

TEMELJNA PITANJA OKO PROGRAMSKOG INŽENJERSTVA

1. Što je programska potpora (software) ?

PP je računalni program i pridružena dokumentacija.

2. Što je programsko inženjerstvo ?

Disciplina koja se bavi metodama i alatima za profesionalno oblikovanje i produkciju programske potpore uzimajući u obzir cijenu.

3. Koje je razlika između programskog inženjerstva i računarske znanosti ?

RZ se bavi teorijskim dijelom, PI se bavi praktičnom primjenom.

4. Koja je razlika između programskog inženjerstva i inženjerstva sustava ?

Inženjerstvo sustava se bavi svim aspektima sustava dok PI se bavi razvojem programske infrastrukture, upravljanja te primjene.

5. Što je proces programskog inženjerstva ?

Skup aktivnosti čiji je razvoj ili evolucija programskog produkta. (specifikacija – analizom zahtjeva specificirati što sustav treba činiti i koja su ograničenja razvoja, razvoj i oblikovanje – izbor arhitekture i produkcija sustava, validacija i verifikacija – provjera da li sve radi kako treba, evolucija – rukovanje promjenama sukladno novih zahtjevima)

6. Što je model procesa programskog inženjerstva ?

Pojednostavljeno predstavljanje procesa iz određene perspektive.

7. Kakva je struktura cijene (troška) u programskom inženjerstvu ?

Cijena sadrži cijenu razvoja, oblikovanja, ispitivanja i održanja.

8. Što su metode programskog inženjerstva ?

To su strukturni pristupi u razvoju i oblikovanju programske potpore što uključuje izbor modela sustava, pravila, preporuke i naputke.

9. Što je CASE (engl. *Computer-Aided Software Engineering*) ?

To su programski produkti namijenjeni automatiziranoj podršci aktivnostima u procesu programskog inženjerstva.

10. Koje su značajke (atributi) dobre programske potpore ?

Prihvatljivost – razumljiv koristan i kompatibilan o ostalim sustavima, pouzdanost – korisnik mora vjerovati u pouzdanost sustava, održavanje – evolucija je sukladna izmijenjenim zahtjevima.

11. Koje su osnovne poteškoće i izazovi u programskom inženjerstvu ?

Heterogenost – različite tehnike i metode razvoja

Vrijeme isporuke, povjerenje.

12. Koje vrste projekata postoje u programskom inženjerstvu ?

Korektivni, adaptivni, re-inženjerstvo, integrativni, unaprjeđujući.

13. Što je profesionalna i etička odgovornost ?

Povjerljivost, kompetencije, poštivanje prava intelektualnog vlasništva, ne zlorabiti računalne sustave.

Inženjerstvo zahtjeva

- proces pronalaženja, analiziranja, dokumentiranja i provjere zahtijevanih usluga, te ograničenja u uporabi
- Zahtjevi sami za sebe su specifikacija

Zahtjevi obzirom na razinu detalja:

- Korisničke zahtjeve – moraju biti razumljivi ne-tehničkom osoblju, pišu se prirodnim jezikom te grafovima
- Zahtjeve sustava – pišu se strukturnim prirodnim jezikom, jezikom za oblikovanje sustava i matematičkom notacijom
- Specifikacija programske potpore – najdetaljniji opis i objedinjuje korisničke te sustavske zahtjeve

Zahtjevi obzirom na sadržaj:

- Funkcionalni – izjave o uslugama koje sustav mora sadržavati, kako sustav reagira na poticaje, te kako se mora ponašati u određenim situacijama
- Nefunkcionalni – ograničenja u uslugama i funkcijama
- Zahtjevi domene primjene – proizlaze iz domene sustava kao i oni koji karakteriziraju tu domenu

Klasifikacija nefunkcionalnih zahtjeva:

- Zahtjevi programskog produkta – ponašanje na određen način
- Organizacijski zahtjevi
- Vanjski zahtjevi – zahtjevi izvan sustava i razvojnog procesa

Izražavanje zahtjeva sustava:

- Strukturirani prirodan jezik
- Jezik za opis oblikovanja
- Grafička notacija
- Matematička specifikacija

Kompletnost i konzistencija zahtjeva

- Kompletni zahtjevi – sadrže opis svih zahtijevanih mogućnosti
- Konzistentni zahtjevi – nemaju konflikata u opisima

Izražavanje zahtjeva

- Strukturirani prirodni jezik – definiranje standardnih obrazaca i formulara kojima se zahtjevi izražavaju
- Jezik za opis oblikovanja (npr. SDL)
- Grafička notacija – grafički jezik proširen tekstom
- Matematička specifikacija – notacija na osnovu matematičkog koncepta

Zahtjevi sustava:

- Proceduralno sučelje – API
- Struktura podataka
- Predstavljanje podataka

Procesi inženjerstva zahtjeva

- NEMA JEDINSTVENOG PROCESA!
- Generičke aktivnost:
 - Studija izvedivosti
 - Izlučivanje zahtjeva
 - Validacija zahtjeva
 - Upravljanje promjenama zahtjeva

Modeli procesa inženjerstva zahtjeva:

- Klasični
- Spiralni

Spiralni proces:

- Trostupanska aktivnost (specifikacija, validacija, izlučivanje)
- U svakoj iteraciji različit je intenzitet aktivnosti
- Svaka iteracija ima različitu razinu detalja

Generičke aktivnosti u procesu inženjerstva zahtjeva:

- Studija izvedivosti
- Izlučivanje i analiza zahtjeva
- Validacija zahtjeva
- Upravljanje promjenama zahtjeva

Studija izvedivosti:

- U početku određuje da li se predloženi sustav isplati
- Kratka studija koja u pisanom dokumentu provjera pitanja:
 - Da li sustav doprinosi ciljevima organizacije?
 - Da li se sustav može izvesti postojećom tehnologijom?
 - Da li se sustav može integrirati u već postojeći?
- Provedba se temelji na određivanju koje informacije su potrebne za studiju, prikupljanje informacija i pisanje izvješća

Aktivnosti u spirali:

- Izlučivanje zahtjeva
- Klasifikacija i otkrivanje zahtjeva
- Ustanovljavanje prioriteta i pregovaranje
- Dokumentiranje zahtjeva

Pogledi – način strukturiranja zahtjeva da oslikavaju perspektivu i fokus različitih sudionika

Tipovi pogleda:

- Pogledi interakcije – direktni kontakt npr. bankomat
- Indirektni pogledi – dionici ne koriste sustav direktno
- Pogledi domene primjene

Metode u izlučivanju i analizi zahtjeva:

- Intervjuiranje (zatvoreni, otvoreni) – sa pripremljenim skupom pitanja, bez pripremljenog skupa pitanja
- **Scenariji**
 - opis početne situacije, opis normalnog tijeka događaja, opis što se može krivo dogoditi, informacije o paralelnim aktivnostima, opis gdje scenarij završava
 - Primjeri iz stvarnog života o načinu korištenja sustava
- **Obrasci uporabe**
 - temelji se na ideji scenarija
 - Model obrasca je pogleda koji ističe ponašanje sustava kako ga vide vanjski korisnici
 - Model razdjeljuje funkcionalnost sustava u transakcije razumljive korisnicima
 - Tri temeljna elementa u modelima obrazaca:
 - Obrasci uporabe
 - Aktori
 - Odnosi
 - Obrascima modelirati zahtjeve korisnika te scenarije ispitivanja sustava

Zaključci o UML obrascima:

- Modeliraju funkcionalne zahtjeve sustava
- Temeljeni su na ideji scenarija
- Služe za izlučivanje zahtjeva prema pogledu interakcije
- **UML dinamički dijagrami interakcija:**
 - Sekvencijski
 - Kolaboracijski

Validacija zahtjeva:

- Pokazati da zahtjevi odgovaraju sustavu koji naručitelj želi
- **Tehnike validacije:**
 - Recenzija – sistematska ručna analiza
 - Izrada prototipa
 - Generiranje ispitnih slučajeva

Što se u zahtjevima provjerava?

- Valjanost
- Kompletnost
- Adaptabilnost
- Realnost
- Konzistencija
- Provjerljivost
- Razumljivost
- Slijednost

Etnografija:

- Znanost koja istražuje kako ljudi stvarno rade, a ne kako bi definicija poslovnog procesa to propisivala
- Koristi se za izlučivanje zahtjeva uzimajući u obzir aktivnosti drugih sustavom involviranih ljudi

Klasifikacija promjene zahtjeva:

- Okolinom promijenjeni zahtjevi – promjena zbog okoline u kojoj organizacija posluje
- Novonastali zahtjevi – zahtjevi kupca
- Posljedični zahtjevi – nastaju nakon uvođenja u sustav, promjena procesa rada
- Zahtjevi kompatibilnosti – ovise o procesima drugih sustava

Proces programskog inženjerstva:

- Strukturirani skup aktivnosti potreban da se oblikuje i razvija programski produkt

Generičke aktivnosti:

- Specifikacija
- Razvoj i oblikovanje
- Validacija i verifikacija
- Evolucija

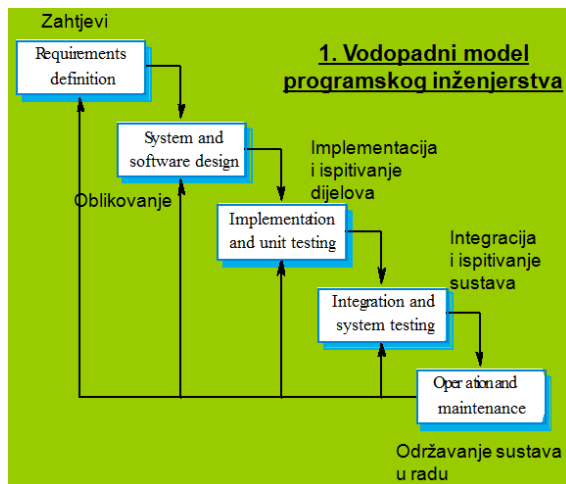
Model procesa programskog inženjerstva:

- Apstraktna reprezentacija procesa, predstavlja opis procesa iz određene perspektive

Najznačajniji procesi i modeli:

1. Vodopadni – odvojene i specifične faze specifikacije i razvoja
2. Evolucijski – isprepleteni specifikacija, evolucija i razvoj
3. Komponentno usmjeren – sustav se gradi od postojećih komponenata
4. RUP proces – temelji se na oblikovanju preko modela

Primjer vodopadnog modela:



Procesne faze vodopadnog modela:

- Analiza zahtjeva i definicije
- Razvoj i oblikovanje sustava i programske potpore
- Implementacija i ispitivanje modula
- Integracija i testiranje sustava
- Uporaba sustava i održavanje

TEMELJ VODOPADNOG SUSTAVA: SVAKA FAZA SE MORA ZAVRŠITI PRIJE POKRETANJA NOVE FAZE!!!!!!!!!!!!!!

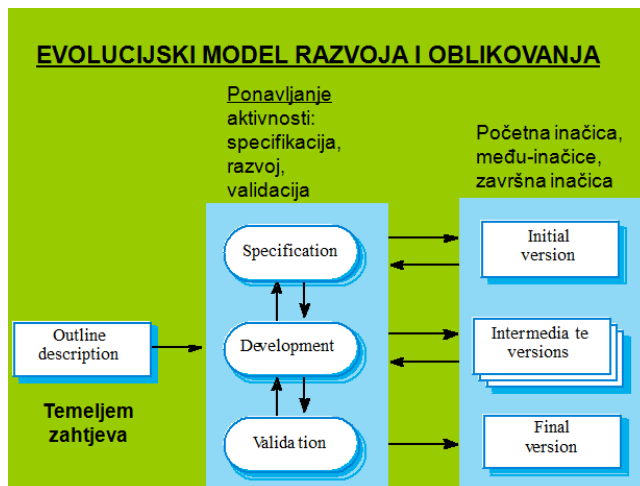
Problemi vodopadnog modela:

- Dijeljenje projekta na particije je teško zbog promjena koje kupac želi
- Model je prikladan ako su zahtjevi razumljivi i promjene svedene na minimum
- Vodopadni model se koristi kod velikih inženjerskih projekata gdje se sustav razvija na nekoliko odvojenih mjesta

Evolucijski model:

- Koristi dva postupka:
 - Istraživački razvoj i oblikovanje:
 - Kontinuiran rad s kupcem
 - Započinje se dobro definiranim početnim zahtjevima te se nove specifikacije dodaju prijedlozima kupca
 - Metoda odbacivanja prototipa:
 - Započinje se grubo definiranim zahtjevima koji se tijekom postupka razjašnjavaju

Primjer evo modela:



Problemi evo modela:

- Proces razvoja nije jasno vidljiv
- Sustavi su loše strukturirani

Primjena evo modela:

- Male i srednje interaktivne sustave
- Dijelove velikih sustava
- Sustavi s kratkim vijekom trajanja

CBSE – model zasnovan na komponentama:

- Sustav se integrira višestrukom upotrebom postojećih komponenata ili uporabom gotovih
- Specifikacija i analiza zahtjeva
- Analiza komponenata
- Modifikacija zahtjeva
- Oblikovanje sustava s višestrukom uporabom
- Razvoj i integracija

RUP – rational unified process

- Moderan model procesa programskog inženjerstva izveden na temelju jezika za modeliranje UML-a u pridruženih aktivnosti
- Dijagrami su modelirani UML standardom
- Modeli su dokumentirani s jednim ili više dijagrama
- Opisuje se kroz tri perspektive:
 - **Dinamička** – pokazuje slijed faza procesa kroz vrijeme
 - **Statička** – pokazuje aktivnosti unutar pojedinih faza procesa
 - **Praktična** – sugerira aktivnosti kroz iskustvo i dobru praksu
- Faze RUP procesa:
 - Početak
 - Elaboracija – plan projekta, specifikacija značajki i arhitekture sustava

- Konstrukcija
- Prijenos produkta korisnicima

Iteracije u modelima procesa programskog inženjerstva:

- Postoje dva međuovisna pristupa:
 - Inkrementalni pristup
 - Spiralni razvoj i oblikovanje
- **Inkrementalni pristup:**
 - Sustav se ne isporučuje u cjelini korisniku nego se dijeli u inkrementalne dijelove koji predstavljaju djelomične funkcionalnosti
 - Zahtjevi se dijele prema prioritetima te tako oni najviši prioriteta se isporučuju u ranim dijelovima sustava
- Prednosti inkrementalnog pristupa:
 - Kupac sa svakim dijelom dobiva svoju vrijednost
 - Smanjen rizika za neuspjeh
 - **Ekstremno programiranje:**
 - Pristup se bazira na oblikovanju i isporuci malih funkcionalnih inkremenata
 - Postupak uključuje kontinuirano poboljšanje koda
 - U razvoju se koristi programiranje u paru

Spiralni razvoj i oblikovanje:

- Proces se predstavlja spiralom
- Svaka petlja u spirali predstavlja jednu fazu procesa
- Nema ranih završenih faza
- **Sektori u spiralnom modelu:**
 - Postavljanje ciljeva
 - Procjena i smanjivanje rizika
 - Razvoj i validacija
 - Planiranje

Generičke aktivnosti u procesu programskog inženjerstva:

- **Faze:**
 - Specifikacija i izlučivanje zahtjeva
 - Oblikovanje i implementacija programskog produkta
 - Validacija i verifikacija
 - Evolucija
- **Specifikacija i izlučivanje zahtjeva:**
 - Aktivnosti:
 - Studij izvedivosti
 - Izlučivanje zahtjeva
 - Validacija zahtjeva
 - Upravljanje promjenama zahtjeva
- **Oblikovanje i implementacija programskog produkta:**
 - Oblikovanje programskog produkta:
 - Implementacija

- **Ispitivanje programske potpore:**
 - Testni podatci (I) – postavljeni ulazi za testiranje
 - Očekivani izlaz (O)
 - Testni slučaj – uređen par (I, O)
 - Stvarni izlaz – rezultat dobiven provođenjem testa
 - Kriterij prolaza testa – kriterij usporedbe očekivanog i stvarnog izlaza određen prije provođenja testa
- **Formalna verifikacija:**
 - Postupak provjere da formalni model dijela izvedenog sustava (I) odgovara formalnoj specifikaciji (S) sa matematičkom izvjesnošću
- **Case – computer aided software engineering**
 - Case podupire automatizaciju oblikovanja raznim alatima
 - Funkcionalna perspektiva
 - Alati se klasificiraju prema specifičnoj funkciji
 - Procesna perspektiva
 - Alati se klasificiraju prema aktivnostima koje podupiru u procesu
 - Integracijska perspektiva
 - Alati se klasificiraju prema njihovoj organizaciji u integralne cjeline
 - Alati – podupiru individualne zadatke u procesu
 - Radne klupe (workbench) – podupiru pojedine faze procesa
 - Razvojne okoline (environments) – podupiru cijeli ili značajan dio procesa programskog inženjerstva