

Pitanja za PIRIS

DOMACI 1

Podatak je činjenica koja ima neko značenje, i obično je karakterise subjektivna percepcija. – Pogrešno

U osnovne elemente IS spadaju: Baza podataka, Softver, Hardver, Telekomunikacioni podsistem

Povezati aktivnost sa odgovarajućom fazom razvoja IS:

Specifikacija potrebnih ulaganja → Planiranje

Izrada/nabavka softvera i formiranje baze podataka → Implementacija

Analiza izvodljivosti projekta → Planiranje

Izbor i instalacija računarske opreme → Implementacija

Definisanje ciljeva i zadataka IS → Planiranje

Logičko i fizičko projektovanje → Analiza i dizajn

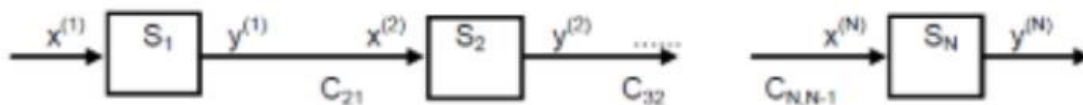
Obuka korisnika → Implementacija

Definisanje performansi sistema → Analiza i dizajn

Za koji model životnog ciklusa razvoja IS važi da se razvija tako što se razdvajaju i razvijaju pojedine funkcije IS i vrši njihovo isprobavanje, počevši od jednostavnijih ka složenijim. Svaka sledeća prototip funkcija je bolja i složenija od prethodne. – evolutivni

U menadžerske IS spadaju: DSS, MIS, ESS

Na slici (ispod) je prikazan otvoreni lanac povezivanja podsistema.



Subjektivnost je vezana za podatke. – Pogrešno

IS se prema namjeni mogu podijeliti u dvije velike grupe: operativni i menadžerski

Osnovni resursi svakog IS su: Ljudski resursi, podaci, softver, mreza za prenos podataka

Skup međusobno povezanih komponenti koje zajedno služe za prikupljanje, preuzimanje, obradu, skladištenje i distribuciju informacija u cilju lakšeg planiranja, kontrole, koordinacije, analize i donošenja odluka u poslovnim organizacijama je – informacioni sistem

"Crveno", "žuto", "zeleno", "čekanje" su informacije. – Pogrešno

Softirati faze razvoja IS sa njihovim redoslijedom izvršavanja.

1. Planiranje
2. Analiza i dizajn
3. Implementacija

4. Funkcionisanje i održavanje
5. Vrednovanje i kontrola

DOMACI 2

Pri projektovanju logickog modela neophodno je detaljno specificovati potrebne resurse – Pogresno

Odluka za odbijanje placa razvoja IS znaci: Moguce je da se predlozeni plan koristi u narednom periodu i Mogucu dopunu predstavljenog plana dodatnim informacijama

Prevodjenje fizickog projekta na konkretnu opremu, konkretne softvere koji ukljucuju konkretne baze podataka je logicko projektovanje – Pogresno

Obicno su troskovi održavanja softvera veci od troskova održavanja hardvera, narocito kod starijih sistema. – Tacno

Logicko projektovanje je dio faze: - Analize i dizajna IS

Povezati aktivnosti sa odgovarajucom fazom razvoja:

Izbor i instalacija opreme, izrada ili nabavka softvera i formiranje baze podataka -

>Implementacija

Provjera, ciscenje, podesavanje i zamjena potrošnog materijala, dijelova uređaja ili cijelih uređaja, prepravke i azuriranje softvera -> Funkcionisanje i održavanje

Fizicko i logicko projektovanje sistema -> Analiza i dizajn

Specifikacija ulaganja i analiza izvodljivosti -> Planiranje razvoja

Definisanje objektivnog seta kriterijuma za utvrđivanje uticaja IS na poslovanje, brzine obavljanja funkcija, pogodnosti upotrebe i dr. -> Vrednovanje i kontrola

Prevodjenje logickog projekta na konkretnu opremu, konkretne softvere koji ukljucuju konkretne baze podataka je fizicko projektovanje. – Tacno

Povezati odgovarajuće elemente IS sa vrstom modela (dizajna) kome pripada:

Skladištenje: CD, server, cloud -> Fizicki dizajn

Obrada: PC, operativni sistem, aplikativni softver -> Fizicki dizajn

Izlaz: sadržaj, format, organizacija, kolicina, frekvencija -> Logicki dizajn

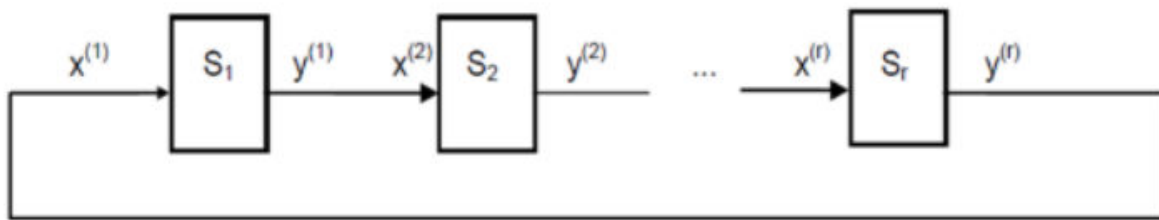
Kontrola: sifre, pristup, log-> Fizicki dizajn

Procedure: backup, unos podataka, način korišćenja-> Fizicki dizajn

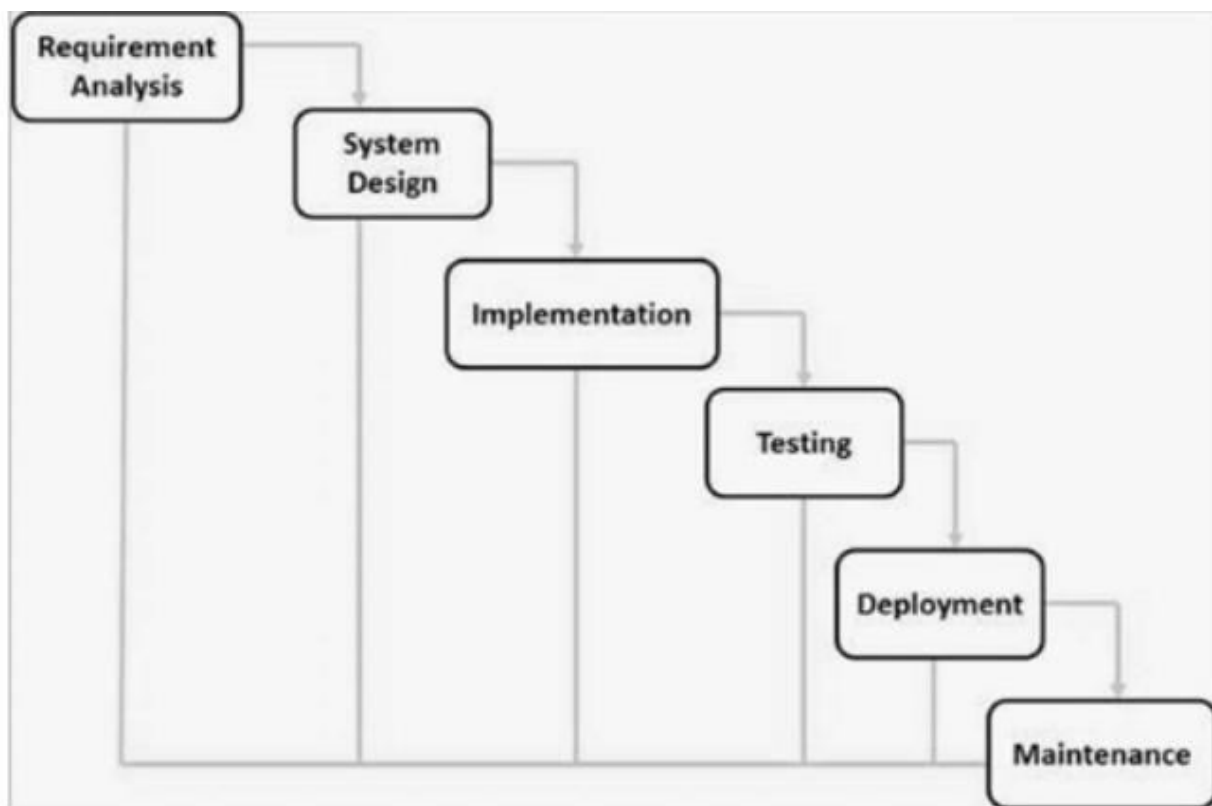
Procedure: aktivnosti, pravila, lokacija, vrijeme-> Logicki dizajn

DOMACI 3

Na slici (ispod) je prikazan **zatvoreni** lanac povezivanja podsistema



Na slici (ispod) je prikazan model **vodopada** linearnog zivotnog ciklusa



Povezati IS sa kategorijom kojoj pripada:

Sistemi kontrole procesa -> Operativni IS

Sistemi automatizacije kancelarije -> Menadzerski IS

Upravljacki IS -> Menadzerski IS

Sistemi podrške odlucivanju -> Menadzerski IS

Online sistemi transakcione obrade podataka -> Operativni IS

IS zasnovani na znanju-> Menadzerski IS

DOMACI 4

Taktika prikupljanja informacija kojom korisnik traži više mogućnosti, a zapravo mu je potrebna samo jedna ili dvije, te dodatne zahtjeve koriste za postizanje bolje nagodbe sa analitičarom se naziva: Taktika dimne zavjese

"Softverski proizvod će ispisati statusnu poruku u redovnim intervalima, ne manjim od 60 sekundi" je primjer: Nepotpunog zahtjeva

Za intervju važi:

- treba pitati o svemu vaznom
- sprovodi se kroz usmeno ispitivanje
- dopustiva su pitanja otvorenog i zatvorenog tipa

Taktika prikupljanja informacija kojom korisnik navodi (preko)brojne potrebe, gomilu nepotrebnih izvještaja, ekranskih formi, sortiranja, izračunavanja i sl.: a ----> taktika kuhinjskog sudopera

Tehnički stručnjak koji oblikuje sistem tako da zadovolji zahtjeve korisnika, prevodi poslovne zahtjeve i ograničenja u tehničko rješenje, oblikuje datoteke, baze podataka, ulaze, izlaze, ekranske forme, mrežu i programe, integriše rješenje je: ----> dizajner sistema

Zahtjevi koji opisuju zadatke koje korisnik mora obaviti služeći se aplikacijama spadaju u kategoriju: ----> korisnicki zahtjevi

Zapisnik treba da sadrži:

- Vrijeme i mjesto održavanja razgovora
- Sadržaj razgovora
- Naziv projekta
- Spisak uesnika i ime zaspinicara

Opisuje podatke i veze između podataka → **Konceptualni model**

Ko obrađuje podatke, gdje se podaci nalaze i gdje se podaci obrađuju → **Model resursa/sredstava**

Kako se prikupljaju, obrađuju i distribuiraju podaci → **Model funkcija**

Opisuje strukturu podataka i logičkih datoteka → **Logički model**

Šta su podaci i šta oni opisuju → **Model podataka**

Kada se podaci obrađuju, odnosno razmatra učinke koje događaji.... → **Model događaja**

Primjer: "Parser će brzo generisati izvještaj o greškama HTML oznaka, koji omogućava brzu ispravku grešaka kada program koriste početnici u HTML-u" → **NEODREDJENOG ZAHTJEVA**

Za upitnik važi:

- **SPROVODI SE U PISANOJ FORMI**
- **DOZVOLJENA SU PITANJA ZATVORENOG I OTVORENOG TIP**

Taktika prikupljanja informacija kojom korisnik traži više mogućnosti, a zapravo mu je potrebna samo jedna ili dvije, te dodatne zahtjeve koriste za postizanje bolje nagodbe sa analitičarom se naziva: → **TAKTIKA DIMNE ZAVJESE**

- ❖ Svojstvo koje sistem mora da posjeduje → **Nuzno svojstvo sistema**
- ❖ Proizvoljni zahtjevi, svojstva i mogućnosti bez kojih se može, iako bi bilo lijepo ih imati → **Neobavezna svojstva sistema**
- ❖ Funkcije koje korisnik želi da ima u sistemu → **Pozeljno svojstvo sistema**

Primjer: "Softverski proizvod će se trenutno prebaciti između ispisivanja i skrivanja oznaka koje se ne mogu štampati" predstavlja primjer: → **NEOSTVARIVOG ZAHTJEVA**

Ako sistem ne uključuje nužne zahtjeve, taj sistem ne može ispuniti svoju svrhu, ne može da prođe validaciju. → **TACNO**

Provjerava njegovu ispravnost te ga isporučuje u primjenu, konstruiše komponente sistema na osnovu utvrđenih specifikacija → **Graditelj sistema(programer)**

Naručuje i plaća razvoj i održavanje sistema, postavlja prioritete i određuje politiku njegovog korišćenja → **Vlasnik sistema**

Razumiju i poslovanje i računarsku obradu podataka, provode sistemsku analizu i dizajn. Povezuju one koji trebaju računar i one koji poznaju informacione tehnologije → **Sistem analiticari**

Neposredno koristi IS pri obavljanju svakodnevnih poslova ili koristi informacije dobijene iz IS
→ Korisnik

Tehnički stručnjak koji oblikuje sistem tako da zadovolji zahtjeve korisnika, prevodi poslovne zahtjeve i ograničenja u tehničko rješenje, oblikuje datoteke, baze podataka, ulaze, izlaze, ekranske forme, mrežu i programe, integriše rješenje, a može biti i graditelj sistema →
Projektant

DOMACI 5

Povezati pojmove i definicije

Logički dio posla koji se obavlja kao nedjeljiva cjelina -> Dogadjaj

Postupak, način rada, doslijedna izmjena stanja, aktivnost ili zadatak kojim se obavlja neki posao -> Proces

Skup logički povezanih trajnih poslovnih aktivnosti i zadataka. koje se stalno obavljaju -> Funkcija

Po IDEF metodologiji, upravljanje se odnosi na: Ogranicenje na izvršavanje procesa

Pravila modelovanja funkcije:

- dijete smije da ima samo jednog roditelja

- roditelj mora imati barem dvoje

- svaki proces je roditelj ili dijete

Kod IDEFO bloka pojedini elementi su:

- Lijeva -> ulaz

- Desna -> izlaz

- Gornja -> upravljanje

- Donja -> mehanizam

Po IDEF metodologiji, mehanizam se odnosi na: nešto što se koristi u procesu, ali se ne troši

Razdvajanje funkcije na njene komponente se naziva -> Dekompozicija

Funkcija je logički dio posla koji se obavlja kao jedna nedjeljiva cjelina. - Pogrešno

Pri modelovanju funkcije, jedno od pravila je da dijete može da ima više od jednog roditelja. □
Pogrešno

Po IDEF metodologiji, izlaz se odnosi na: rezultat procesa

Dekompozicija funkcija je: razdvajanje f-je na njene komponente

Za IDEF3 model, dijagramu opisa procesa pripadaju:

Aktivnost, Logika, Ponašanje, Prelaz stanja, Stanje objekta

DOMACI 6

Kod mrežnog modela podataka jedan isti objekat može biti i nadređeni i podređeni. - Tacno

Model podataka je polazna osnova za: projektovanje i realizaciju baze podataka

Posmatranje skupa objekata i njihovih veza kao objekat višeg nivo je: Agregacija

Komponente modela podataka kojima se iskazuje dinamičnost realnog sistema, a time i prelazak baze podataka iz jednog stanja u drugo su: Operatori

Kontrolisano uključivanje detalja u opis sistema, izostavljanje detalja u pojedinim fazama i predstavljanje samo zajedničkih opštih osobina pojedinih koncepata, pri čemu se zanemaruju nebitne osobine realnog sistema a takođe i nebitne veze, tako da se u model unose samo one opšte, bitne jer model treba da odslika realni sistem je: Apstrakcija

Logičke granice postavljene kako na pojedine podatke tako i na njihove međusobne veze, kojima se iskazuju dozvoljena stanja sistema i dozvoljeni načini prelaska iz stanja u stanje predstavljaju ograničenja modela podataka

skup podataka strukturisanih na način opisan u šemi, skup svih pojavljivanja entiteta navedenih kategorija, njihovih osobina i veza između njih → baza podataka

skup generickih pravila pomocu kojih je moguće definisati semu → jezik za definiciju podataka jedinstveni način predstavljanja skupa podataka i njihove interpretacije preko strukture podataka, skupa ograničenja i skupa operatora → model podataka

skup klasa ili kategorija, njihovih osobina i veza između njih → sema modela podataka

Jedan ili više atributa koji jednoznačno određuju svaki entitet u klasi entiteta predstavlja ključ klase entiteta.

DOMACI 7

Relacioni model podataka obično koristi tabelarni model prikazivanja podataka. – Tacno

Model podataka koji omogućava vrlo jednostavan i prirodan prikaz podataka skupom tabela koje se mogu definisati kao matematička relacija je: Relacioni model podataka

Relacioni model koji podržava samo koncepte strukture i pravila integriteta a nema skup operatora naziva se: semiracionalni model podataka

Karakteristike relacionog modela podataka su: ne postoje jednaki redovi u tabelama, poredak kolona nije vazan, sve vrijednosti u tabeli imaju „atomske“ karakter tj. ne mogu se razbiti na komponente

Prilikom prevođenja iz ER modela u relacioni model važi:
ime tipa entiteta postaje -> ime seme relacije
svaki objekat postaje -> seme relacije
identifikator objekta (entiteta) postaje -> primarni ključ seme relacije

DOMACI 8

Korisnički pomoćni elementi koji vode korisnika korak po korak kroz proceduru su: čarobnjaci (wizard)

Korisnički interfejs se može smatrati jednim od najvažnijih dijelova poslovne aplikacije, zato što najveći broj korisnika koristi aplikaciju. - Tačno

Pitanje "Kako će korisnici da budu u interakciji sa sistemom?" predstavlja pitanje koje pripada: Dizajnu korisničkog interfejsa

Omogućava brzo i lako istraživanje alternativnog dizajna -> nisko vjerno dizajniranje
Odličan za razmišljanje i brza povratna informacija sa jasnom identifikacijom problema dizajna -> nisko vjerno dizajniranje

Pokazuje detaljni izgled ekrana i elemente interfejsa -> nisko vjerno dizajniranje

Pokazuje glavne osobine, strukture i navigaciju -> nisko vjerno dizajniranje

Brza primjena i izmjena ali zahtjeva kompjuter i softver -> visoko vjerno dizajniranje

Obično se razvija iz nisko vjernog dizajna -> visoko vjerno dizajniranje

U dizajnu sistema, mogućnost da se sistemom upravlja na svim nivoima je: Upravljivost

DOMACI 9

Otkrivanje informacija je primjer narušavanja bezbjednosti kada napadač šalje e-mail poruke koje su bile napisane od nekog drugog. → POGRESNO

Bezbedonosni mehanizmi za zaštitu sistema/aplikacije su:

- SHEMA AUTENTIFIKACIJE
- PROXI
- FIREWALL

Ranjivost je slaba tačka u bezbjednosti, tako da napadač koristi mogućnost neautorizovanog pristupa u mrežu organizacije ili resursa na mreži. → TACNO

Kada maliciozni korisnik izmijeni rad sistema/aplikacije, konfiguraciju ili podatke. → Kvarenje
Sistem administrator ili osoba zadužena za bezbjednost nije u mogućnosti da provjeri
malicioznog korisnika/napadača. → Odricanje
Napadač krade ili pogađa password poverljivog korisnika, koji koristi za pristupi zabranjenim
resursima. → Laziranje identiteta
neautorizovani korisnik može da vidi privatne podatke → otkrivanje informacija
napad kojim se pokušava ugasiti ili zabraniti pristup resursima i kompajliranje → DoS
korisnik dobije neautorizovani pristup kompjuteru i onda izmijeni rad, konfiguraciju ili podatke →
kvarenje

DOMACI 10

CASE tehnologije su namijenjene automatizaciji procesa razvoja softverskih proizvoda. → tacno

Projektovanje logičke strukture IS: logika prikupljanja, obrade i izdavanja informacija kao i
logička struktura tehničko - tehnoloških resursa potrebnih za realizaciju projektovanog sistema
je: → idejni projekat

Naručuje se jedan projekat IS i sa njim se jasno definiše zadatak projektnog tima

Projektovanje logičke strukture IS: logika prikupljanja, obrade i izdavanja informacija kao i
logička struktura tehničko - tehnoloških resursa potrebnih za realizaciju projektovanog sistema
→ Idejni projekat

Konkretizacija rješenja informacionog sistema koje je dato u idejnom projektu → Glavni
projekat

CASE alati koji automatizuju faze programiranja, implementacije, eksploatacije i održavanja su:
→ programerski case

CASE tehnologije namijenjene za automatizaciju faze analize i dizajna → middle CASE

CASE tehnologije namijenjene za automatizaciju faze programiranja, testiranja i implementacije
sistema → lower CASE

CASE tehnologije namijenjene za automatizaciju faze strategijskog planiranja sistema i
upravljanja projektima → upper CASE