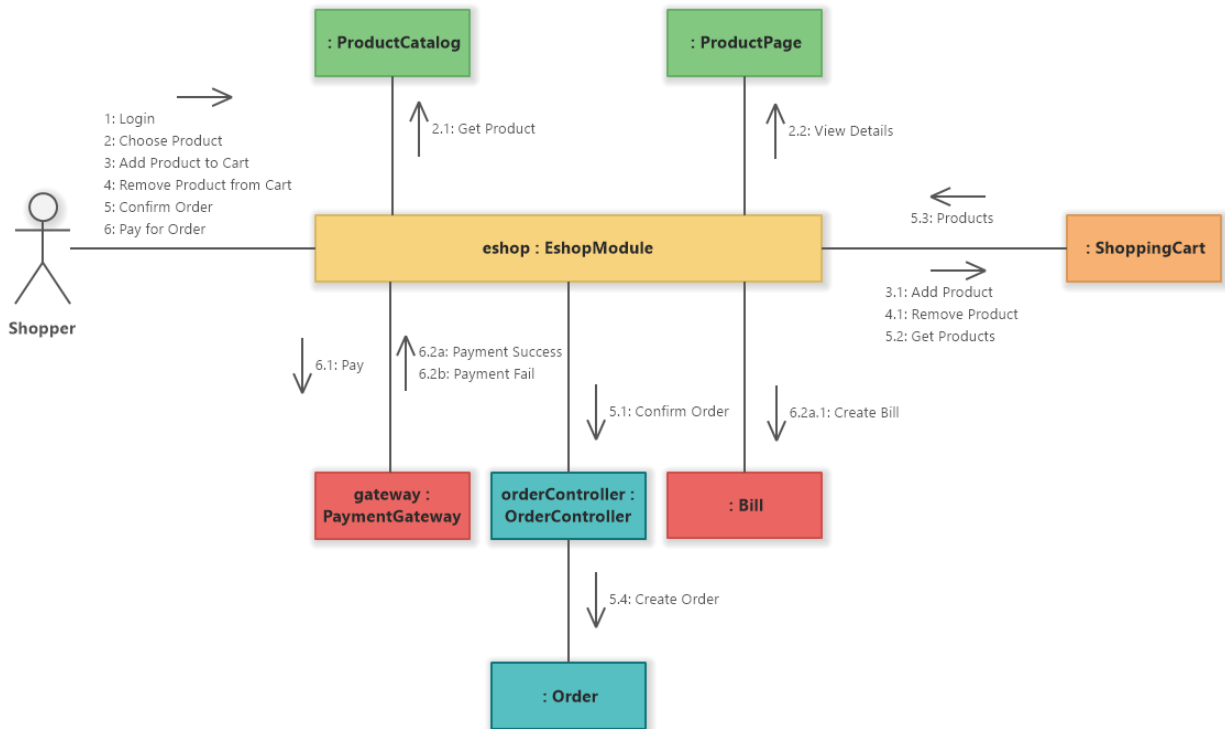
105. Šta je dijagram komunikacije (UML)? Čemu služi i kada se koristi?

Predstavlja objekte, njihove međusobne odnose i poruke koje razmenjuju. Pažnja se po pravilu posvećuje strukturnoj organizaciji objekata koji učestvuju u razmeni poruka.

106. Koji su osnovni elementi dijagrama komunikacije (UML)?
Nacrtati primer.

Dijagram komunikacije sadrži:

- objekte (može i aktere)
- poruke
- komentare i napomene



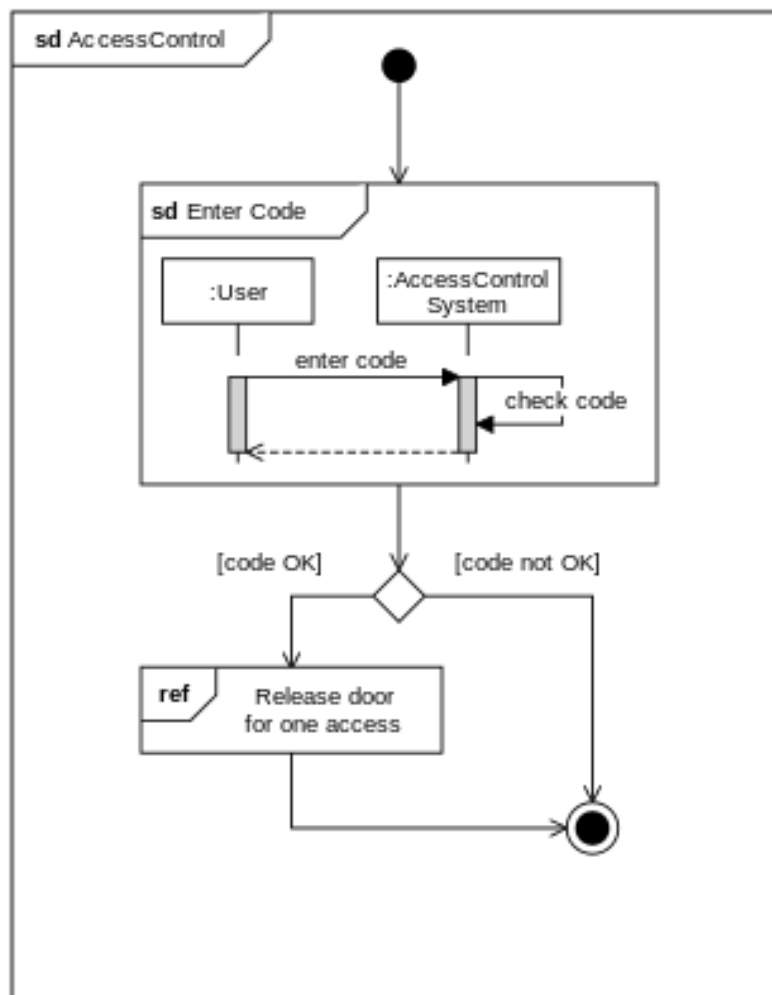
107. Šta je dijagram interakcije (UML)? Čemu služi i kada se koristi?

Dijagram interakcija je varijanta dijagrama aktivnosti u kojoj je akcenat na upravljanju procesima ili sistemom. Svaki čvor/aktivnost u dijagramu može da predstavlja neki drugi dijagram interakcija ili aktivnosti.

108. Koji su osnovni elementi dijagrama interakcije (UML)?
Nacrtati primer.

Dijagram interakcija sadrži:

- objekte
- manje dijagrame aktivnosti ili interakcija
- slučajevne upotrebe
- tok odvijanja procesa
- grananje ili spajanje
- početak ili kraj



109. Šta je dijagram sekvence (UML)? Čemu služi i kada se koristi?

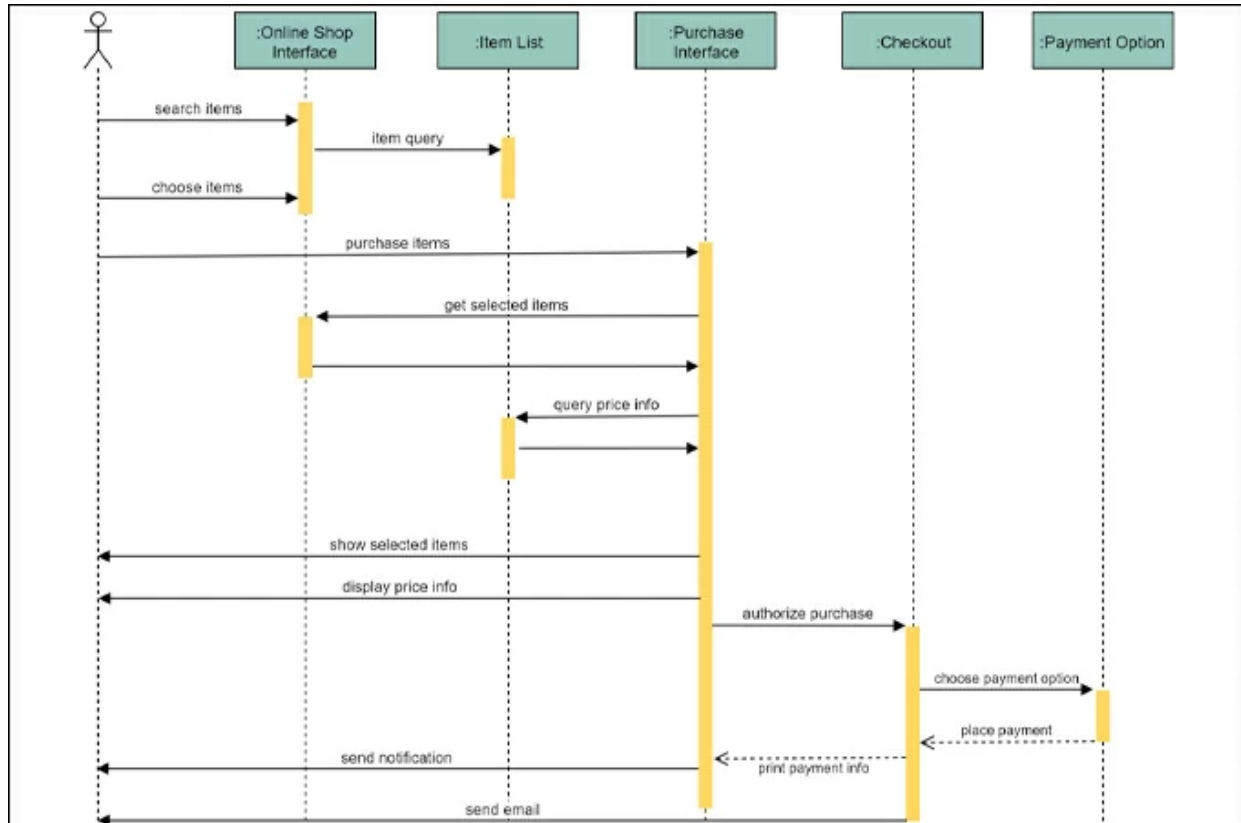
Dijagram sekvence opisuje redosled razmenjivanja poruka tokom odvijanja procesa. Koristi se za detaljniji opis nekog slučaja upotrebe.

110. Koji su osnovni elementi dijagrama sekvence (UML)?
Nacrtati primer.

Dijagram sekvence sadrži:

- objekte (ili učesnike)
 - pravougaone oznake objekata na vrhu dijagrama
 - obično su imenovani, ali privremeni objekti mogu da budu i neimenovani

- vertikalne linije “života” koje pokazuju period aktivnosti objekta u posmatranom procesu
- poruke
 - horizontalne linije koje povezuju linije života objekta
 - pune strelice su sunhrone, a svetle strelice su asinhronne poruke
 - isprekidane linije su poruke koje nose rezultate



111. Objasniti šta su holističke tehnike u okviru razvoja IS.

Holističke tehnike su tehnike koje pokušavaju da sagledaju projekat kao jednu veliku celinu bez posebnog odvajanja procesa od funkcija ili strukture.

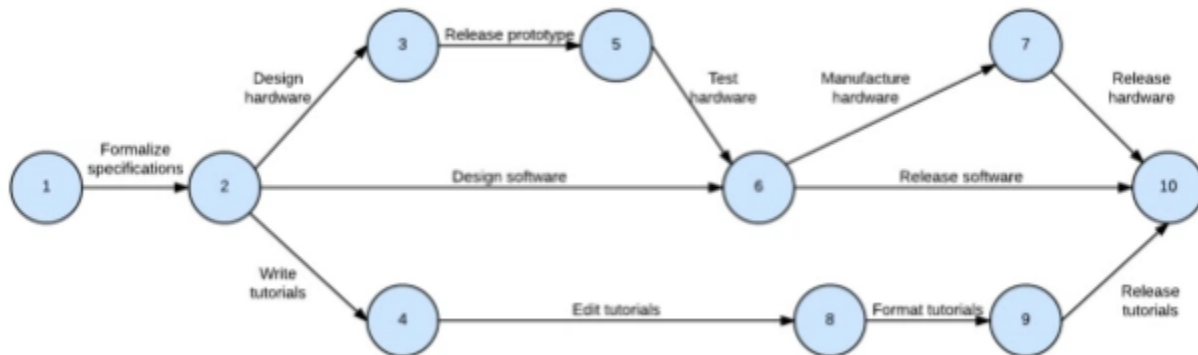
Primer je “Velika slika”:

- Na relativno neformalan način se predstavljaju najvažniji elementi sistema i povezuju se sličicama
- Broj i detaljnost elemenata zavisi od raspoloživog prostora
- ravnomerno se kombinuju različiti elementi (proces, entiteti,

112. Šta su PERT dijagrami? Kada se, zašto i kako koriste?

PERT dijagrami nam služe za vizualizaciju međuzavisnosti pojedinačnih koraka projekta.

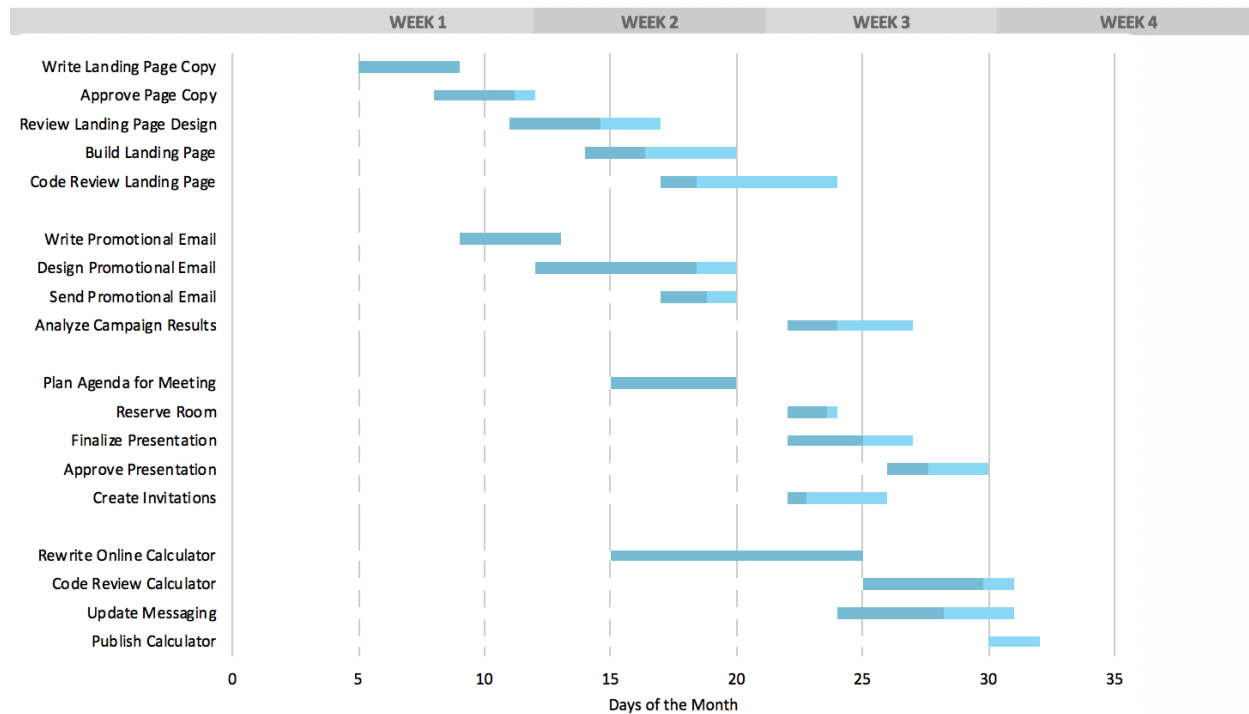
Da bismo nacrtali PERT dijagram, neophodno je da se prvo odrede sve glavne aktivnosti na projektu i onda njihove zavisnosti. Nakon ova dva koraka možemo da nacrtamo acikličan graf zavisnosti. Poslednji korak je definisanje vremenskih okvira svih aktivnosti (rokova) i/ili redosleda zadataka.



113. Šta su GANT dijagrami? Kada se, zašto i kako koriste?

GANT dijagrami nam služe za vizualizaciju svih aktivnosti (vrste) u odnosu na vreme (kolone). Omogućava nam da vidim:

- Koje sve aktivnosti postoje
- Kada svaka aktivnost počinje i kada se završava
- Koliko dugo se očekuje da svaka aktivnost traje
- Gde se aktivnosti preklapaju i koliko
- Očekivani početak i kraj projekta



114. Kako se razvojne metodologije dele prema orijentaciji?
115. Navesti neke procesno orijentisane metodologije.
116. Šta je STRADIS? Navesti osnovne korake (bez opisivanja)?
117. Objasniti korak Inicijalno izučavanje (STRADIS).
118. Objasniti korak Detaljno izučavanje (STRADIS).
119. Objasniti korak Određivanje i projektovanje alternativnih rešenja (STRADIS).
120. Objasniti korak Fizičko projektovanje (STRADIS).
121. Šta je STRADIS? Navesti osnovne korake (bez opisivanja)?
122. Šta je Jurdonov sistemski metod? Navesti osnovne karakteristike.
123. Navesti i ukratko objasniti osnovne korake Jurdonovog sistemskog metoda.
124. Navesti neke kombinovane metodologije.
125. Šta je SSADM? Navesti osnovne karakteristike.
126. Navesti celine i nivoe SSADM.
127. Objasniti celinu Studija izvodljivosti SSADM.
128. Objasniti celinu Analiziranje zahteva SSADM.

129. Objasniti celinu Specifikacija zahteva SSADM.

130. Objasniti celinu Logička specifikacija zahteva SSADM.

131. Objasniti celinu Fizičko projektovanje SSADM.

132. Šta znači pojam „problem“ u kontekstu razvoja IS?

Problem je sve ono što utiče negativno na proces i rezultat razvoja IS-a.

133. Šta znači pojam „neuspeh“ u kontekstu razvoja IS?

Neuspešan razvoj IS može da bude posledica:

- neuspele realizacija samog projekte
- uspešne realizacije, ali projekat ne ispunjava očekivanja (neupotrebljiv rezultat)
- uspešne realizacije i ispunjenih očekivanja, ali sa značajno većim troškovima od planiranih

134. Šta je merilo neuspeha u razvoju IS?

Osnovno merilo neuspeha je izgubljena materijalna vrednost tj. plaćena cena neuspeha:

- uložena sredstva
- izgubljeno vreme
- posledice po čitav poslovni sistem

135. Koje su osnovne vrste neuspeha u razvoju IS?

Osnovne vrste neuspeha su:

- Prekoračenje troškova (sve ostale vrste su usko vezane sa ovom)
- Prekoračenje vremenskih rokova
 - troškovi zbog produženja razvoja
 - troškovi zbog kašnjenja puštanja u rad
- Rezultat nije upotrebljiv (u planiranim razmerama)
 - sistem je implementiran u skladu sa zahtevima, ali ne odgovara stvarnim potrebama
 - neispunjenost nefunkcionalnih zahteva
- Odustajanje od projekta
 - usled nekog od prethodnih razloga (ili više njih)

136. Šta znači pojam „neupotrebljiv rezultat“? Kakvi vidovi neupotrebljivosti postoje?

Čak i kada postoji planiranje i kada se projekat implementira po planu, rezultat može da bude neupotrebljiv ili da predstavlja neuspeh. Aspekti neupotrebljivosti:

- Neupotrebljiv korisnički interfejs
 - loše rešeni ergonomske aspekti (ljudski faktori)
 - nepostojanje fizičkog (hardverskog) odziva
 - neintuitivan izgled korisničkog interfejsa
 - problematične kontrole interfejsa
 - spor odziv
 - nepouzdanost
 - ...
- Procedura korišćenja nije ostvariva / dobra u realnim uslovima

Mogući uzroci:

- Nerealni ili nejasni ciljevi projekta
- Neprecizna procena potrebnih resursa
- Loše definisani zahtevi
- Slaba komunikacija između klijenta, razvijaoa i korisnika

137. Koji su najčešći problemi na strani naručioca IS?

Najčešći problemi na strani naručioca su:

- Nerealni ili nejasni ciljevi projekta
 - Ako razvojni tim prihvati projekat sa nerealnim ciljevima, onda je greška na obe strane
- Neusklađenost ciljeva i strategija
 - Neusklađenost ciljeva i strategija u okviru preduzeća
- Politika ulagača (stakeholder)
 - Svaki informacioni sistem teži da poveća red u sistemu, ali ne vole svi red
 - Informacioni sistem formalizuje procese. Ako nešto nije formalizovano, to znači da tu ima prostora za "zloupotrebu".
- Komercijalna pritisak
 - Forsiranje puštanja sistema ranije u rad
 - Puštanje sistema ranije u rad uglavnom zahteva kompromise, što je veoma rizično u slučaju IS.
- Otpor korisnika prema primeni IS
 - Vezano za politiku ulagača

Sve ove probleme razvojni tim može potencijalno da prepozna.

138. Koji su najčešći problemi na strani razvijalaca IS?

Najčešći problemi na strani razvijalaca:

- Slabo vođenje projekta
- Neprecizna procena potrebnih resursa
- Slabo izveštavanje o stanju projekta
- Neupravljeni rizici
- Upotreba "nezrelih" tehnologija
- Nesposobnost da se iznese složenost projekta
- Tok praktičnog razvoja bez čvrstih principa i pravila

139. Koje su vrste i odlike neuspeha usled „lošeg planiranja“?

Loše planiranje je jedan od najčešćih uzroka neuspeha. Može da ima dva osnovna oblika:

- nedovoljno dobro planiranje
 - nedovoljna analiza problema
 - loše i/ili neprecizno definisane zahteve
 - posledice:
 - nesrazmerno veliki broj naknadnih korekcija zahteva
 - slaba upotrebljivost rešenja
 - probijanje rokova
- preterano planiranje
 - preširoko i nekoncentrisano ulaženje u projekat
 - suviše duboka i obimna analiza sa ranim detaljnim projektovanjem
 - preširoko ulaženje u implementaciju
 - preambicioznost, nesrazmerno realnim mogućnostima
 - posledice:
 - kasno uočavanje propusta
 - otežana tranzicija
 - probijanje rokova

Loše planiranje je često posledica kombinacije prethodna dva oblika umesto samo jednog.

140. Šta je upravljanje rizicima? Šta je cilj i čime se bavi?

Upravljanje rizicima je proces prepoznavanja, procenjivanja i kontrolisana svega onoga što bi u projektu moglo da krene naopako pre nego što postane pretnja uspešnom dovršavanju projekta ili implementacije informacionog sistema.

141. Šta je SCERT?

SCERT (Synergistic Contingency Evaluation and Review Technique) je tehnologija za staranje o rizicima koja se sastoji od četiri koraka:

- Prepoznavanje značajnih aspekata, odgovarajućih rizika i mogućih odgovora.

- strukturiranje rizika i odgovora
 - prepoznavanje opštih i specifičnih odgovara
 - prepoznavanje pravila odlučivanja
 - rezultat je lista rizika i odgovara
- prepoznavanje parametara po kojim se prepoznaje ishod, odgovarajući scenariji i verovatnoća
 - parametri uključuju bar vreme, novac i bezbednost
- interpretacije parametri i rukovanje rizicima
 - procena nastupanja rizika tokom samog procesa
 - reagovanje na ostvarene rizike

142. Koji su osnovni elementi upravljanja rizicima?

Osnovni elementi su:

- procenjivanje rizika
 - Obuhvata sve poslove koji se odrađuju pre nego što rizik nastupi
- kontrola rizika
 - Realno praćenje rizika tokom razvoja

143. Koje aktivnosti obuhvata procenjivanje rizika?

Procenjivanje rizika obuhvata sve one poslove koji se odrađuju pre nego što rizik nastupi.

Aktivnosti:

- prepoznavanje rizika
 - pravljenje liste proveru
 - ispitivanje pravila i pokretača odlučivanja
 - poređenje sa iskustvom (analiza pretpostavki)
 - dekompozicija
- analiza rizika
 - procena verovatnoće i iznosa rizika
 - modeli performansi i troškova
 - mrežna analiza (uslovljenosti i uticaja)
 - statistička analiza odlučivanja
 - analiza faktora zahteva
- dodeljivanje prioriteta rizicima
 - analiza izloženosti
 - analiza uticaja
 - redukcija povezanosti rizika

144. Koje aktivnosti obuhvata kontrola rizika?

Kontrola rizika obuhvata sve one poslove koji nam omogućavaju da izađemo na kraj sa rizicima tj. konkretno rešavanje problema. Aktivnosti:

- planiranje upravljanja rizicima - priprema za suočavanje rizicima

- kupovina informacija
- izbegavanje rizika
- prebacivanje rizika
- umanjenje rizika
- pojedinačno planiranje
- integracija u opšti plan rizika
- razrešavanje rizika
 - prototipovi
 - simulacije
 - testiranje
 - analiziranje
 - upravljanje kadrovima
- praćenje rizika
 - praćenje progressa po tačkama (milestones)
 - detaljno praćenje nekoliko najvažnijih rizika
 - ponavljanje procena rizika
 - popravljavanje uslova i eventualnih posledica

145. Navesti bar 8 od 10 najčešćih izvora rizika u razvoju IS?

Česti izvori rizika i tehnike upravljanja tih rizika:

- manjak osoblja
 - popuna kadrova
 - odabir kadrova
 - unakrsna obuka
 - izgradnja morala
 - preraspoređivanje ključnih ljudi
- nerealni rokovi i budžeti
 - detaljno procenjivanje troškova iz više izvora
 - projektovanje prema troškovima / inkrementalni razvoj
 - pročišćivanje zahteva
- preterivanje ("pozlaćivanje")
 - pročišćivanje zahteva
 - izrada prototipova
 - analiza isplativnosti
 - projektovanje prema troškovima
- neprekidni niz izmena u zahtevima
 - postavljanje praga izmena
 - skrivanje informacija
 - inkrementalni razvoj
- slabosti eksterno realizovanih poslova
 - provera referenci izvođača
 - ugovori o kaznenim merama
 - raspisivanje nadmetanja za posao

- slabosti u eksterno nabavljenim komponentama
 - referentni testovi
 - inspekcije
 - provere referenci
 - analiza kompatibilnosti
 - rano testiranje
- slabe performanse u realnom radu
 - referentni testovi
 - simulacija, pravljenje prototipova
 - upravljanje i korigovanje postupaka
- rad na granicama računarskih nauka
 - tehnička analiza
 - analiza isplativnosti
 - provera referenci
 - pravljenje prototipova

146. Kako se izračunava procena izloženosti riziku? Navesti primer sa neka dva rizika.

Za svaki nezadovoljavajući ishod $NI[i]$ se određuje:

- verovatnoća nastupanja tog ishoda: $P(NI[i])$
- gubitak usled nastupanja: $G(NI[i])$

Onda se izloženost riziku računa kao: $\sum_{i=1, n} P(NI[i]) * G * NI[i]$. Jasno je da procena izloženosti riziku na jednom $NI[i]$ nema mnogo smisla, jer je trošak ili 0 ili $G(NI[i])$. U slučaju da imamo veliki skup nezadovoljavajućih ishoda, onda ima smisla da se računa suma izloženosti riziku tj. očekivani trošak.

Mana ovog pristupa je što za mali skup ne može da radi, a za veliki skup ima velika odstupanja. Postoje mnoge tehnike koje pokušavaju da poprave tačnost ove formule.

147. Šta je dijagram zavisnosti poslova? Kada se pravi i čemu služi?

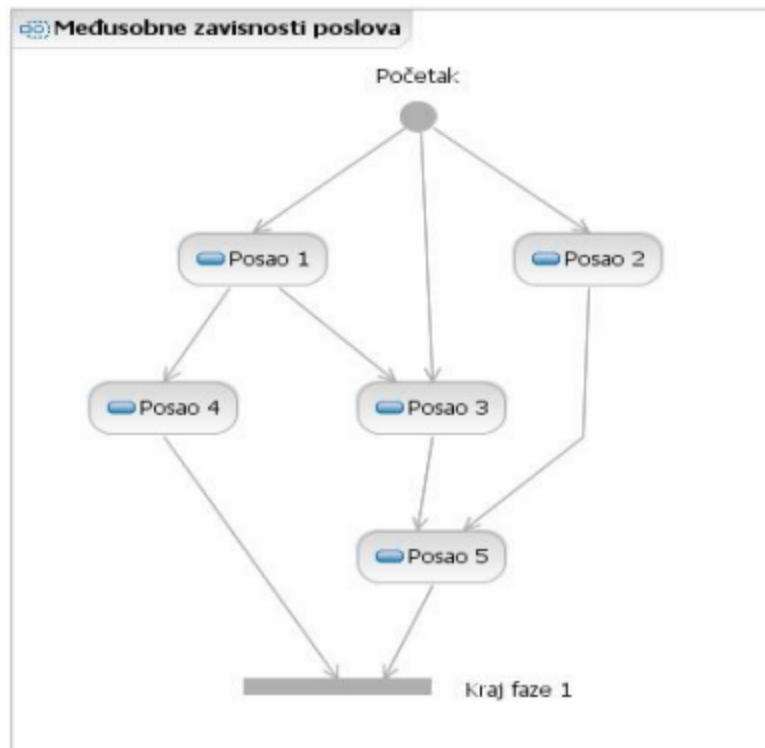
Tokom razvoja postupak se deli na manje poslove:

- Za svaki od njih se pravi posebna procena rizika
- Neuspeh jednog od poslova može da se preslika na druge
 - kašnjenje
 - odlaganje
 - odustajanje
 - promena zahteva

Zbog toga se često prave dijagrami zavisnosti poslova. Za svaki posao se vodi evidencija rizika, rokova i troškova. Na taj način se može lakše sagledati i izračunati kako rizici u nekom od

poslova utiču na rizike u drugom, pa i u čitavom projektu. Dijagrami zavisnosti poslova su slični Pert i Gant dijagramima.

148. Nacrtati primer dijagrama zavisnosti poslova. Objasniti.



149. Kojim pristupima se utiče na smanjenje rizika u savremenim razvojnim metodologijama?

Savremene razvojne tehnologije počivaju na konceptima koji omogućavaju smanjenje rizika.

150. Kojim savremenim konceptima razvoja se pokušava smanjiti rizik nastupanja neuspeha?

Neki od najvažnijih savremenih koncepata razvoja IS koji značajno smanjuju rizike su:

- Inkrementalni razvoj
- Određivanje koraka prema rokovima
- Pojačana komunikacija prema subjektima
- U žiži su objekti, a ne procesi
- Pravljenje prototipova

151. Šta su osnovne odlike koncepta inkrementalnog razvoja? Kvaliteti i slabosti?

Princip:

- Umesto da se razvoju pristupa kao velikom celovitom poslu, sa složenim projektima i implementacijom, posao se deli u niz inkrementalnih faza posvećenim manjim celinama posla.

Dobit:

- Svaka od faza je značajno manja i jednostavnija nego čitav posao, čime se pojednostavljaju planiranje i implementacija
- Veća tačnost planiranja troškova i rokova, i tačnije izveštavanje o toku razvoja

Rizik:

- Ako se u prvim koracima potpuno zanemare predstojeći, postoji rizik da će u narednim koracima biti potrebne veće izmene
- Ako se u prvim koracima uzmu u obzir svi predstojeći, postoji rizik da se njihova složenost priliži složenosti čitavog sistema ("naduvavanje koraka"), čime ovakav pristup gubi smisao
- Često nije moguće sagledati unapred broj, cenu i ukupno trajanje svih koraka

152. Šta su osnovne odlike koncepta određivanja koraka prema rokovima? Kvaliteti i slabosti?

Koncept "Određivanje koraka prema rokovima" se oslanja na inkrementalni razvoj i nema smisla bez njega, ali utiče na to kako razvijamo pojedinačne inkremente. Klasičan pristup inkrementalnom razvoju podrazumeva da odredimo koje sve elemente treba da implementiramo u prvoj fazi i da to uradimo. Drugi pristup je da se odredi koji svi zadaci mogu da se uradi za dati period (do roka). Ovaj pristup je obrnut u smislu da se prvo postavljaju rokovi, pa se onda biraju elementi koji se implementiraju.

Princip:

- Za svaki inkrementalni korak se prvo odrede budžet i rokovi, a tek posle toga poslovi koji će korakom biti obuhvaćeni.

Dobit:

- Drastično smanjivanje rizika od prekoračivanja budžeta ili rokova
- Uspostavljanje ritma redovnog isporučivanja novih verzija sistema
 - Redovnija kontrola kvaliteta i viši stepen pomeranja između subjekata
 - Postepeno prilagođavanje korisnika novim elementima sistema

Rizik:

- Uspostavljanje tesnih okvira inkrementalnog koraka može otežati pojedine korake i podići ukupnu cenu i trajanje razvoja
 - Neki elementi sistema se ne mogu prirodno podeliti u različite korake
 - Svaki korak zahteva troškove isporučivanja što u zbiru može da postane velika stavka u slučaju malih koraka

- U prvim koracima se obično biraju poslovi koji donose veću dobit. Pri kraju razvoja postoji rizik da se ne implementiraju poslovi “čija je cena veća od dobiti” iako su značajni za sistem kao celinu

153. Šta su osnovne odlike koncepta pojačane komunikacije među subjektima IS? Kvaliteti i slabosti?

Princip:

- U planiranju IS (i svakog pojedinačnog koraka) uključuju se u razmatranje sve vrste subjekata u što većem broju

Dobit:

- Dobija se tačnija slika o potrebnim ciljevima iz različitih uglova
- Smanjuje se rizik razvoja neupotrebljivog rešenja
- Subjekti se kroz proces razvoja pripremaju za upotrebu

Rizik:

- Previše informacija može dovesti do preteranog planiranja, a time i do “naduvavanja” pojedinačnih koraka
- Prevelikim razmatranjem mišljenja subjekata koji su navikli na postojeće procese i ne sagledavaju planirane izmene može se smanjiti obim suštinskih funkcionalnih izmena u okviru koraka, a time i nepotrebno povećati broj koraka da se dođe do “konačnog” rešenja

154. Šta su osnovne odlike koncepta stavljanja objekata u središte pažnje umesto procesa? Kvaliteti i slabosti?

Procesi imaju veću tendenciju nastajanja i nestajanja u odnosu na objekte. Takođe, lepa osobina objekata je enkapsulacija tj. ako se lepo modeliraju, promena procesa koja obuhvata neki objekat, ne mora da utiče i na druge objekte sistema.

Princip:

- U središte pažnje se pri razvoju stavljaju objekti, a ne procesi

Dobit:

- Objekti su obično stabilniji od procesa
- Takav pristup je prilagođeniji inkrementalnom razvoju
- Smanjuje se rizik od pogrešnih odluka u ranim fazama razvoja
 - u ranim koracima se više pažnje posvećuje objektima
 - u kasnijim koracima se više pažnje posvećuje procesima

Rizik:

- Potpuno zanemarivanje procesa u ranim fazama razvoja može voditi ka pogrešnoj arhitekturi sistema, što je kasnije veoma skupo za promenu (npr. zanemarivanje nekog procesa koji čini osnovu celog sistema)
- Sasvim detaljno razmatranje procesa u ranim koracima pretilo da dovede do “naduvavanja” prvih koraka

155. Šta su osnovne odlike koncepta pravljenja prototipova? Kvaliteti i slabosti?

Princip:

- U okviru analize problema se pravi prototip koji odražava način funkcionisanja IS

Dobit:

- Olakšava se ne-tehničkim subjektima da u ranim fazama razvoja uoče određene nedostatke
- Smanjuje se rizik od pogrešnih odluka u ranim fazama razvoja

Rizik:

- Prototipovi obično odražavaju funkcionalne aspekte i elemente korisničkog interfejsa, ali ne i unutrašnju strukturu IS. Posledica je da samo delimično odgovaraju OO metodologijama
- Nedovoljno široko napravljen prototip i nedovoljno široko razmatranje prototipova mogu da prikriju nedostatke u drugim aspektima IS
- Previše pažnje posvećene prototipu pretili da "naduva" fazu njegove izrade

156. U kojim okolnostima se razvijaju OO metodologije razvoja IS?

Neke od okolnosti:

- Postojanje metodologija koje strukturano pristupaju analizi i opisivanju procesa
- Potreba za strukturanim opisivanjem podataka
- Potreba za opisivanjem entiteta koji menjaju stanja
- Podizanje nivoa apstraktnosti posmatranja elemenata sistema
- Programiranje upravljano događajima
- Vizuelni korisnički interfejsi
- Posvećena modularnost softvera
- Skraćivanje razvojnog ciklusa
- Tranzicija modela
- Višestruka upotrebljivost softvera
- ...

157. Objasniti osnovne karakteristike OO razvojnih metodologija.

Osnovne karakteristike i koncepti su:

- U žižu stavljaju objekte, a ne procese
- Sve OOM se odlikuju skraćivanjem trajanja razvojnih ciklusa
 - RUP (i druge) propisuju inkrementalni razvoj
 - Agilne metodologije propisuju određivanje koraka prema rokovima i troškovima
- Pojačana komunikacija među subjektima u svim fazama razvoja

158. Koje su osnovne karakteristike objektno-orijentisanih tehnika?

Pogledati pitanje 157.

159. Šta je objekat? Navesti dve definicije.

Objekat je u mogućnosti da odgovori na sledeća pitanja:

- Ko sam ja? (identitet)
- Šta ja znam? (struktura)
- Šta ja mogu da uradim? (ponašanje)

160. Šta je klasa? Šta čini klasu?

Klasa je skup objekata. Klasa se sastoji iz naziva, atributa i metoda. Atributi i metodi mogu da budu javni ili privatni, ali se uglavnom teži da atributi budu privatni.

161. Koji su osnovni koncepti objektno-orijentisanih metodologija?

Objektno-orijentisane tehnike:

- enkapsulacija
- interfejs
- polimorfizam
- nasleđivanje
 - specijalizacija i generalizacija
 - hijerarhija klasa

162. Objasniti koncept enkapsulacije kod OOM.

Struktura objekata je njihova interna stvar

- ne sme se izlagati spoljašnjem svetu
- atributima se sme pristupati samo posredno

Postoje slučajevi kada neki atributi predstavljaju osnovu ponašanja objekata ili klasa

- tada im se može omogućiti javni pristup
- i tada je ipak preporučeno enkapsuliranje

Svrha

- apstrahovanje strukture metodama
- viši nivo međusobne nezavisnosti klase od modula u kojima se upotrebljava

163. Objasniti koncept interfejsa kod OOM.

Objekat (klasa) pruža spoljašnjim korisnicima samo skup metoda kojim mogu komunicirati sa njim. Takav pristup metoda se naziva javan interfejs. Interfejs se oblikuje tako da omogući obavljanje jednog celovitog posla.

Svrha

- Suština objekta je u njihovom ponašanju. Interfejs omogućava to ponašanje.
- Sužavanjem interfejsa na suštinu funkcionisanja objekata prikriva se sva složenost implementacije i pruža se viši nivo nezavisnosti objekata od okruženja.

Napomena:

- Ako objekat ima više interfejsa, znači da ima više funkcija, pa je potrebno razmotriti njegovo razlaganje na više objekata.

164. Objasniti koncept polimorfizma kod OOM.

Polimorfizam podrazumeva da se jednom napisan kod može upotrebljavati za različite vrste objekata. Postoje tri vrste polimorfizma:

- hijerarhijski
- parametarski
- implicitni

Svi OOP jezici podrazumevaju hijerarhijski polimorfizam i većina (savremenih) podrazumeva i parametarski.

165. Objasniti koncept nasleđivanja kod OOM.

Nasleđivanje klase je ekvivalentno uvođenju jednosmerne parcijalno uređene relacije "jeste" između klasa:

- Klasa A "jeste" klasa B akko svaki objekat klase A ima sve osobine koje imaju objekti klase B

Svrha:

- Nasleđivanje se koristi za eksplicitno označavanje sličnosti među klasama (objektima)
- Predstavlja osnovu za hijerarhijski polimorfizam:
 - Ako se nešto može uraditi sa objektom klase B, onda se to može uraditi i sa objektom klase A koja je izvedena iz B

166. Koji se periodi mogu prepoznati u razvoju OO metodologija? Objasniti.

Periodi:

- Početni koraci (-1997)
 - veliki broj različitih notacija i metodologija

- nijedna kompletna i dovoljna široka
- akcenat je na razvoju metoda i tehnika
- Oblikovanje UML-a (1995-2005)
 - nekoliko dubljih metodologija koncentrisanih na različite faze razvoja
 - pokušaji objedinjavanja metoda i tehnika u ujednačene notacije
 - akcenat na notaciji
- Post-UML koraci (2000-)
 - ujednačena notacija
 - široko shvatanje procesa razvoja
 - potpuno posvećivanje metodologijama
 - RUP i druge savremene metodologije

167. Šta znači skraćenica OOAD?

OOAD - Objektno orijentisani analiza i dizajn. U centru pažnje su objekti koji figurišu u sistemu:

- entiteti
 - sve vrste podataka koji postoje u sistemu
- subjekti
 - različite vrste korisnika sistema
- servisi
 - usluge koje sistem pruža korisnicima
- interfejsi
 - podsistemi putem kojih subjekti koriste usluge

168. Koji je aspekt IS je primaran u slučaju OOAD?

Pogledati pitanje 167.

169. Koji su osnovni postupci u OOAD (Coad & Yourdon)?

Osnovni postupci:

- Pronalaženje klasa i objekata
- Prepoznavanje struktura
- Prepoznavanja subjekata
- Određivanje atributa
- Određivanje usluga

170. Kako se pronalaze klase u OOAD?

Neophodno je da se:

- Analizira domen
 - Ne razlikuje se mnogo od projektovanja baza podataka tj. EER modela
 - Svaka imenica je kandidat za klasu ili objekat
 - Svaka imenica je objekat, ali ne mora da bude i klasa

- Ako je objekat samo podatak, onda ne mora da bude klasa
- Primer: Imenica koja se odnosi na neki ceo broj
- Svaki glagol predstavlja operaciju koju obavlja neki objekat u saradnji sa nekim drugim (ili sam)
- Prepoznaju se stabilne klase i objekti
- Oni će predstavljati jezgro budućeg sistema i aplikacija

Posmatraju se i traže:

- strukture
- drugi sistemi sa kojima posmatrani sistem sarađuje ili razmenjuje informacije
- uređaji koji se koriste u sistemu
- događaji koji se moraju pamtit i zapisivati
- fizička i geografska lociranost, organizacione jedinice, sastav timova

Značajni aspekti pri oblikovanju klasa:

- sve što je potrebno pamtit i čini potencijalne atribute
- potrebno ponašanje čini potencijalne metode
- objekti obično imaju više od jednog atributa, ako postoji samo jedan, objekat se dovodi u pitanje
- klase obično imaju više od jednog objekta, ako postoji samo jedan, klasa se dovodi u pitanje
- atributi moraju da se odnose na sve objekte klase
- ponašanja mora da se odnosi na sve objekte
- svi aspekti atributa i ponašanja moraju da potiču iz domena problema, a ne iz domena implementacije
- izbegavanje redundantnosti i izvedenih atributa

171. Šta su elementi strukture u OOAD? Kako se ustanovljavaju?

Na osnovu analize klasa i objekata utvrđuju se njihovi međusobni odnosi:

- generalizacija / specijalizacija
- asocijacija
- kompozicija / agregacija

172. Kako se prepoznaju subjekti (OOAD)?

Cilj prepoznavanja subjekata je segmentacija sistema na podsisteme po kriterijumu korisnika (cilj ima sličnosti sa DTP)

Ovo je posebno značajno u velikim sistemima sa velikim brojem usluga i/ili klasa podataka.

173. Kako se prepoznaju atributi (OOAD)?

Određivanje atributa klasa i objekata se ne razlikuje mnogo od odgovarajućeg postupka u tehnici modeliranja entiteta i odnosa.

Nazivi atributa moraju da budu u skladu sa rečnikom podataka. Za svaki atribut se određuje tip i dodaje opis.

174. Kako se određuju usluge (OOAD)?

Za razliku od strukturnih metodologija, u OO se puna pažnja uslugama (postupcima, procesima) posvećuje na kraju.

Servisi odgovaraju "atomičnim" celinama poslova, koji bi trebalo da u praksi odvijaju uvek u potpunosti.

Obično se prepoznavaju dve vrste usluga:

- računarske usluge
- usluge praćenja i reagovanja na događaje
 - atomsko
 - manuelno

175. Šta je RUP?

RUP je metodologija razvoja. Osnovni elementi:

- Skup postulata i principa na kojima se zasniva razvoj
- Razvojno okruženje i skup metoda
- Jezik za opisivanje procesa i modela

RUP je pristup razvoju softvera koji počiva na:

- iterativnom ciklusu
- stavljanju arhitekture u središte procesa
- upravljanje prema slučajevima upotrebe

Glavna razlika između agilnih metodologija i RUP je u tome što se od početka ima u vidu arhitektura. To je pogodno za velike projekte, gde je promena arhitekture veoma skupa.

Pojmovi:

- postulat (filozofija) - konceptualna zamisao i okosnica na kojoj počiva RUP
- princip - pravilo ili skup pravila čije poštovanje se pretpostavlja u okviru RUP-a
- posao (task) - vrsta aktivnosti koju obavlja pojedinac ili tim u okviru RUP-a
- disciplina - skup srodnih poslova koje može da obavlja pojedinac ili tim odgovarajuće stručnosti

- rezultat rada (artefakt) - pojedinačni rezultat rada koji nastaje kao posledica obavljanja jednog ili više poslova u okviru metodologije

176. Koji postulati ilustruju filozofiju RUP-a? Navesti bar 5. Objasniti?

Osnovni postulati:

- Prilagođavanje procesa
- Usklađivanje sukobljenih prioriteta
- Saradnja celokupnog razvojnog tima
- Iterativno predstavljanje rezultata
- Podizanje nivoa apstrakcije
- Stalno posvećivanje kvalitetu

177. Šta podrazumeva postulat „prilagođavanja procesa“ RUP-a?

Koncepti:

- Složenost procesa mora odgovarati složenosti projekta i karakteristikama razvojnog tima
- Formalnost elemenata procesa mora odgovarati konkretnim fazama

Dobit:

- Efikasan razvojni proces
- Povećana agilnost projekta
- Realno planiranje i kompetentne procene

178. Šta podrazumeva postulat „usklađivanja sukobljenih interesa“ RUP-a?

Koncepti:

- Razumevanje poslovnih prioriteta
- Određivanje prioriteta projekta i zahteva
- Uparivanje potreba sa mogućnostima softvera
- Razumevanje kojim se rezultatima može upravljati
- Usklađivanje ponovne upotrebe rezultata sa potrebama

Dobit:

- Usklađivanje aplikacija sa poslovnim i korisničkim potrebama
- Smanjivanje zastupljenosti specifičnog razvoja
- Optimizacija poslovne vrednosti

179. Šta podrazumeva postulat „saradnje celokupnog razvojnog tima“ RUP-a?

Koncepti:

- Motivisanje osoblja za puno angažovanje
- Pravljenje samostalnih timova
- Podsticanje mešovityh timova (npr. analitičar, programer, tester, ...)
- Obezbeđivanje efikasnog okruženja za saradnju
- Upravljanje proizvodima rada (artefaktima) u cilju podsticanja saradnje
- Integrisanje poslovnih, softverskih i izvršnih timova

Dobit:

- Produktivnost timova
- Veće slaganje poslovnih potreba i procesa razvoja

180. Šta podrazumeva postulat „iterativnog predstavljanja rezultata“ RUPa?

Koncepti:

- Omogućavanje povratnih informacija inkrementalnim isporučivanjem novih rezultata korisnicima
- Prilagođavanje planova razvoja inkrementalnom procesu i dobijenim odgovorima korisnika
- Uvažavanje izmena i upravljanje izmenama
- Suočavanje sa najvećim tehničkim, poslovnim i operativnim rizicima u ranim fazama

Dobit:

- Rano smanjenje rizika
- Visoka predvidivost
- Poverenje među zainteresovanim stranama

181. Šta podrazumeva postulat „podizanja nivoa apstrakcije“ RUP-a?

Koncepti:

- Ponovljena upotreba postojećih rezultata
- Upotreba alata i jezika višeg nivoa, radi smanjenja obima i složenosti dokumentacije
- Fokusiranje na arhitekturu
- Projektovanje sa akcentom na kvalitet, pouzdanost, razumevanje i upravljanje složenošću

Dobit:

- Produktivnost
- Smanjena složenost

182. Šta podrazumeva postulat „stalnog posvećivanja kvalitetu“ RUP-a?

Koncepti:

- Puna kontrola tima nad kvalitetom

- Puno i kontinualno testiranje, paralelno sa razvojem, integrisanjem i isporučivanjem
- Inkrementalno građenje projekta sa automatizovanim testiranjem

Dobit:

- Viši kvalitet
- Raniji uvid u tok razvoja i kvalitet

183. Koji su osnovni principi RUP-a? Navesti bar 7.

Osnovni principi:

- Vizija
- Plan
- Rizici
- Poslovni slučajevi
- Arhitektura
- Prototip
- Procenjivanje
- Izmena zahteva
- Podrška korisnicima
- Prilagođavanje procesa

184. Šta je „vizija“ (RUP)?

Kada hoćemo nešto da pravimo, potrebno je u par rečenica da objasnimo šta želimo da pravimo.

Formalnije, oblikovanje „vizije“ podrazumeva jasno sagledavanje potreba i očekivanja od projekta:

- pojmovi
- problemi
- ulagači
- karakteristike IS
- funkcionalni zahtevi
- nefunkcionalni zahtevi
- druge pretpostavke za ostvarivanje projekta

Posledica odstupstva vizije:

- Promašena tema ili cilj razvoja, tako da rezultat ne zadovoljava očekivanja

185. Kako izgleda životni ciklus IS po metodologiji RUP?

Osnovu životnog ciklusa projekta čine četiri perioda razvoja:

- Početak (Inception)
- Razrada (Elaboration)
- Konstrukcija (Construction)

- Prelazak (Transition)

Svaki od perioda razvoja:

- počinje po završetku prethodnog
- završava se jasno određenim rezultatom
- na kraju obuhvata procenjivanje uspešnosti i eventualno ponavljanje nekih postupaka
- može biti iterativan

Čitav razvojni ciklus može da se ponavlja

- u istom ili izmenjenom obliku

Pojmovi:

- razvojni ciklus - jedan prolazak kroz četiri perioda
- generacija - rezultat jednog razvojnog ciklusa
- evolucioni ciklus - razvojni ciklus osim prvog (inicijalni ciklus - prvi ciklus)
 - obično su im kraći periodi Početak i razrada

186. Navesti i kratko objasniti periode razvoja po RUP-u?

Periodi razvoja RUP-a su:

- Početak: Određivanje šta se pravi
 - naziva se i Uvod i Usvajanje
 - vizija
 - osnovni poslovni slučaj
 - projektni zahtevi visokog nivoa
 - inicijalni slučajevi upotrebe (10-20% dovršeni)
 - upravljanje okvirima projekta
- Razrada: Određivanje kako se pravi
 - detaljni zahtevi (80%)
 - definisanje stabilne i izvršne arhitekture
 - implementiranje nekih ključnih komponenti
 - oko 10% koda implementirano
 - zasnivanje arhitekture na ključnim slučajevima upotrebe
 - oko 20% slučajeva upotrebe čini 80% arhitekture
 - projektovanje, implementacija i testiranje nekih ključnih slučajeva upotrebe
- Konstrukcija: Pravljenje proizvoda
 - dovršavanje zahteva i projektnog modela
 - projektovanje, implementacija i testiranje svake od komponenti
 - upotreba prototipova i uključivanje krajnjih korisnika u proces
 - inkrementalna evolucija izvršnog sistema do gotovog sistema
 - građenje dnevnih ili nedeljnih verzija sa automatizovanim procesom građenja
 - testiranje pri svakom građenju
 - isporučivanje potpuno funkcionalnog softvera (beta)

- Prelazak: Isporučivanje proizvoda krajnjim korisnicima
 - pravljenje inkrementalnih izdanja radi otklanjanja neispravnosti
 - ažuriranje korisničke i instalacione dokumentacije
 - ažuriranje opisa izdanja
 - završne analize

Napomena: Prvi ciklus je karakterističan, jer u periodima moraju da se uzmu u obzir ceo projekat. Ovo se najviše odnosi na periode početak i razradu. Kasniji ciklusu su lakši.

187. Kakav je međusoban odnos perioda razvoja u okviru životnog ciklusa po metodologiji RUP?

Svaki od perioda razvoja počinje završetkom nekog drugog perioda.

188. Čime se odlikuje period životnog ciklusa „usvajanje“ (RUP)?

Pogledati pitanje 186.

189. Čime se odlikuje period životnog ciklusa „razrada“ (RUP)?

Pogledati pitanje 186.

190. Čime se odlikuje period životnog ciklusa „konstrukcija“ (RUP)?

Pogledati pitanje 186.

191. Čime se odlikuje period životnog ciklusa „prelazak“ (RUP)?

Pogledati pitanje 186.

192. Šta je „razvojni ciklus“ (RUP)?

Pogledati pitanje 186.

193. Šta je „generacija“ (RUP)?

Pogledati pitanje 186.

194. Šta je „evolucioni ciklus“ (RUP)?

Pogledati pitanje 186.

195. Navesti okvirnu procenu trajanja svakog od perioda životnog ciklusa RUP.

Trajanje perioda životnog ciklusa RUP-a zavisi od samog projekta, ali okvirno je:

- Početak: 10%
- Razrada: 30%
- Konstrukcija: 50%
- Prelazak: 10%

196. Navesti okvirnu procenu obima poslova za svaki od perioda životnog ciklusa RUP.

Obim poslova perioda životnog ciklusa RUP-a zavisi od samog projekta, ali okvirno je:

- Početak: 5%
- Razrada: 20%
- Konstrukcija: 65%
- Prelazak: 10%

197. U čemu je razlika između životnog ciklusa RUP i spiralnog životnog ciklusa IS?

Glavna razlika je u tome što RUP stavlja veliki akcenat na arhitekturu.

198. Šta je strategija planiranja životnog ciklusa? Navesti (samo nazive) bar 3 strategije planiranja životnog ciklusa (RUP)?

Složenost i obim pojedinačnih perioda razvoja nisu strogo propisani već se mogu prilagoditi prirodi projekta i stvarnim potrebama.

Šabloni planiranja životnog ciklusa se nazivaju strategiji.

Primeri:

- Inkrementalna strategija
- Evoluciona
- Inkrementalno isporučivanje
- "Veliki projekat"
- Hibridne strategije

199. Po čemu se međusobno razlikuju strategije životnog ciklusa RUP? Objasniti na primeru dve strategije.

Strategije životnog ciklusa RUP-a se razlikuju po načinu realizacije samih projekata. Strategije se biraju na osnovu prirode projekta.

200. Čime se odlikuje inkrementalna strategija životnog ciklusa RUP? Kada se primenjuje?

Osnovna ideja:

- Određuju se korisničke potrebe
- Definišu se sistemi zahteva
- Zatim se ostatak razvoja kroz sekvencijalnu izgradnju

Objašnjenje: Od celine projekta mi uzimamo samo deo koji se realizuje. Nakon toga se taj deo popravljala kroz iteracije. Ukoliko ne znamo šta će se raditi tačno u svakom delu, onda nema potrebe da sve razrađujemo, već je dovoljno da razradimo ono što ćemo da radimo.

Pogodna ako je:

- dobro poznat domen problema
- dobro razumevanje rizika
- iskusan razvojni tim

U prevodu, ova strategija je pogodna za projekte gde je poznat domen i otprilike se unapred zna šta treba da se radi.

Karakteristike iteracije:

- Kratka iteracija Početak - definisanje okvira, vizije projekta i poslovnog slučaja
- Jedna iteracija Razrade - definisanje zahteva i uspostavljanje arhitekture
- Više iteracija Konstrukcije (akcenat je ovde) - slučajevi upotrebe se postepeno implementiraju i arhitektura se ostvaruje
- Više iteracija Prelaska - migriranje rezultata u korisničko okruženje

Ne mora strogo jedna iteracija početka i razrade. Poenta je da su početak i razrade u kasnijim iteracijama manje zastupljene ili su integrisane sa konstrukcijom.

201. Čime se odlikuje evolucionarna strategija životnog ciklusa RUP? Kada se primenjuje?

Osnovna ideja:

- Pretpostavlja se da se ne mogu unapred u potpunosti razumeti korisničke potrebe
- Ne definišu se unapred slučajevi upotrebe, već se unapređuju u svakom koraku izgradnje

Objašnjenje: Glavna razlika u razvoju u odnosu na inkrementalnu strategiju je u tome što se ovde implementiraju više komponenti površno, pa one evoluiraju kroz iteracije. Te komponente se razvijaju u svakoj iteraciji. U ovoj strategiji mora od početka da se razmišlja o svemu.

Pogodna je ako:

- nov ili nepoznat domen problema
- neiskusan razvojni tim

U prevodu, donekle suprotno inkrementalnoj strategiji po pogodnosti.

Karakteristike iteracije:

- Kratka iteracija Početak - definisanje okvira i vizije projekta i poslovnog slučaja
- Više iteracija Razrade - postepeno definisanje i unapređivanje zahteva
- Jedna iteracija Konstrukcije - slučajevi upotrebe se implementiraju i arhitektura se proširuje
- Više iteracija Prelaska - Migriranje rezultata u korisničko okruženje

Više iteracija razrade ne znači da se samo planira, a da se ništa pored toga ne radi. Poenta je da je ipak akcenat na razradi.

Evolucionarna strategija teži da bude duža od inkrementalne strategije zbog većeg planiranja i zbog toga što se očekuje veća izmena postojećeg koda (zato što ne znamo unapred kako treba tačno da izgleda sistem)

202. Čime se odlikuje strategija „inkrementalnog isporučivanja“ životnog ciklusa RUP? Kada se primenjuje?

Osnovna ideja:

- U nekim slučajevima je važno rano isporučiti prve rezultate
- Tada se veći deo posla obavlja u iteracijama Prelaska
- Prelazak započinje rano i ima više iteracija
- Ova strategija zahteva veoma stabilnu arhitekturu, što je obično teško ostvariti u malom broju uvodnih iteracija

Ključna razlika je u tome što je akcenat na periodu Prelasku. Samo isporučivanje može da se vrši u kod evolucione strategije i kod inkrementalne strategije.

Pogodna je za:

- Dobro poznat domen problema
 - arhitektura i zahtevi mogu da se stabilizuju rano u razvojnem ciklusu
 - nizak nivo novina u problemu
- Iskusan tim
- Inkrementalno isporučivanje ima visoku vrednost za klijenta

U prevodu, pogodna je strategija za slične slučajeve kao i za inkrementalnu strategiju, ali akcenat je na tranziciji.

Karakteristike iteracije:

- Kratka iteracija Početak - definisanje okvira i vizije projekta i poslovnog slučaja
- Jedna iteracija Razrade - zasnivanje stabilne arhitekture
- Jedna iteracija Konstrukcije - slučajevi upotrebe se implementiraju i arhitektura se ostvaruje
- Više iteracija prelaska - migriranje rezultata u korisničko okruženje

203. Čime se odlikuje strategija „veliki projekat“ životnog ciklusa RUP? Kada se primenjuje?

Osnovna ideja:

- Tradicionalni “pristup vodopada” predstavlja izobličen slučaj sa jednom iteracijom u fazi Konstrukcije
- Obično je posledica veliki broj iteracija u fazi Prelaska

Plan je da se projekat završi u jednom razvojnom ciklusu. Ova strategija se zapravo primenjuje kada projekat nije veliki (suprotno svom imenu). Neophodno je da je razvojni tim dovoljno sposoban za izađe na kraj sa projektom u jednom ciklusu (mali izazov i rizik).

Pogodna je za:

- Dodaje se mali broj dobro definisanih funkcionalnih izmena proizvodu koji je veoma stabilan
- Nove funkcionalnosti su dobro definisane i shvaćene
- Tim je iskusan i u domenu problema i sa postojećim proizvodima

Karakteristike iteracije:

- Kratka iteracija Početak - definisanje okvira i vizije projekta i poslovnog slučaja
- Jedna veoma dugačka iteracija Konstrukcije - slučajevi upotrebe se implementiraju i arhitektura se ostvaruje
- Više iteracija Prelaska - migriranje rezultata u korisničko okruženje

204. Čime se odlikuju hibridne strategije životnog ciklusa RUP? Kada se primenjuju?

Osnovna ideja:

- U praksi retko može da se prati neka tipična strategija
- Obično se dolazi do neke hibridne strategije, koja predstavlja kombinaciju različitih elemenata pomenutih strategija
- Upravo je mogućnost oblikovanja i primenjivanja hibridnih strategija jedna od najvažnijih osobina fazno-iterativnog razvoja koja propagira RUP:
 - Za složene probleme ili nepoznate domene problema, gde je neophodno više istraživanja, povećava se broj iteracija u Razradi
 - Za razvojno složene probleme, gde se složenost prevodi u dizajn koda, povećavaju se broj i veličina iteracija u Konstrukciji

- Ako je potrebno isporučiti proizvod inkrementalno, povećava se broj iteracija Prelaska

Filozofija RUP-a se baš i poklapa sa ovom strategijom tj. ideja RUP-a je prilagođavanje trenutnim potrebama tima.

205. Šta su discipline (RUP)? Navesti bar 6 disciplina.

Postoje skupovi poslova koji su međusobno povezani u nekoliko disciplina. Discipline:

- Poslovno modeliranje
- Zahtevi
- Analiza i projektovanje
- Konfigurisanje i upravljanje izmenama
- Okruženje
- Implementacija
- Upravljanje projektom
- Testiranje

206. Šta je disciplina „poslovno modeliranje“ (RUP)? Čime se bavi? Svrha? Koje poslove obuhvata (bar 5)?

Disciplina „poslovno modeliranje“ je posvećena:

- Razumevanju postojećeg stanja
- Sagledavanju mogućih unapređenja
- Stvaranju vizije budućeg sistema

Svrha:

- Razumevanja postojećeg stanja i problema u poslovnog organizacija
- Prepoznavanje potencijalnih promena u organizaciji
- Procenjivanje uticaja promena u organizaciji
- Staranje da svi ulagači na isti način razumeju organizaciju
- Izvođenje zahteva za softverski sistem
- Razumevanje kako bi se budući sistem uklopio u organizaciju

Poslovi:

- Procenjivanje ciljnog sistema
- Postavljanje i prilagođavanje ciljeva
- Prepoznavanje poslovnih ciljeva i ključnih indikatora performansi
- Analiza poslovne arhitekture
- Pravljenje rečnika uobičajenih poslovnih termina
- Uređivanje poslovnih pravila
- Pronalaženje poslovnih aktera i slučajeva upotrebe
- Analiza funkcionalnih celina
- Analiza poslovnih procesa

- Definisanje konteksta poslovnog sistema
- Prepoznavanje poslovnih procesa
- Doterivanje definicija poslovnih procesa
- Definisanje ostvarivanja poslovnih procesa
- ...

207. Šta je disciplina „zahtevi“ (RUP)? Koja je svrha ove discipline? Koje poslove obuhvata (bar 7)?

Disciplina “zahtevi” je posvećena:

- prikupljanju zahteva ulagača i
- njihovom transformisanju u skup rezultata rada zahteva, koji određuju:
 - okvire sistema koji se gradi
 - i detaljne zahteve u odnosu na ono što sistem mora da radi

Svrha:

- Obezbeđivanje i održavanje dogovora sa ulagačima u vezi toga šta bi sistem trebalo da radi
- Pružanje boljeg razumevanja zahteva razvijaoima
- Definisanje granica sistema
- Pružanje osnove za planiranje tehničkog sadržaja iteracija
- Pružanje osnove za procenjivanje vremena i troškova razvoja sistema
- Definisanje korisničkog interfejsa sistema, sa akcentom na potrebe i ciljeve korisnika

Poslovi:

- Prikupljanje zajedničkog rečnika
- Pronalaženja aktera i slučajeva upotrebe
- Razvijanje vizije sistema
- Prikupljanje zahteva ulagača
- Razvijanje pratećih specifikacija
- Određivanje prioriteta slučajeva upotrebe
- Detaljno dokumentovanje slučajeva upotrebe
- Detaljno dokumentovanje softverskih zahteva
- Struktuiranje modela slučajeva upotrebe
- Revidiranje zahteva

208. Šta je disciplina „analiza i projektovanje“ (RUP)? Čime se bavi? Svrha? Koje poslove obuhvata (bar 5)?

Disciplina “analiza i projektovanje” je posvećena:

- transformisanju rezultata rada u disciplini “Zahtevi” u rezultate rada koji određuju projekat softvera koji će se razvijati

Svrha:

- Transformisanje zahteva u projekat budućeg sistema
- Postepeno razvijanje robusne arhitekture sistema
- Prilagođavanje projekta implementacionom okruženju i postizanju punih performansi

Poslovi:

- Arhitekturna analiza
- Izvođenje "dokaza" validnosti arhitekture
- Procenjivanje ispravnosti arhitekture
- Analiza slučajeva upotrebe
- Identifikovanje projektnih mehanizama
- Identifikovanje elemenata projekta
- Ugrađivanje postojećih elemenata projekta
- Strukturiranje implementacionog modela
- Opisivanje arhitekture procesa izvršnog sistema
- Opisivanje distribucije
- Revidiranje arhitekture
- Projektovanje korisničkog interfejsa
- Pravljenje prototipa korisničkog interfejsa
- Revidiranje projekta
- Projektovanja slučajeva upotrebe
- Projektovanje podsistema
- Projektovanje klasa
- Projektovanje baze podataka

209. Šta je disciplina „konfigurisanje i upravljanje izmenama“ (RUP)? Čime se bavi? Svrha? Koje poslove obuhvata (bar 5)?

Disciplina "konfigurisanje i upravljanje izmenama" je posvećena:

- kontrolisanju i sinhronizovanju rezultata rada nastalih tokom iteracija
- komponovanju pojedinih rezultata rada u softverski sistem

Svrha:

- Podrška razvojnog metodu
- Održavanje integriteta proizvoda
- Obezbeđivanje kompletnosti i korektnosti konfigurisanog proizvoda
- Pružanje stabilnog okruženja za razvijanje proizvoda
- Ograničenje izmena rezultata rada u skladu sa politikom projekta
- Omogućava praćenje razlova, vremena i aktera izmena na rezultatima rada

Poslovi:

- Podnošenje zahteva izmene
- Revidiranje zahteva za izmene
- Potvrđivanje duplikata ili odbijenih zahteva za izmene

- Raspoređivanje i dodeljivanje poslova
- Planiranje konfigurisanja projekta i kontrola izmena
- Podešavanje okruženja za upravljanje konfigurisanjem sistema
- Menjanje i postavljanje konfiguracije
- Upravljanje ključnim tačkama projekta i izdavanja
- Praćenje i izveštavanje o stanju konfiguracije

210. Šta je disciplina „okruženje“ (RUP)? Čime se bavi? Svrha? Koje poslove obuhvata (bar 5)?

Disciplina „okruženje“ je posvećena:

- Organizaciji elemenata koji čine okruženje za razvoj softvera
- Tehničkoj i komunikacionoj podršci razvojnom timu
- Uključuje procese i alate

Svrha:

- Obezbeđivanje da organizacija za razvoj softvera (timovi ili pojedinci) imaju odgovarajuće okruženje za rad, koje podržava njihov rad na razvoju softvera

Poslovi:

- Povezivanje razvojnog procesa sa projektom
- Pokretanje razvojnog procesa
- Podrška razvoju

211. Šta je disciplina „implementacija“ (RUP)? Čime se bavi? Svrha? Koje poslove obuhvata (bar 5)?

Disciplina „implementacija“ je posvećena:

- razvoju, organizaciji, testiranju i integrisanju komponenti na osnovu projektne dokumentacije

Svrha:

- Određivanje organizacije koda kroz implementaciju podsistema organizovanih po slojevima
- Implementiranje elemenata projekta u vidu elemenata implementacije (izvorni kod, izvršni programi i drugo)
- Testiranje razvijenih komponenti kao jedinica
- Integrisanje pojedinačnih rezultata (pojedinaca ili timova) u izvršnom sistem

Poslovi:

- Strukturiranje implementacionog modela
- Planiranje integracije podsistema
- Implementacija elemenata projekta

- Analiza ponašanja pri izvršavanju
- Implementacija razvojnih testova
- Izvođenje razvojnih testova
- Revidiranje koda
- Integrisanje podsistema
- Integrisanje sistema

212. Šta je disciplina „upravljanje projektom“ (RUP)? Čime se bavi? Svrha? Koje poslove obuhvata (bar 5)?

Disciplina „upravljanje projektom“ je posvećena:

- svim različitim aktivnostima koje se tiču upravljanja projektom

Svrha:

- Obezbeđivanje okruženja za upravljanje projektima razvoja softvera
- Pružanje praktičnih vodilja za planiranje, rad sa kadrovima, vođenje i praćenje projekata
- Pružanje okvira za upravljanje rizicima

Poslovi:

- Prepoznavanje i procenjivanje rizika
- Razvoj poslovnog slučaja
- Započinjanje projekta
- Revizija pri odobrenju projekta
- Definisanje organizacije i osoblja
- Planiranje faza i iteracija
- Revizija plana projekta
- Razvoj plana iteracije
- Revizija plana iteracije
- Raspoređivanje i dodeljivanje poslova
- Izveštavanje o stanju
- Organizacija revizije
- Vođenje revizije
- Obezbeđivanje osoblja
- Započinjanje iteracije
- Prepoznavanje i procenjivanje rizika
- Završna faza
- Zatvaranje projekta

213. Šta je disciplina „testiranje“ (RUP)? Čime se bavi? Svrha? Koje poslove obuhvata (bar 5)?

Disciplina „testiranje“ je posvećena:

- procenjivanju i proveravanju kvaliteta proizvoda

Svrha:

- Traženje i dokumentovanje propusta u kvalitetu softvera
- Određivanje preporuka na osnovu prepoznatog kvaliteta softvera
- Proveravanje i dokazivanje pretpostavki koje potiču iz projekta i specifikacije zahteva
- Proveravanje da li softverski proizvod radi kako je projektovano
- Proveravanje da su zahtevi implementirani na odgovarajući način

Poslovi:

- Identifikovanje ideje testiranja
- Određivanje pristupa testiranju
- Implementiranje skupova testova
- Izvođenje skupova testova
- Određivanje rezultata testiranja
- Procenjivanje i argumentovanje kvaliteta
- Struktuiranje implementacija testova

214. Navesti skupove uloga subjekata u okviru metodologije RUP. Objasniti po jednom rečenicom.

Skupovi uloga:

- Opšte uloge
 - ulagač, koordinator pregledanja, pregledač, tehnički pregledač, ...
- Rukovodioci
 - projekta, kontrole izmena, testiranja, konfigurisanja, administrator sistema, isporučivanja
- Analitičari
 - ulagač, određivač zahteva, analitičar sistema, analitičar poslovnih procesa, poslovni arhitekta, poslovni projektant
- Razvijaoci
 - softversko arhitekta, projektant, projektant baze podataka, projektant korisničkog interfejsa, implementator, integrator, sigurnosti arhitekta
- Testeri
 - upravljač testove, analitičar testova, projektant testova, tester
- Primena i podrška
 - inženjer procesa, administrator sistema, razvijalac obuke, grafički urednik, autor tehničke dokumentacije, specijalisti za alate

215. Navesti i objasniti analitičarske uloge subjekata u metodologiji RUP.

Pogledati pitanje 214.

216. Navesti i objasniti razvijačke uloge subjekata u metodologiji RUP.

Pogledati pitanje 214.

217. Šta je rezultat rada (RUP)?

Rezultat rada je sve ono što nastaje kao rezultat jednog ili više poslova tokom rada na analizi ili razvoju sistema. Klasifikuju se prema disciplini ili prema vrsti. Primeri: Projektovanje baze podataka (projektovanje), implementacija baze podataka (implementacija), interfejs komponente (projektovanje), implementirane komponente (implementacija).

218. Navesti bar 7 vrsta rezultata rada (RUP).

Vrste rezultata rada:

- procene
- koncepti
- infrastruktura
- modeli
- elementi modela
- planovi
- procesi
- podaci projekta
- rešenja
- specifikacije

219. Navesti i objasniti vrste modela u metodologiji RUP?

- Model slučajeva upotrebe
- Analitički model
- Model isporučivanja
- Projektni model
- Model podataka
- Implementacioni model
- Karta navigacije (kroz kor. int.)
- Modeli poslovanja
 - analitički
 - model isporučivanja
 - projektni
 - slučajeva upotrebe
- Model usluga
- Model analize opterećenja

220. Navesti vrste planova u metodologiji RUP?

- Plan razvoja softvera
- Plan iteracije
- Plan integracije
- Plan upravljanja zahtevima
- Plan obezbeđivanja kvaliteta
- Plan ustanovljavanja problema
- Plan upravljanja konfiguracijom
- Plan upravljanja rizicima
- Plan merenja
- Plan testiranja
- Plan isporučivanja
- Plan usvajanja proizvoda

221. Šta je bezbednost informacija?

Bezbednost informacija je stepen u kome je onemogućeno ugrožavanje informacija.

Bezbednost informacije je zaštita informacija i IS od neautorizovanog pristupa, upotrebe objavljivanja, menjanja, uništavanja, kao i od sprečavanja raspoloživosti.

Pojmovi:

- Opasnost
 - svaki događaj koji potencijalno ugrožava bezbednost
- Slabost
 - svaki element sistema, ili njegova karakteristika, koji izlaže sistem okolnostima
- Rizik
 - verovatnoća da se dogodi nešto loše (da se opasnost ostvari)
- Uticaj
 - predmet, obim i/ili dubina mogućih posledica neke opasnosti

222. Navesti i ukratko objasniti najvažnije predmete ugrožavanja bezbednosti informacija.

Predmetom ugrožavanja bezbednosti nazivamo elemente informacionih sistema koji mogu da budu dovedeni u pitanje usled ugrožavanja bezbednosti.

Osnovni predmeti ugrožavanja:

- Poverljivost informacija (confidentiality)
- Integritet informacija (integrity)
- Raspoloživost informacija (Availability)

Nazivaju se i bezbednosna trojka CIA ili CAI

Odgovarajući postupci ugrožavanja predmeta (DAD):

- odavanje (disclosure)
- menjanje (alteration)
- ometanje (denial)

223. Šta je poverljivost informacija i kako se može ugroziti?

Poverljivost informacija je sposobnost da se podaci zaštite od neovlašćenog pristupa. Koncept je sličan privatnosti (ali ne isti) i predstavlja neophodan sastavni deo privatnosti. Poverljivost informacija se ugrožava odavanjem informacija.

224. Šta je integritet informacija i kako se može ugroziti?

Integritet informacija je sposobnost da se spreči neovlašćeno ili nepoželjno menjanje podataka. Ugrožava se menjanjem informacija.

Što sistem ima više pravila integriteta, to je nekontrolisano menjanje teže.

225. Šta je raspoloživost informacija i kako se može ugroziti?

Raspoloživost informacije je sposobnost da se pristupi podacima kada su potrebni. Ugrožava se ometanjem informacija.

226. Navesti i ukratko objasniti najvažnije vrste napada na informacione sisteme.

Napadi su namerni izazvane opasnosti.

Osnovne vrste napada:

- Upad (interception)
- Onesposobljavanje (interruption)
- Menjanje informacija (modification)
- Pravljenje informacija (fabrication)

227. Opisati ukratko vrstu napada “upad”? Šta je predmet ugrožavanja?

Upad je svaki pokušaj neovlašćenog pristupanja nekog elementa informacionog sistem. Napad ugrožava poverljivost.

Osnovne mete su:

- korisnički nalozi i lozinke
- komunikacija

Neke od tehnika:

- pecanje
- socijalni inženjering
- provocirane korisničke greške
- presretanje komunikacije
- umetanje upita

228. Opisati ukratko vrstu napada “onesposobljavanje”? Šta je predmet ugrožavanja?

Onesposobljavanje je napad koji ima za posledicu neki vid pune ili delimične neraspoloživosti sistema ili njegovog dela. Onesposobljavanje ugrožava raspoloživost.

Osnovne mete su:

- ključni uređaji računarskog sistema
- ključne komunikacije računarskog sistema

Neke od tehnika:

- onesposobljavanje upotrebe sistema (DOS)
- fizičko onesposobljavanje
- podmetanje zlonamernog koda

229. Opisati ukratko vrstu napada “menjanje informacija”? Šta je predmet ugrožavanja?

Napad kojim se menjaju postojeći podaci (zahteva prethodno upad ili slično...). Ugrožava integritet podataka, ali posredno može da ugrozi i raspoloživost.

Osnovni ciljevi su:

- menjanje radi ostvarivanja drugog cilja
- menjanje radi prikrivanja
- menjanje radi uništavanja

Neke od tehnika:

- prethodni upad
- umetanje upita

230. Opisati ukratko vrstu napada “pravljenje informacija”? Šta je predmet ugrožavanja?

Napad kojim se proizvode novi napadi u sistemu. Ova vrsta napada ugrožava integritet (i raspoloživost ako je u pitanju velika količina novih podataka). Može da se posmatra kao podvrsta menjanja. Često je mnogo lakše napraviti nove podatke nego menjati postojeće.

Osnovna meta su:

- važne baze podataka
- važne komunikacije

Neke od tehnika:

- upad u sistem
- presretanje komunikacije

Pravljenje informacija se često koristi za sakrivanje nekih izmena. Primer: Dodaje se ogroman broj novih studenata u nadi da neće da se primeti promenjena ocena nekog studenta.

231. Koji slojevi sistema se prepoznaju pri planiranju zaštite?

Zaštita se izvodi planski:

- Sagledavaju se moguće opasosti i rizici
- Implementiraju se odgovarajući zaštitni mehanizmi
- Definišu se protokoli reagovanja na opasnosti

Zaštita sistema se organizuje po slojevima, navodeni su od najužeg do najšireg:

- Podaci
- Aplikacija
- Server
- Interna mreža
- Granice mreže
- Spoljašnja mreža

232. Na koji način se štite podaci IS?

Načini:

- kriptovanje
- kontrola pristupa
- rezervne kopije
- proveravanje upada
- analiza slabosti

233. Na koji način se štite aplikacije IS?

Načini:

- sistemi za objedinjenu proveru autentičnosti
- filtriranje sadržaja
- provera ispravnosti podataka
- praćenje i pravljenje dnevnika aktivnosti
- provera upada
- analiza slabosti

234. Na koji način se štite serveri (čvorovi) IS?

Načini:

- provera autentičnosti
- zaštita od virusa
- zaštitni zid
- sistem za prepoznavanje upada
- sistem za prevenciju upada
- kriptovanje lozinki
- praćenje i pravljenje dnevnika aktivnosti
- provera upada
- analiza slabosti

235. Na koji način se štiti interna mreža IS?

Načini:

- sistem za prepoznavanje upada
- sistem za prevenciju upada
- praćenje i pravljenje dnevnika aktivnosti
- provera upada
- analiza slabosti

236. Na koji način se štiti granica mreže IS?

Načini:

- zaštitni zid
- proksi
- nadzor paketa
- praćenje i pravljenje dnevnika aktivnosti
- provera upada
- analiza slabosti

237. Na koji način se štiti spoljašnja mreža IS?

Načini:

- izolovanje mreže
- virtualno privatno umrežavanje (VPN)
- praćenje i pravljenje dnevnika aktivnosti
- provera upada
- analiza slabosti

238. Kako se bezbednosne opasnosti / problemi dele prema uzroku? Objasniti.

Uzroci opasnosti:

- Spoljašnje opasnosti: opasnosti čiji su nosioci subjekti van organizacije
- Unutrašnje opasnosti: opasnosti čiji su nosioci subjekti zaposleni u organizaciji
- Nezgode: opasnosti koje nastanu bez eksplicitne ljudske aktivnosti

Najčešće i najviše pažnje se pridaje spoljašnjim opasnostima, ali unutrašnje opasnosti i nezgode prave više štete, jer do njih dolazi češće i IS su za njih slabije pripremljeni.

239. Kako se bezbednosne opasnosti / problemi dele prema načinu ispoljavanja? Objasniti.

Načini ispoljavanja:

- Otvorene probleme
 - To su problemi koji su relativno lako uočljivi
 - Obično se radi o ugrožavanju integriteta ili raspoloživosti
 - Često je baš i cilj da se primeti da postoji problem
 - Relativno retko nastaju kao posledica unutrašnjih napada
 - Često služe kao maska za prikrivene probleme
- Prikrivene probleme
 - To su problemi koji se relativno teško uočavaju
 - Obično se radi o ugrožavanju poverljivosti
 - Često među najvažnije ciljeve preduzetih napada spada i što bolje prikrivanje problema
- Nenamerni problemi:
 - Imaju veoma različita ispoljavanja
 - Nastaju kao posledica nezgoda ili usled nepažnje unutrašnjeg korisnika

240. Navesti uobičajene opasnosti po bezbednost IS.

Vrste opasnosti:

- Greške i propusti
- Prevare i krađe

- Unutrašnje sabotaza
- Infrastruktura i fizički problemi
- Zlonamerni hakeri
- Industrijska špijunaža
- Zlonamerni kod
- Špijunaža stranih država
- Pretnje ličnoj privatnosti

241. Objasniti opasnosti “greške i propusti”, “prevare i krađe” i “infrastrukturni i fizički problemi”.

“Greške i propusti”

- skoro uvek su rezultat nepažnje unutrašnjih korisnika
- nastaju:
 - pri unošenju podataka
 - pri obradi podataka
 - pri instalaciji i konfigurisanju softvera
 - pri procedurama održavanja sistema

“prevare i krađe”

- obično prikrivene, podjednako unutrašnje i spoljašnje
- podvrste:
 - unutrašnje prevare
 - prevare u javnom prostoru
 - krađe podataka
 - krađe hardvera

“infrastrukturni i fizički problemi”

- obično su u pitanju nezgode, ali mogu da budu i posledice sabotaze
- podvrste problema:
 - problem sa napajanjem
 - prekid komunikacije
 - curenje vode i nestanak vode
 - problemi sa kanalizacijom
 - prekidi u transportnim uslugama
 - požari, poplave
 - građanski nemir, štrajkovi

242. Objasniti opasnosti “unutrašnja sabotaza”, “industrijska špijunaža” i “međudržavna špijunaža”.

“unutrašnja sabotaza”

- obično otvorene, podjednako unutrašnje i spoljašnje (uz unutrašnju)

- vidovi:
 - uništavanje opreme
 - postavljanje logičkih bombi
 - neispravan unos podataka
 - brisanje podataka
 - menjanje podataka
 - obaranje sistema
 - uzimanje podataka za taoce
- različiti uzroci:
 - osveta
 - “viši” ciljevi
 - dosada i zabava
- Procenjuje se:
 - manje slučajeva sabotaze nego krađe
 - ali sabotaze su obično skuplje

“industrijska spijunaža”

- najčešće spoljašnja i prikrivena, ali može da bude i unutrašnja
- cilj prikupljanje tuđih podataka
 - radi ostvarivanja sopstvene vrednosti
 - radi nanošenja štete drugome
- u stalnom porastu

“međudržavna spijunaža”

- po svemu slična industrijskoj spijunaži, ali sa drugačijim ciljevima
- postoji oduvek, samo menja oblike
- najčešće prikrivena i spoljašnja, ali može da bude i unutrašnja
- čin prikupljanja tuđih podataka
- u stalnom porastu

243. Objasniti opasnosti “zlonamerni hakeri”, “zlonamerni kod” i “pretnje ličnoj privatnosti”.

“zlonamerni hakeri”

- po pravilu spoljašnje opasnosti, najčešće otvorenog tipa
- ovde obično ubrajamo sve zlonamerne neovlašćene upada u računarski sistem
 - mogu da budu spoljašnji i unutrašnji
 - uglavnom se ciljaju značajni društveni ili ekonomski ciljevi
 - mnogi sistemi se napadaju svakodnevno
- veoma nepredvidivi
 - i ciljevi i namere često nisu unapred predvidivi
 - napadi hakera čine da se ljudi osećaju nesigurno, što je često glavni cilj napada
- Često im se pridaje veći značaj nego što realno zaslužuju
 - fokusiranje na ovu vrstu vodi zanemarivanju drugih vrsta opasnosti

“zlonamerni kod”

- obično spoljašnje opasnosti
- u fazi širenja prikrivene, a u fazi dejstva otvorene
- frontalne, ali mogu da budu i usmerene na konkretne ciljeve
- virusi, crvi, trojanski konj, logičke bombe i drugo
- broj virusa raste eksponencijalno, a broj incidenata sporije

“pretnje ličnoj privatnosti”

- Uglavnom spoljašnja, najčešće prikrivena
- Ima sličnosti sa spijunažom
- čin prikupljanja tuđih podataka
 - radi ostvarivanja sopstvenih ciljeva
 - radi nanošenja štete
- postaje sve veći problem
 - porast upotrebe računarstva u oblaku
 - ogromne količine ličnih podataka u različitim računarskim sistemima
 - zaposleni u tim sistemima često imaju neograničen pristup ličnim podacima

244. U čemu se kriju uzroci postojanja problema bezbednosti?

Uzroci:

- greške i propusti
- nepošteni zaposleni
- razočarani zaposleni
- infrastrukturni problemi
- voda
- spoljašnji uticaji (virusi, spijunaža, ...)

245. Objasniti pojmove identifikacija, autentikacija i autorizacija.

Identifikacija je provera indentiteta.

- Identifikacija je pretpostavljanje identiteta subjekta
- Uobičajeno se odvija na osnovu imena:
 - imena osobe
 - korisničkog imena
 - ikonice korisnika
 - broja računa
 - ID kartice
 - otiska prsta
 - DNK
- Sama od sebe obično predstavlja slab koncept

Autentikacija je provera autentičnosti identiteta (obično ide u paru sa identifikacijom).

- Autentikacija je skup metoda za ustanovljavanje istinitosti saopštenog identiteta

- Faktori autentikacije su elementi koji učestvuju u proveru (što je više faktora, to je provera pouzdanija)
- Uobičajeni faktori su ono što osoba može da zapamti ili nosi sa sobom

Autorizacija je provera prava specifičnost subjekta (naziva se i kontrola pristupa)

- Kontrola pristupa je staranje od tome:
 - Ko sme da koristi koje podatke i na koji način?
 - Ko sme da radi koje poslove?

246. Šta je i kako se vrši provera identiteta? Šta je falsifikovanje identiteta?

Provera identiteta (verifikacija identiteta) je korak nakon identifikacije, ali pre autentikacije. Od osobe se traži da potvrdi svoj identitet.

Svaki metod identifikacije može da bude predmet falsifikovanja. Krađa identiteta počiva na uspešnom falsifikovanom identitetu. Uglavnom odstupa zbog odstustva autentikacije.

247. Objasniti najvažnije faktore autentikacije.

Faktori:

- Memorijski faktori
 - lozinka
 - PIN
 - fraze za propuštanje
 - neka sekvenca postupaka
 - bilo šta što može da se zapamti
- Fizički faktori:
 - ID kartica
 - token
- biometrijski faktori
 - visina, težina, boja kože ili očiju
 - otisak prsta
 - uzorak irisa ili retine
 - karakteristične mere lica
 - rukopis

Mouće da se koristi i više faktora (kombinacija)

248. Objasniti praćenje aktivnosti kao deo sistema bezbednosti IS.

Obračunavanje: Svaka obavljena operacija ostavlja trag koji omogućava da se naknadno proveru koji subjekat ju je izvršio

- radi naplate ili

- radi provere zloupotrebe

Proveravanje: Proveravanje značajnih informacija sistema

- radi provere ugrožavanja bezbednosti
- i drugih razloga

Pravljenje dnevnika: Fizičko zapisivanje informacija o operacijama u sistemu.

- tehnička pretpostavka za ostvarivanje nadzora

249. Navesti uobičajene slabosti IS u pogledu bezbednosti.

Uobičajene slabosti IS:

- Apatija korisnika
 - iako su svesni mogućnosti napada, korisnici uobičajeno imaju stav "to se neće dogoditi meni"
 - ležerno se deli softver
- Nedovoljna bezbednosna kontrola
 - mnogi računari, posebno lični, nemaju zaštitu protiv virusa i drugih bezbednosnih opasnosti
- Loša upotreba postojećih bezbednostnih alata
 - antivirusni programi moraju biti ažurni
 - ležerno dodeljivanje privilegija nego što je neophodno
 - omogućavanje slabih lozinki
- Slabosti operativnih sistema:
 - širina i veličina operativnih sistema i drugog sistemskog softvera zahteva veliki broj programera
 - dovoljno je da tim nije dorastao poslu i ostaće neki sigurnosti propust
- Neovlašćena upotreba
 - postoje ljudi kojima je izazov upadanje u tuđe sisteme
 - kada jednom upadne u sistem, izazov je da načini štetu
- Mrežna anonimnost
 - zlonamerna osoba na mreži je anonimna, što otežava otkrivanje i olakšava zlonamerno pristupanje

250. Navesti osnovna sredstva za prevazilaženje bezbednosnih slabosti IS.

Sredstva:

- Treninzi
 - bezbednosti službenici moraju da budu u toku sa aktuelnim problemima i rešenjima
- Bezbednosna analiza pri nabavci softvera
 - pre stavljanja u rad novog softvera potrebna je odgovarajuća bezbednosna analiza pri sagledavanju i prevazilaženju eventualnih bezbednostih rizika

- Nadgledavanje
 - nadgledavanje korisnika, sistema, mreže
- Plan za slučaj opasnosti
 - pri prepoznavanju vanrednog slučaja se mora reagovati brzo
 - svaki sekund utrošen na razmišljanje potencijalno uvećava štetu
 - moraju postojati definisane precizne procedure postupanja u vanrednim slučajevima
- Ograničavanje deljenja
 - svi deljenji resursi moraju biti strogo upravljani
 - zabrana spoljašnjih uređaja za prenos podataka
 - periodično čišćenje i sklanjanje podataka koji su bili deljeni
- Samoizolacija u slučaju napada
 - u slučaju prepoznavanja da je napad u toku, obično je najočiglednija mera potpuni prekid komunikacije interne mreže sa spoljašnjom mrežom
 - to je izvesno temeljno suprostavljanje napadu
 - međutim, ako je poslovanje tesno zavisno sa spoljašnjom mrežom, to nije dobro rešenje
- Proveravanje (audit)
 - proveravanje omogućava uočavanje neobičajenih aktivnosti i podizanja nivoa pažnje
- Rezervne kopije
 - neophodno je imati definisanu preciznu proceduru pravljenja rezervnih kopija i izvođenja postupka oporavka podataka
- Temeljna zaštita od zlonamernog softvera
 - antivirus programi
 - antišpijunski program
 - zaštitni zid

251. Šta je fleksibilnost arhitekture softvera?

Težimo da pravimo softver tako da je ponovno upotrebljiv. Zbog toga je neophodno da softver bude proširiv i fleksibilan.

Fleksibilnost arhitekture predstavlja sposobnost

- da se cela arhitektura
 - prilagođava promenama u okruženju
 - primenjuje u novim okolnostima
- da elementi arhitekture
 - prilagođavaju promenama u okruženju
 - primenjuju u novim okolnostima
 - primenjuju u drugim novim projektima

Fleksibilnost je bliska ponovnoj upotrebljivosti.

252. Šta je proširivost arhitekture softvera?

Proširivost arhitekture je svojstvo arhitekture

- da može da se proširuje
- da se proširivanje ostvaruje
 - primarno dodavanjem novih elemenata
 - bez značajnog menjanja postojećih elemenata

253. U čemu je značaj fleksibilnosti i proširivosti softvera?

Fleksibilnost i proširivost omogućavaju pojednostavljivanje životnog ciklusa sistema za sve učesnike. Kao posledica fleksibilnosti i proširivosti dobijamo sledeće karakteristike:

- modularnost
- konfigurabilnost
- robusnost
- mogućnost brze reakcije na promene u poslovnim zahtevima
- ...

254. Šta je SOA?

SOA (Service-Oriented Architecture) je rezultat evolucije softverske industrije sa ciljem postizanja maksimalne fleksibilnosti i proširivosti.

Modularizacijom se celine sistema oblikuju u komponente koje mogu da “samostalno” pružaju određene “celovite” usluge. Takve komponente se nazivaju servisi.

Komponente se dobijaju funkcionalnom dekompozicijom tj. prepoznavamo celine koje obavljaju neke zaokružene funkcije.

Definicija: SOA (servisno orijentisana arhitekture) je arhitekturni stil koji promoviše korišćenje labavo povezanih servisa radi obezbeđivanja visoke fleksibilnosti na interoperabilan i tehnološki nezavisan način.

Tehnologija implementacije jednog servisa ne treba da utiče na tehnologiju implementacije nekog drugog servisa, ali moraju da postoje protokoli komunikacije koji koriste istu tehnologiju.

255. Šta predstavlja aplikacija u okviru arhitekture orijentisane prema servisima?

Aplikacija predstavlja kompoziciju i grupisanje labavo povezanih, usko definisanih i na standardima utemeljenih servisa u veće složenije celine.

256. Koje su osnovne karakteristike SOA?

Karakteristike:

- Softverski elementi se dele kao “samostalni” servisi. Za tim je proizvod servis, a ne aplikacija. Tim ne vidi aplikaciju, već vidi servis koji treba da implementira sa svim specijalnim slučajevima.
- Postoje ograničenja što se tiče protokola, ali razvoj pojedinih servisa je fleksibilan
- Spregnutost među servisima je vrlo niska
- Cena komunikacije je zbog strogih protokola relativno visoka
- Uklapa se relativno sa agilnom metodologijom
- Savremeni principi razvoja:
 - modularnost
 - enkapsulacija
 - niska spregnutost
 - ponovna upotrebljivost
 - implementacija kao sastavljenje
 - ...

257. Objasniti pojam servisa u kontekstu SOA.

Servis je izdvojena (diskretna) jedinica poslovne (tehničke) funkcije koja je ponovno upotrebljiva.

Servisi su:

- povezani
- slabo spregnuti
 - definisani su interfejsima nezavisnim od implementacije
 - korisnik servisa zavisi samo od interfejsa
 - promoviše se fleksibilnost pri izmenama implementacije
- transparentni u odnosu na lokaciju
 - ne koriste druge servise na osnovu lokacije
- međusobno sarađuju (interoperabilni)
- raspoloživi nezavisno od implementacije ili protokola prenosa
 - korisnici i pružaoci servisa mogu da biti implementirani na različitim platformama
 - komunikacioni protokoli servisa treba da budu kompatibilni sa različitim platformama
 - otvoreni standardi

Uloge:

- Servise nudi pružaoc servisa (Service Provider) koji objavljuje interfejs i politike.
- Servise koriste korisnici servisa (Service Consumer) koji koriste interfejs primenom pouzdanih i bezbednih protokola.
- Servisima posreduju medijatori servisa (Service Mediator) koji obezbeđuju otkrivanje, izbor i praćenje rada.

258. Kako se opisuju servisi (SOA)?

Servisi se opisuju primenom standardizovanih postupaka za opisivanje interfejsa:

- metapodaci servisa
 - određuje se servis, njegove operacije, ulazne i izlazne parametre
 - određuje se kako se servisu pristupa, kao i koja je njegova lokacija (service endpoint)

259. Šta je registar servisa (SOA)?

Registar servisa predstavlja vid kataloga putem koga mogu da se objavljuju (publish) i pronalaze (discover) servisi. Servise objavljuju organizacije koje pružaju servise:

- radne jedinice u organizaciji
- organizacije

Registar servisa je sredstvo za dodatno spuštanje nivoa spregnutosti. Korisnik servisa ne mora da zna ko pruža servis, gde se fizički nalazi implementacija servisa i u nekim slučajevima čak ni sve detalje interfejsa (operacija).

260. Koje su osnovne vrste servisa u SOA?

Osnovne vrste servisa:

- servisi poslovnih procesa
 - servis koji završava neku poslovnu (celovitu) funkciju
- servisi poslovnih transakcija
 - servis za transakciju koji menja sliku stanja domena
 - servis poslovnog procesa može da koristi nekoliko servisa poslovnih transakcija
- servisi poslovnih funkcija
 - servis koji vraća neki rezultat i ne menja stanje domena
- servisi tehničkih funkcija
 - nemaju pristup poslovnoj bazi podataka i nezavisne su od stanja domena
 - obrađuju neke tehničke funkcije (npr. interfejs ka mejl serveru)

Za postojeću aplikaciju mogu da se naprave i njoj dodele servisi fasade

- izlažu kroz interfejs servisa funkcija aplikacije
- sakrivaju implementaciju

SOA u osnovi nema slojeve, ali zbog prirode vrsta servisa se često u implementaciji IS-a prave neki slojevi:

- fasadni servisi čine najviši sloj
- servisi poslovnih procesa koriste servise poslovnih transakcija i funkcija
- servisi poslovnih transakcija mogu da koriste poslovne funkcije
- servisi poslovnih funkcija mogu da koriste neke tehničke funkcije

261. Šta su servisi poslovnih procesa (SOA)?

Pogledati pitanje 260.

262. Šta su servisi poslovnih transakcija (SOA)?

Pogledati pitanje 260.

263. Šta su servisi poslovnih funkcija (SOA)?

Pogledati pitanje 260.

264. Šta su servisi tehničkih funkcija (SOA)?

Pogledati pitanje 260.

265. Šta su Veb servisi? Objasniti osnove Veb servisa.

Veb servise je funkcija kojoj se pristupa preko veba (npr. preko http protokola). Primer: Google Maps nudi veb servis gde mi pošaljemo naše trenutne geografske koordinate kao ulaz, a kao rezultat dobijamo grad u kojem se nalazimo.

Postoji jezik WSDL (Web Service Description Language) za opisivanje veb servisa koji počiva na XML-u.

Mana veb servisa je što je neefikasan način komunikacije (skupo je pozivanje).

266. Objasniti pojmove virtualno i virtualizacija.

Virtualno:

- izvorno:
 - skoro isto kao nešto, ali ne u potpunosti ili ne u skladu sa strogom definicijom
 - “praktično isto”
- uobičajeno u računarstvu:
 - nešto što ne postoji fizički, ali se softverski stvara utisak postojanja

Virtualizacije je postupak stvaranja virtualne verzije nečega.

267. Šta je virtualna mašina? Navesti osnovne komponente koje učestvuju u virtualizaciji sistema.

Virtualna mašina je dodatni sloj apstrakcije između hardvera i operativnog sistema. Ideja je da neposredno rukovanje hardverom preuzima nadzornik virtualnih mašina (VMM - Virtual Machine Monitor).

NVM se oslanja na hardver i može istovremeno da opslužuje više operativnih sistema na istom računaru.

- Iz ugla svakog OS-a izgleda kao da je računar pod njegovom potpunom kontrolom.
- Iz ugla aplikativnog softvera nije vidljivo da li je OS neposredno nad hardverom ili se između nalazi NVM.

268. Objasniti motivaciju za virtualizaciju računara.

Razlikuju se motivi za virtualizaciju:

- radnih stanica
 - povećana produktivnost
 - neki programi ne postoje za sve vrste OS
 - različiti OS nisu jednako pogodni za sve vrste poslova
 - jedostavnije je imati više VM na jednom računaru nego održavati više fizičkih računara na jednom mestu
 - izolovanje osetljivih sistema
 - zaštita od virusa
 - obuka administratora, isprobavanje osetljivih postupaka
 - obučavanje u razvoju sistemskog softvera
 - razvojne radne jedinice
 - realno ponašanje mrežnog okruženja na jednoj radnoj jedinici
 - izgradnja na različitim razvojnim okruženjima
 - testiranje
 - virtualni uređaji
 - pojednostavljena distribucija pripremljenih složenih okruženja
- servera
 - konsolidacija servera
 - više manjih servera se virtualizuje na jednom računaru
 - razdvajanje odgovornosti servera, a time i olakšano održavanje
 - veća iskorišćenost hardvera
 - ušteda na hardveru
 - napredna serverska okruženja
 - olakšana replikacija
 - jednostavnija implementacija visoke raspoloživosti
 - prebacivanje aktivnih mašina sa jednog na drugi računar

269. Ukratko navesti istoriju virtualizacije računara.

1965. IBM Mainframe - razvijen koncept virtualizacije kao rešenje

1967. Dijkstra - predlaže različite nivoe apstrakcije VM

1990. Oak (Java) - primena aplikativne VM

1999. VMWare i druga PC rešenja

270. Koje su osnovne vrste virtualnih mašina? Objasniti specifičnosti i razlike.

Vrste virtualnih mašina:

- Sistemske virtualne mašine
 - pružaju apstrakciju čitavog računara
 - primeri: KVM, VMWare, VirtualBox, ...
- Procesne (aplikativne) virtualne mašine
 - pružaju delimičnu apstrakciju računara, dovoljnu za izvršavanje jednog procesa
 - Primeri: JVM, CLR, ...
- Virtualizacija na nivou OS-a
 - kombinacija - u okviru OS-a pružaju parcijalne instance istog OS-a
 - Primeri: FreeBSD Jail, LXC, OpenVZ

271. Šta su sistemske virtualne mašine? Na koje se sve načine implementiraju?

Sistemske virtualne mašine pružaju apstrakciju čitavog računara. Postoje različiti nivoi virtualizacije:

- puna virtualizacija
- hardverski podržava virtualizacija
- parcijalna virtualizacija
- paravirtualizacija

272. Šta je puna virtualizacija? Šta je hardverski podržana virtualizacija?

Puna virtualizacija simulira sve elemente hardvera potrebne za rad neizmenjenog operativnog sistema VM. Primeri: VMWare (Workstation, Server), Oracle VirtualBox, ...

Hardverski podržava virtualizacija pruža mehanizme za efikasan rad nadzornika, simuliranje hardvera i izolovanje BM. Primeri: VMWare (Workstation, Fusion), Oracle VirtualBox, ... Slično kao puna virtualizacija, ali se očekuje da hardver (procesori) imaju posebne instrukcije koje pomažu virtualizaciju nekih elemenata hardvera.

273. Šta je parcijalna virtualizacija? Šta je paravirtualizacija?

Parcijalna virtualizacija simulira većinu elemenata hardvera, ali ne sve. U okviru VM ne može da radi potpun OS (može jedna ili više aplikacija). Predstavlja nešto između sistemske i procesne virtualizacije.

Paravirtualizacija ne simulira hardver, već pruža alternativni API koji zamenjuje određene komponente gostujućeg OS-a (oslanjajući se na pravi OS). Zamenjuju se one komponente koje neposredno pristupaju hardveru, a osetljive instrukcije se kodiraju u skladu sa potrebama VM. Ova vrsta paralelizacije nudi veću efikasnost, ali manju fleksibilnost (sve VM moraju da izvršavaju isti OS).

274. Šta se primarno analizira pri razmatranju primene virtualnih mašina u nekom IS? Koji su mogući doprinosi virtualizacije servera u IS?

Osnovni kriterijumi su:

- spuštanje cene (hardvera, održavanja)
- podizanje fleksibilnosti
- skraćivanje perioda neaktivnosti sistema

Doprinosi:

- Konsolidacija
- Udvajanje virtualnih mašina
- Dinamička test okruženja
- Migracija virtualnih mašina
- Skaliranje

275. Objasniti konsolidaciju, kao doprinos virtualizacije? Objasniti.

Konsolidacija servera:

- više manjih servera se virtualizuje na jednom fizičkom računaru
- razdvajanje odgovornosti servera, a time i olakšano održavanje
- veća iskorišćenost hardvera
- ušteda na hardveru

Postavljanjem više malih VM na jedan fizički server

- štedi se prostor
- smanjuje se potrošnja struje
- smanjuje se učestalost hardverskih kvarova

276. Objasniti redundantnost, kao doprinos virtualizacije? Objasniti.

Redundantnost se odnosi na udvajanje virtualnih mašina.

Udvajanje VM je mnogo jednostavnije:

- umesto ponavljanja instalacije, VM se samo iskopira
- pravljenje kopija VM na različitim serverima je jednako sigurno kao držanje fizički različitih servera
- ako rezervna VM ne radi ništa na nekom serveru, znači da ima prostora za još drugih VM koje mogu nešto da rade

Udvajanje računara nad deljenim diskovima:

- često rešenje
- virtualni diskovi su na bezbednom (redundantim) uređajima
- više kopija VM koristi potpuno iste virtualne diskove
- jedna je aktivna, ako nešto otkáže odmah postoji spremna druga VM na drugom hardveru

277. Objasniti migraciju i skaliranje, kao doprinose virtualizacije? Objasniti.

Doprinosi migracija:

- Prebacivanje VM sa jednog na drugi server je jednostavno
 - Na novu mašinu se instalira samo NVM ili OS sa NVM
 - Tada mogu da se iskopiraju potrebne VM
 - savremena komercijalna rešenja omogućavaju ovu funkcionalnost bez prekida VM

Doprinosi skaliranja:

- Skaliranje sistema je lakše u virtualizovanom okruženju
 - kada se doda novi fizički server, na njega se prebace izabrane VM i time se rasterete zagušeni serveri
 - sve je to mnogo brže i jednostavnije uz skraćivanje perioda neaktivnosti sistema

278. Koja su najvažnija ograničenja virtualizacije računara? Kada virtualizacija nije dobro rešenje?

Virtualizacija servera nije efikasna ako je server podložan preterano visokom opterećenju nekog hardverskog resursa (ili više njih - procesor, memorija, diskovi, mreža, ...)

- deljenjem snage računara na više VM smanjuje se snaga svake od VM
- ako na serveru postoji samo jedna VM, čak i ona radi manje efikasno nego što bi radila da nije VM
- (čak i tada je lakše migrirati VM na novi računar nego u slučaju fizičkih mašina)

279. Objasniti pojam “računarstvo u oblaku”?

Naredni korak virtualizacije predstavlja “računarstvo u oblaku”

- čitava računarska mreža se apstrahuje jednim velikim virtualnim računarskim sistemom
- prividno mnogo programa radi na jednom velikom sistemu, a ustvari se izvršavaju pojedinačno, ili čak po delovima, na različitim računarima te mreže

Računarstvo u oblaku je skup hardverskih zasnovanih usluga, koje nude računarske, mrežne i skladištene resurse uz visoko apstrahovano upravljanje hardverom.

U osnovne koncepte spadaju:

- apstrakcija
 - upotreba apstrahovanih elemenata oblaka
 - sakrivanje stvarne implementacije
- virtualizacija
 - apstrahovani elementi se često ponašaju poput stvarnih
 - računarski sistemi, mrežni resursi, skladišta podataka, baze podataka, servisi, ...

280. Koji su osnovni doprinosi računarstva u oblaku?

Računarstvo u oblaku omogućava:

- usluge po zahtevu
 - bez složenih postupaka instalacije i održavanje
- intenzivan mrežni pristup
 - koriste se glavna čvorišta i mrežna komunikacija nije usko grlo
- automatsko zauzeće i oslobođenje resursa
 - resursi se alociraju (i plaćaju) prema potrebi
 - unapred se određuju samo granice
- naplatu prema utrošku
 - ne plaća se ono što se ne koristi (uglavnom)
- stručna sistemska podrška
 - uz odgovarajuću cenu...

281. Koje su osnovne vrste oblaka? Objasniti razlike.

Vrste oblaka prema uslugama:

- infrastruktura kao usluga
- platforma kao usluga
- softver kao usluga

282. Objasniti vrstu oblaka „Infrastruktura kao usluga“.

Poslužilac nudi sve elemente računarske infrastrukture u virtualizovanom obliku:

- virtualnu mašinu (procesore)
- skladišni prostor (diskove)
- memoriju
- mrežnu infrastrukturu
- ...

Ideja je da se nudi fizički računarski centar za pravljenje virtualnih mašina po potrebi.

Korisnik obezbeđuje sve nivoe softvera, uključujući i OS (sve što nije hardver)

- softver se implementira i pušta u rad praktično isto kao u stvarnim uslovima...
- ... ali na virtualizovanoj infrastrukturi

283. Objasniti vrstu oblaka „Platforma kao usluga“.

Poslužilac apstrahuje infrastrukturu i nudi platformu za izvršavanje aplikacija:

- aplikativni serveri
- SUBP
- drugi vidovi skladišnog prostora
- programi i aplikativni okviri
- ...

Ideja je da umesto čistog hardvera, dobijamo nešto na malo “višem” nivou kao što je aplikativni server. Primer: Ako želimo da pustimo veb aplikaciju u produkciju, nama je neophodan samo aplikativni server i može da bude praktičnije nego samo hardverski deo (korisnik ne mora da obezbedi OS).

Korisnik obezbeđuje razvijen aplikativni softver i konkretne strukture (baze) podataka

- softver se implementira i pušta u rad na poseban način...
- ... prilagođen okvirima koje nudi platforma

284. Objasniti vrstu oblaka „Softver kao usluga“.

Poslužilac nudi pojedine softverske usluge, koje mogu da se koriste i kao celina i pojedinačno:

- usluga elektronske pošte
- usluga plaćanja putem kreditne kartice
- usluga skladišnog prostora
- usluga stalno raspoložive veb lokacije
- ...

Korisnik ne implementira ništa (od tih usluga)

- samo ih konfiguriše prema potrebama...
- ... i koristi iz drugih aplikacija

285. Objasniti karakteristične aspekte bezbednosti računarstva u oblaku.

Aspekti:

- upravljanje identitetima
- fizička i lična bezbednost
- dostupnost podataka
- bezbednost aplikacija
- privatnost podataka
- pravni aspekti

286. Šta je pokretno računarstvo?

Pokretno računarstvo je svaki računarski sistem kod koga je neki od značajnih elemenata pokretan

- pokretni fizički elementi
 - računari
 - bazne stanice
- pokretni logički elementi
 - instance aplikacije
 - softverske komponente

287. U čemu je značaj pokretnog računarstva u odnosu na informacione sisteme?

Iz ugla informacionog sistema pokretno računarstvo

- proširuje oblast primene
- proširuje domen korisnika IS
- donosi nove izazove

Na primeru informacionog sistema fakulteta:

- Nove primene
 - stalna povezanost
 - redovnija komunikacija
 - automatsko identifikovanje
 - upotrebljivost na svakom mestu
 - podrška naučnim istraživanjima
 - ...
- novi korisnici
 - praktično svi, stalno, svuda
 - terenski rad
 - ...
- novi problemi

- povezivanje komponenti
- bezbednost
- ...

288. Šta je pokretni sistem za upravljanje bazama podataka? Objasniti osnovne razlike u odnosu na nepokretne sistema za upravljanje bazama podataka.

Pokretni SUBP pruža punu funkcionalnost SUBP-a uz pokretljivost korisnika i delova sistema

- korisnik može započeti transakciju bez obzira na svoj položaj
- garantuje se lokalna konzistentnost
- garantuje se globalna konzistentnost
- garantuje se mogućnost oporavka sistema

Osnovne osobine (koje razlikuju PSUBP od klasične DSUBP):

- geografska pokretljivost (klijenta)
 - postoje prostorna replikacija (vrednost zavisi od lokacije)
- mogućnost upostavljanja i raskidanja veze
- obrada podataka (na obe strane)
- bežična komunikacija
- transparentnost (obrada podataka na strani klijenta ne utiče na njegove komunikacije)
- skalabilnost

289. Objasniti odnos pokretljivosti pokretnih jedinica i načina izvršavanja transakcija.

- Pokretna jedinica može “proizvoljno” uspostavljati i prekidati vezu sa sistemom
- Veza se može prekinuti i neplanirano
- Pokretna jedinica može da promeni položaj u toku transakcije
- Pokretljivost proizvodi dodatne probleme u odnosu na obradu transakcija, a pre svega u odnosu na konzistentnost podataka

Uticaj pokretljivosti na atomičnost:

- Atomičnost: baza podataka nikada ne sme da sadrži podatke koji su proizvod samo delimično izvršene transakcije
- Pokretljivost ne utiče na pojam ali otežava implementaciju atomičnosti
- Dnevnici transakcija: Uobičajeno sredstvo za obezbeđivanje atomičnosti u centralizovanim sistemima, U slučaju pokretljivosti ne mogu da se primenjuju u istom obliku. Mogu da se vode na tri mesta:
 - MU – pokretna jedinica (mobile unit): nije dobro usled slabih mogućnosti pokretnih jedinica
 - BS – bazna stanica (base station): najčešće korišćeno rešenje

- MSC – centar za priključivanje mobilnih jedinica: (mobile switching center), nije dobro usled velikog ukupnog obima operacija u sistemu

Uticaj pokretljivosti na konzistentnost

- Baza podataka je u konzistentnom stanju ako su ispoštovani svi uslovi integriteta definisani za tu bazu podataka
- U centralizovanim i distribuiranim sistemima postoji samo jedna ispravna vrednost za svaki podatak, Replicirana baza podataka je u “uzajamno konzistentnom stanju” ako sve kopije podataka imaju iste vrednosti
- U pokretnim sistemima postoje dve vrste konzistentnosti:
 - prostorna konzistentnost: Sve vrednosti podataka prostorne replikacije se odnose na tačno jedan region i zadovoljavaju uslove konzistentnosti definisane za taj region
 - vremenska konzistentnost: Sve vremenske replike (na različitim položajima) podataka moraju imati iste vrednosti
 - Svaka pokretna jedinica koja započne transakciju u nekom regionu mora dobiti konzistentan pogled na region
 - Baza podataka mora da obezbedi trajnost efekata izvršavanja transakcije u tom regionu
 - Zbog toga region mora da zadovolji i prostornu i vremensku konzistentnost

Uticaj pokretljivosti na izolovanost

- Nijedna transakcija ne sme da ometa rad druge transakcije
- Svaki segment izvršavanja mora da zadovoljava uslove izolovanosti na nivou na kome se izvršava, izolovanost se može obezbeđivati nezavisno za različite regione, ali samo tako da bude ostvarena u odnosu na celovitu bazu podataka

Uticaj pokretljivosti na trajnost

- Sve potvrđene izmene moraju ostati trajno zapisane u bazi podataka.
- U slučaju pokretljivih sistema, trajnost se odnosi kako na globalne tako i na regionalne podatke.
- Na slučajeve prostornih i vremenskih replika odnose se (redom) regionalna i globalna trajnost podataka.

Uticaj pokretljivosti na potvrđivanje transakcija

- Koncept potvrđivanja se ne razlikuje od distribuiranih sistema
- Fragment izvršavanja zadovoljava potvrđivanje zavisno od položaja akko se sve operacije u tom fragmentu završavaju potvrđivanjem i postoji preslikavanje položaja u podatke.
- Tada sve operacije u fragmentu rade nad prostornom replikom koja odgovara konkretnom regionu
- Transakcija i njeno potvrđivanje se odvijaju na jednom položaju

- Skup podataka na kome se radi može da bude neposredno određen položajem ili da zavisi od položaja (“rezervisati 2 mesta u ribljem restoranu u krugu od 1km”, “rezervisati 2 mesta u ribljem restoranu u krugu od 1km od fakulteta”)
- Čitava transakcija je locirana u kopiji podataka koja se odnosi na jedan region
- Region se određuje na početku transakcije

Uticaj pokretljivosti na izvođenje transakcija

- Pokretna jedinica može da bude:
 - neprekidno povezana: mobilna jedinica je neprekidno povezana sa serverom baze podataka, opciono može da kešira potrebne podatke radi podizanja performansi ili ih svaki put ponovo tražiti od servera, ako je potrebno, može da uđe u pritajeni režim (engl. doze, ne komunicira ali sluša baznu stanicu) radi uštede el. energije i zatim da se ponovo aktivira, ovaj režim je skup za održavanje i obično nije neophodan za obrađivanje korisničkih poslova
 - nepovezana: nakon osvežavanja podataka mobilna jedinica samoinicijativno prekida vezu i nastavlja obradu podataka lokalno, u planirano (unapred određeno) vreme se povezuje i šalje sve potrebne podatke serveru, server primenjuje primljene podatke tako da se očuva globalna konzistentnost
 - povremeno povezana: slično kao nepovezana, veza se može prekinuti u bilo kom trenutku od strane sistema, korisnika ili usled spoljnih faktora, ne postoji planirano vreme uključivanja ili isključivanja veze korisno za pokretne agente koji rade sa korisnicima (isporuka pošiljki, uređivanje artikala u samoposlugama,...), povezivanje na zahtev je jedan vid povremene povezanosti

Problemi usled nepovezanosti

- Konzistentnost podataka se teško definiše za nepovezane i povremeno povezane pokretne jedinice
- Posledica je da nije trivijalno određivanje okvira i obima transakcija
- U zavisnosti od vrste problema: neke transakcije su bezbedne i u slučaju nepovezanosti, druge su rizične (nisu bezbedne)

Pokretni SUBP (MDS – Mobile DBMS) pruža punu funkcionalnost SUBP-a uz pokretljivost korisnika i delova sistema: korisnik može započeti transakciju bez obzira na svoj položaj, garantuje se lokalna konzistentnost, garantuje se globalna konzistentnost, garantuje se mogućnost oporavka sistema

Osnovne osobine PSUBP: geografska pokretljivost (klijenata), mogućnost uspostavljanja i raskidanja veze, obrada podataka (na obe strane), bežična komunikacija, transparentnost (obrada podataka na strani klijenta ne utiče na njegove komunikacije), skalabilnost

Uloge računara u sistemu

- Fiksno povezani računari predstavljaju: nepokretne čvorove (fixed host – FH), bazne stanice (base station – BS), stanice za podršku pokretljivosti (mobile support stations - MSS)

- Pokretne jedinice mogu biti: pokretni čvorovi (mobile host – MH), pokretne jedinice (mobile unit - MU)

Transakcije u PSUBP-u

- Različiti delovi transakcija se potencijalno izvršavaju na različitim računarima
- Neophodna je upotreba koordinatora i mehanizma dvofaznog potvrđivanja transakcija

290. Na šta se odnosi „etika u računarstvu“?

Etika se obično definiše kao skup moralnih normi kojima se osoba ili grupa rukovode u svom ponašanju.

Etika u računarstvu je skup moralnih normi kojima se osoba ili grupa koja se bavi istraživanjem i razvojem u oblasti računarske teorije i prakse. Postoje opšte i specifične norme (odnose se na specifičnost određenih oblasti)

291. Navesti bar 5 opštih etičkih principa etičkog koda ACM-a.

- doprinosi društvu i dobru čovečanstva, uz prihvatanje da su svi ljudi zainteresovani za računarstvo
- izbegavaj nanošenje štete
- budi častan i pouzdan
- budi fer i ne diskriminiši
- poštuju rad na novim idejama i pronalascima
- poštuju privatnost
- ceni poverljivost

292. Navesti bar 7 profesionalnih odgovornosti etičkog koda ACM-a.

- teži visokom kvalitetu procesa i proizvoda
- održavaj visoke standarde stručnosti i etičnosti
- upoznaj i poštuju pravila rada (zavisna od okruženja)
- prihvati i pruži profesionalnu ocenu
- pruži iscrpnu i temeljnu procenu rač.sistema uključujući i posledice i rizike
- obavljaj posao samo u oblasti za koju si kompetentan
- neguj javnu svest i razumevanje računarstva, srodnih tehnologija i njihovih posledica
- pristupaj računarskim i komunikacionim resursima samo ako si ovlašćen ili kada je to neophodno za javno dobro
- projektuj i implementiraj sisteme koji su bezbedni

293. Zašto su etička pitanja posebno važna u oblasti informacionih sistema?

- radi se sa ljudima
- radi se za ljude
- često IS imaju širi uticaj nego što je lokalna upotreba
- često neposredno utiču na uslove profesionalnog života
- javni IS utiču i na uslove javnog života
- većina normi zbog svega toga dobija posebnu težinu