***Software Engineering – SWE – Softversko Inžinjerstvo***

***– Pitanja za usmeni deo ispita –***

**I Nedelja**

1. **Šta je software?**
   * Softver je računarski program, pridružena dokumentacija i konfiguracioni podaci neophodni da bi softver radio korktno.
   * Softverski sistem se obično sastoji od:
     + Određenog broja programa
     + Konfiguracionih fajlova
     + Sistemske dokumentacije koja opisuje strukturu sistema
     + Korisničke dokumentacije
     + Web site-ova za podršku korisnicima
2. **Koji su tipovi SW proizvoda?**
   * Postoje dva osnovna tipa SW:
     + Generički (generic) – samostalni sistemi namenjeni za prodaju na slobodnom tržištu
     + Ugovorni (custom) – razvijeni za jednog korisnika prema njegovim zahtevima
3. **Šta je softversko inžinjerstvo?**
   * Softversko inžinjerstvo je inžinjerska disciplina koja se bavi: teorijom, metodama i alatima za profesionalni razvoj softvera.
   * Softversko inžnjerstvo je inžinjerska disicplina koja obuhvata sve aspekte proizvodnje softvera
4. **Razlozi za neuspeh projekata?**
   * Nedostatak kompletne specifikacije zahteva
   * Nedostatak odgovarajuće metodologije razvoja
   * Loša dekompozicija projekta na komponente kojima se uspešno može upravljati
5. **Koja je razlika između SWE i informatike?**
   * Osnovna razlika je da se informatika (nauka o računarstvu) bavi teorijom i osnovama računarstva, dok se softversko inžinjerstvo bavi praktičnom stranom isporuke korisnog softvera.
   * Informatika je osnova za softversko inžinjerstvo
6. **Koja je razlika između SWE i sistemskog inžinjerstva?**
   * Sistem čine softver, hardver i ljudi
   * Sistemsko inžinjerstvo se bavi svim aspektima razvoja sistema zasnovanih na računarima, uključujući hardverski, softverski i procesni inžinjering
     + Softversko inžinjerstvo je deo tog procesa
   * Sistemsko inžinjerstvo je starija disciplina od SWE.
7. **Šta je softverski proces?**
   * Softverski proces je skup aktivnosti i priduženih rezultata čiji je cilj proizvodnja softvera.
   * Aktivnosti zajedničke za sve softverske procese su:
     + Specifikacija softvera (Software Specification)
     + Razvoj softvera (Software Development)
     + Validacija softvera (Software Validation)
     + Evolucija softvera (Software Evolution)
8. **Šta je model softverskog procesa?**
   * Model softverskog procesa je uprošćena reprezentacija softverskog procesa koja predstvalja jedan pogled na ovaj proces.
   * Opšti modeli (paradigme) su:
     + Model vodopada
     + Iterativni razvoj
     + Razvoj zasnovan na korišćenju gotovih komponenti
9. **Šta su metode SWE?**
   * Metode SWE-a su struktuirani prilaz razvoju softvera.
   * Metode obuhvataju:
     + Opise modela (opisi grafičkih modela koji nastaju u toku razvoja),
     + Pravila (ograničenja primenjena na model sistema),
     + Preporuke (saveti za dobro projektovanje) i
     + Vodič kroz proces (kako teku aktivnosti)
10. **Šta su CASE alati?**
    * CASE (Computer Aid Software Engineering) alati su softverski sistemi namenjeni pružanju automatske podrške aktivnostima softverskog procesa
    * Case višeg nivoa (Upper-CASE)
      + Alati koji podržavaju aktivnosti procesa (inžinjering zahteva i projektovanje)
    * Case nižeg nivoa (Lower-CASE)
      + Alati koji podržavaju kasnije aktivnosti (programiranje, debagiranje i testiranje)
11. **Koji su atributi dobrog softvera?**
    * Pogodnost za održavanje (Maintanability)
      + Treba da je u stanju da može da se lako menja
    * Stabilnost (Dependability)
      + Mora da ulica poverenje što podrazumeva da je:
        1. Pouzdan (Reliability)
        2. Bezbednan (Security)
        3. Siguran (Safety)
    * Efikasnost (Efficiency)
      + Mora da ekonomično koristi resurse sistema
    * Upotrebljivost (Usability)
      + Mora da bude pogodan za korišćenje
12. **Koji su ključni izazovi SWE?**
    * Postojeći sistemi (Legacy Systems)
      + Stari, postojeći sistemi koje treba održavati i ažurirati
    * Sve veća raznovrsnost (Heterogenity)
      + Sistemi su distribuirani i ukljuuju raznovrstan hardver i softver
    * Isporuka (Delivery)
      + Zahtevi da se skrati vreme izrade softvera

**II Nedelja**

1. **Šta je sistemsko inžinjerstvo?**
   * Sistemsko inžinjerstvo podrazumeva projektovanje, implementacije, isporuku i eksploataciju sistema koji uključuju hardver, softver i ljude.
2. **Šta je sistem?**
   * Sistem je kolekcija međusobno povezanih komponenti koje rade zajedno radi ostvarenja nekog zajedničkog cilja.
   * Sistem obuhvata softver, hardver (mehaničke, električne i elektronske komponente) i ljude.
3. **Koje su kategorije sistema?**
   * Tehnički sistemi zasnovani na računaru - Sistemi koji uključuju hardver, softver, ali ne uključuju procese i ljude.
   * Socio-tehnički sistemi – Sistemi koji uključuju jedan ili više tehničkih sistema, ali takođe i znanje o tome kako koristiti sistem da bi se postigao cilj. Definišu radne procese (uključujući operatore kao inherentni deo sistema) koji su prilagođeni poslovnim i drugim procesima ogranizacije kojoj su namenjeni
4. **Koji je odnos softverskog i sistemskog inžinjerstva?**
   * Softver je deo sistemskog inžinjerstva.
   * Udeo softvera u sistemima je sve veći
   * Probelmi sistemskog inžinjerstva su slični problemima softverskog inžinjerstva
   * Softver se vidi kao problem u sistemskom inžinjerstvu (nažalost)
5. **Šta su dodatna svojstva sistema?**
   * Dodatna sovjstva (Emergent System Properties) su sva svojstva sistema koja se dobijaju tek kada se kompletira sistem
   * Reč je o svojstvima sistema kao celine
   * Dodatna svojstva su posledica veza između komponenti sistema
   * Mogu se utvrditi i neke čak i meriti tek nakon integracije komponenti u sistem
6. **Koji su primeri dodatnih svojstava sistema?**
   * Ukupna težina sistema – može se izračunati na osnovu svojstava pojedinih komponenti
   * Pouzdanost sistema – zavisi od pouzdanosti komponenti sistema i veza među komponentama
   * Upotrebljivost sistema – složeno svojstvo koje ne zavisi samo od hardvera i softvera sistema, već i od operatora sistema i okruženja gde se sistem koristi
7. **Tipovi dodatnih svojstava?**
   * Funkcionalna svojstva – Javljaju se kad svi delovi sistema rade zajedno da bi ostvarili neki cilj
   * Nefunkcionalna svojstva – Odnose se na ponašanje sistema u radnom okruženju (pouzdanost, performanse, sigurnost, bezbednost) [kritična za sisteme zasnovane na računaru]
8. **Šta je pouzdanost sistema?**
   * Pouzdanost sistema je
9. **Šta utiče na pouzdanost?**
   * Pouzdanost hardvera – verovatnoća da hardverska komponenta bude neispravna i koliko dugo traje reparacija te komponente
   * Pouzdanost softvera – verovatnoća da neka softverska komponenta proizvede neispravan izlaz (softver se za razliku od hardvera ne može pohabati)
   * Pouzdanost operatora – verovatnoća da operator načini grešku
   * Veze između komponenti (njihova interakcija)
   * Okruženje u kome je sistem instaliran
10. **Šta su nepoženje karakteristike?**
    * Nepoželjne karakteristike (“Shall-not” svojstva) su stavke koje sistem ne sme dopustiti kako bi obezbedio svojstva sistema koja se ne mogu meriti, poput sigurnosti i bezbednosti.
11. **Kako se radi modelovanje arhitekture sistema?**
    * Arhitekturni model predstavlja apstraktni pogled na podsisteme koji čine sistem
    * Može uključiti glavne tokove podataka među podsistemima
    * Obično se predstavlja blok dijagramima
    * U modelu se mogu identifikovati različiti tipovi funkcionalnih komponenti
12. **Koji su osnovni tipovi komponenti u alarmnom sistemu:**
    * Senzori
    * Aktuatori
    * Komunikacija
    * Koordinacija (kontroler)
    * Interfejs
13. **Tipovi komponenti (svakog) sistema**
    * Senzorske komponente – Prikupljaju informacije iz okruženja
    * Aktuatorske komponente – Izazivaju neke promene u okruženju
    * Računarske komponente – Transformišu ulaze u izlaze
    * Komunikacioni komponente – Omogućavaju komunikaciju između komponenti sistema
    * Koordinacione komponente – Koordiniraju interakciju među komponentama sistema
    * Interfejsne komponente – Olakšavaju interakciju sa ostalim komponentama sistema
14. **Proces softverskog inžinjerstva?**

Definisanje zahteva

Projektovanje sistema

Razvoj podsistema

Integracija sistema

Instalacija sistema

Evolucija sistema

Rasturanje sistema

**Model vodopada**

* + Definiše životni ciklus jednog sistema
  + Obično se koristi model vodopada budući da treba da se paralelno razvijaju različiti delovi sistema
  + Model vodopada uključuje specificiranje, projektovanje, razvoj i integraciju.

1. **Opis faza modela vodopada**
   * Definisanje zahteva sistema:
     + Definišu se 3 vrste zahteva:
       1. Apstraktni funkcionalni zahtevi – funkcije sistema se definišu na apstraktan način
       2. Svojstva sistema – definišu se nefunkcionalni zahtevi za sistem kao celinu
       3. Nepoželjne karakteristike – specificira se šta sistem ne sme da radi
     + Definisanje organizacionih ciljeva sistema kao celine
       1. Funkcionalni (npr. da obezbedi sistem za obaveštenje o greški koja je nastala)
       2. Organizacioni (npr. da greška u delu sistema ne obori ceo sistem)
     + Problemi (javljaju se nakon specificiranja sistema)
   * Proces projektovanja sistema
     + Podfaze:
       1. Particionisanje zahteva – organizovati zahteve u grupe srodnih zahteva
       2. Identifikovanje podsistema – identifikovanje skupa podsistema koji zajedno mogu da ostvare sistemske zahteve
       3. Vezivanje zahteva za podsisteme – raspodela zahteva po podsistemima (značajno kod korišćenja gotovih komponenti)
       4. Specificiranje funkcionalnosti podsistema
       5. Definisanje interfejsa podsistema – kritična aktivnost za paralelni razvoj podsistema
     + Problemi (podela zahteva hw/sw/people, nekompatibilni hw i sw)
   * Razvoj podsistema
     + Obično se hw, sw i komunikacione komponente paralelno razvijaju (naveden u prethodnoj fazi)
     + Može zahtevati nabavku gotovih komponenti (Commercial Off-The-Shelf COTS) - isplativije
     + Nedostatak komunikacije unutar tima
     + Birokratizacija i spori mehanizmi za predlaganje promena sistema mogu dovesti do toga da se planirano vreme razvoja prekorači
   * Integracija sistema
     + Proces objedinjavanja svih komponenti (hw, sw, people) u jedinstveni sistem
     + Treba ga izvesti inkrementalno, tj. Integrisati jednu po jednu komponentu
     + U ovom koraku se otkirvaju problemi u interfejsu među komponentama
     + Može biti problem u nekoordinisanoj isporuci pojedinih komponenti sistema
   * Instalacija sistema
     + Mogu biti nekorektne pretpostavke o okruženju sistema
     + Može se javiti otpor ljudi da prihvate novi sistem
     + Novi sistem treba jedno vreme da radi zajedno sa postojećim
     + Mogu se javiti fizički problemi u instalacijama (npr. kabliranje)
     + Treba izvršiti obuku operatera
   * Eksploatacija sistema
     + Mogu se pojaviti novi zahtevi
     + Korisnici mogu koristiti sitem na način koji nisu predvideli projektanti
     + Može se javiti porblem u interakciji sa drugim sistemima
       1. Fizički problemi nekompatibilnosti
       2. Problemi konverzije podataka
       3. Povećanje greške operatera zbog lošeg interjesa
   * Evolucija sistema
     + Veliki sistemi imaju dug životni vek. Oni moraju evoluirati da bi zadovoljili nove zahteve
     + Evolucija je skupa
       1. Promene treba analizirati sa tehničke i poslovne strane
       2. Mogu sde javiti problemi u podsistemima koji interaguju sa sistemom koji se menja
       3. Kada se vrše promene narušava se struktura sistema
     + Postojeći sistemi koji se moraju održavati se nazivaju **legacy systems**
   * Rasturanje sistema
     + Stavljanje sistema van upotrebe nakon što se završi njegov životni vek
     + Može zahtevati odlaganje materijala (npr. opasne hemikalije) koji zagađuju okolinu
       1. Pri projektovanju sistema mora se predvideti šta će se raditi sa takvim materijalom
     + Može se zahtevati rekonstrukcija i konverzija podataka na drugi sistem
2. **Šta je nabavka sistema?**
   * Proces nabavke sistema uključuje donošenje odluke koji sistem kupiti i od koga
   * Traženje sistema koji zadovoljava potrebe organizacije
   * Pre nabavke je neophodno izvesti u nekom obliku specijalno i arhitekturno projektovanje sistema
     + Potrebna vam je specifikacija da biste ugovorili razvoj sistema
     + Specifikacija vam može omogućiti kupovinu komercijalnih COTS sistema. Gotovo uvek je to jeftinije nego razvijati sistem.
3. **Načini nabavke sistema?**

Istraživanje tržišta sistema

Adaptiranje zahteva

Izbor sistema

Traženje ponuda

Izbor dobavljača

Objavljivanje tendera

Selekcija izvođača

Ugovaranje

Potpisivanje ugovora

**I Način: Postoje gotovi sistemi**

**II Način: Potrebno je razviti sistem po meri**

**Proces nabavke sistema**

**III Nedelja**

1. **Šta je softverski proces i koje su aktivnosti zajedničke za različite softverske procese?**
   * Sotfverski proces je struktuiran skup aktivnosti neophodnih za razvoj softverskog sistema.
   * Aktivnosti zajedničke za sve softverske procese su:
     + Specifikacija – Definisanje šta sistem treba da radi,
     + Projektovanje i implementacija – Definisanje organizacije sistema i njegova implementacija
     + Validacija – Provera da li sistem radi ono što naručilac želi
     + Evolucija – Promena sistema kao odgovor na promenu potreba naručioca
2. **Model vodopada. Grafički predstaviti i navesti osnovne karakteristike sa prednostima i nedostacima**

Analiza i specifikacija zahteva

Projektovanje sistema i softvera

Implementacija i testiranje delova

Integracija i testiranje sistema

Eskploatacija i održavanje

* Faze u modelu vodopada:
  + Analiza i specifikacija zahteva
  + Projektovanje sistema i softvera
  + Implementacija i testiranje delova
  + Integracija i testiranje sistema
  + Eksploatacija i održavanje
* Osnovni nedostatak ovog modela je nemogućnost efikasnog prihvatanja izmena u zahtevima korisnika kada je proces u toku. U principu, jedna faza mora biti završena da bi se otpočela naredna, pa izmene zahteva resetuju čitav proces.
* Prednost je kada se ovaj model koristi pri projektovanju velikih sistema, gde razvoj razdeljen na nekoliko lokacija olakšava projektovanje (Divide and conquer)
* Primer kada gradimo kuću vs kada projektujemo softver

1. **Model inkrementalnog razvoja**

Okvirni opis

Inicijalna verzija

Konačna verzija

Međuverzije

Specifikacija

Razvoj

Validacija

* Prednosti inkrementalnog razvoja:
  + Smanjena cena izmene korisničkih zahteva u toku razvoja
    - Količina analiza i dokumentacije koju treba ponovno uraditi je mnogo manja u odnosu na model vodopada.
  + Olakšano dobijanje povratne informacije od korisnika u toku razvoja
    - Korisnici imaju uvid i mogućnost komentarisanja trenutno implementiranih funkcionalnosti, a samim tim i u progres realizacije projekta.
  + Moguća brža isporuka korisnog softvera naručiocu:
    - Korisnici mogu da koriste softver pre nego u slučaju mdoela vodopada.
* Problemi inkrementalnog razvoja:
  + Proces nije deljiv
    - Menazžerima su potrebne redovne isporuke kako bi merili napredovanje. Ako se sistem brzo razvija nije efikasno praviti dokumentaciju za svaku verziju sistema.
  + Struktura sistema ima tendenciju degradacije sa dodavanjem novih inkremenata
    - Ukoliko se ne ulaže novac u refaktorisanje i unapređenje softvera, periodične izmene imaju za posledicu degradaciju strukture. Nove i nove izmene softvera postaju sve složenije i skuplje.

1. **Razvoj softvera zasnovan na korišćenju gotovih komponenti?**
   * Zasnovan na sistematskom korišćenju ranije razvijenih komponenti ili komercijalno dostupnih komponenti (COTS – Commercial-off-the-shelf).
   * Faze procesa:
     + Analiza komponenti
     + Modifikacija zahteva
     + Projektovanje sistema korišćenjem gotovih komponenti
     + Razvoj i integracija
   * Korišćenje gotovih komponenti je standardni pristup za razvoj mnogih poslovnih sistema.

Specifikacija zahteva

Projektovanje zasnovano na korišćenju gotovih komponenti

Razvoj i integracija

Modifikacija zahteva

Validacija sistema

Analiza gotovih komponenti

1. **Prototipovanje softvera**
   * Prototip je inicijalna verzija sistema koja se koristi u cilju demonstracije koncepata i isprobavanja projektnih opcija.
   * Prototip se može koristiti u:
     + Procesu inžinjeringa zahteva kako bi potpomogao prikupljanje i validaciju zahteva
     + U procesu projektovanja kako bi se istražile moguće opcije i razvio korisnički interfejs
     + U procesu testiranja kako bi se obezbedili uporedi testovi