**1. (1 bod) Navedite generičke aktivnosti u svim procesima programskog inženjerstva.**

Rješenje: Specifikacija, oblikovanje i implementacija, validacija i verifikacija, evolucija

**2. (1 bod) Kako se nazivaju programski alati koji podupiru aktivnosti procesa programskog inženjerstva?**

Rješenje: CASE , alati Računalom podržano programsko inženjerstvo …

**3. (1 bod) Navedite barem dvije značajke metodologije ubrzanog razvoja (engl. agile methodology) programske potpore.**

Rješenje: Iterativni razvoj, mali inkrementi, kontinuirano poboljšanje PP, naglasak na ljude i suradnju, uključenost korisnika u proces razvoja…

**4. (1 bod) Navedite na koje ste sve načine u projektnoj dokumentaciji izrazili korisničke zahtjeve.**

Rješenje: UC-ovi s pripadajućim dijagramima (funkcionalni zahtjevi), sekv. dijagrami, lista nefunkcionalnih zahtjeva

**5. (1 bod) Što su zahtjevi domene primjene?**

Rješenje: Funkc. i nefunkc. zahtjevi koji proizlaze iz domene primjene/specifični su za domenu primjene/karakteriziraju domenu primjene

**6. (1 bod) Koji se modifikatori vidljivosti operacija koriste u OO paradigmi kako bi se postigla enkapsulacija? Prikažite kako se označavaju na UML-dijagramima razreda.**

Rješenje: Public +, private - , protected #… mogu navesti i package (ali ne moraju)

**7. (1 bod) Koji minimalni uvjet mora ispuniti neki razred da bi bio apstraktni razred? Može li se instancirati takav razred?**

Rješenje: Razred koji ima barem jednu apstraktnu metodu. Ne može.

**1. (1 bod) Što su zahtjevi domene primjene?**

• Rj. Funkc. i nefunkc. zahtjevi koji proizlaze iz domene primjene/specifični su za domenu primjene/karakteriziraju domenu primjene.

**2. (1 bod) Navedite generičke aktivnosti procesa programskog inženjerstva.**

• Rj. Specifikacija, oblikovanje i implementacija, validacija i verifikacija, evolucija

**3. (1 bod) Navedite barem dvije značajke metodologije ubrzanog razvoja (engl. agile methodology) programske potpore.**

• Rj. Iterativni razvoj, mali inkrementi, kontinuirano poboljšanje PP, naglasak na ljude i

suradnju, uključenost korisnika u proces razvoja…

**4. (1 bod) Koja je temeljna značajka vodopadnog modela (engl. Waterfall model) razvoja programske potpore?**

• Rj. Prethodna faza treba se završiti prije prelaska na novu fazu.

**5. (1 bod) Navedite na koje ste sve načine u projektnoj dokumentaciji izrazili korisničke zahtjeve?**

• Rj. UCovi s pripadajućim dijagramima (funkcionalni zahtjevi), sekv. dijagrami, lista nefunkcionalnih zahtjeva

**1. (1 bod) Navedite na koje se sve načine mogu izraziti zahtjevi sustava.**

• Rj. strukturiranim prirodnim jezikom, specijalnim jezikom za opis oblikovanja (npr. SDL), grafičkom notacijom (npr. UML) i matematičkom specifikacijom (FSM, teorija skupova, logika).

**2. (1 bod) Navedite generičke aktivnosti inženjerstva zahtjeva.**

• Rj. studija izvedivosti, izlučivanje zahtjeva , analiza i specifikacija zahtjeva, validacija zahtjeva, upravljanje zahtjevima

**3. (1 bod) Navedite barem dvije značajke metodologije ubrzanog razvoja (engl. agile methodology) programske potpore.**

• Rj. Iterativni razvoj, mali inkrementi, kontinuirano poboljšanje PP, naglasak na ljude i suradnju, uključenost korisnika u proces razvoja…

**4. (1 bod) Koja je temeljna značajka vodopadnog modela (engl. Waterfall model) razvoja programske potpore?**

• Rj. Prethodna faza treba se završiti prije prelaska na novu fazu.

**5. (1 bod) Navedite na koje ste sve načine u projektnoj dokumentaciji izrazili korisničke zahtjeve?**

• UCovi s pripadajućim dijagramima (funkcionalni zahtjevi), sekv. dijagrami, lista nefunkcionalnih zahtjeva

**1. (1 bod) Navedite značajke dobrog programskog proizvoda.**

Rj. Programski proizvod mora osigurati traženu funkcionalnost i performanse, te mora biti prihvatljiv korisniku, pouzdan i mora se moći održavati.

**2. (1 bod) Navedite metode izlučivanja korisničkih zahtjeva.**

Rj. intervjuiranje, scenarij, obrasci uporabe, dinamičke interakcije korištenjem sekvencijskih dijagrama.

**3. (1 bod) Navedite tri osnovna Scrum artefakta.**

Rj. Projektni dnevnik zaostataka (Product Backlog), Sprint dnevnik (Sprint backlog) i inkrement.

**4. (1 bod) Navedite klasifikaciju arhitekture programske potpore po dosegu.**

Rj. Koncepcijska, logička, izvršna.

**5. (1 bod) Objasnite princip dobrog oblikovanja programske potpore: oblikuj konzervativno (engl. design defensively).**

Rj. Ne koristiti pretpostavke kako će netko upotrebljavati oblikovanu komponentu obraditi sve slučajeve u kojima se komponenta može neprikladno upotrijebiti provjeriti valjanost ulaza u komponentu provjerom definiranih pretpostavki.

**1. (1 bod) Navedite najmanje četiri vrste projekata razvoja programske potpore.**

Korektivni, adaptivni, unapređujući, reinženjerstvo, potpuno novi projekti, integrativni, hibridni

**2. (1 bod) Navedite metode izlučivanja korisničkih zahtjeva.**

Rj. intervjuiranje, scenarij, obrasci uporabe, dinamičke interakcije korištenjem sekvencijskih dijagrama

**3. (1 bod) Koje su tri vrste sudionika u Scrum timu?**

Rj. vlasnik proizvoda (product owner), razvojni tim (development team), scrum vođa (scrum master)

**4. (1 bod) Navedite klasifikaciju arhitekture programske potpore po dosegu.**

Rj. Koncepcijska, logička, izvršna.

**5. (1 bod) Objasnite princip dobrog oblikovanja programske potpore: oblikuj konzervativno (engl. design defensively).**

Rj. Ne koristiti pretpostavke kako će netko upotrebljavati oblikovanu komponentu obraditi sve slučajeve u kojima se komponenta može neprikladno upotrijebiti provjeriti valjanost ulaza u komponentu provjerom definiranih pretpostavki.

**1. (1 bod) Generičke aktivnosti procesa programskog inženjerstva mogu se podijeliti u četiri osnovne skupine. Navedite ih prema vremenskom redoslijedu.**

Rj. Specifikacija, oblikovanje i implementacija, validacija i verifikacija, evolucija.

**2. (1 bod) Navedite tipične načine opisa zahtjeva sustava.**

Rj. strukturiranim prirodnim jezikom, specijalnim jezikom za opis oblikovanja (npr. SDL), grafičkom notacijom (npr. UML) i matematičkom specifikacijom (FSM, teorija skupova, logika).

**3. (1 bod) Što je to projektni dnevnik zaostataka (engl. product backlog) kod pristupa SCRUM radnog okvira? Tko je odgovoran za sadržaj, raspoloživost i sortiranje dnevnika zaostataka?**

Rj. Sortirana lista svih zahtjeva na proizvod i jedino mjesto zahtjeva za bilo kakvim promjenama. Vlasnik proizvoda (engl. product owner) je odgovoran za njegov sadržaj, raspoloživost i sortiranje.

**4. (1 bod) Kako se naziva princip dobrog oblikovanja programske potpore kod kojeg se traži grupiranje međusobno povezanih elemenata, a sve ostale elemente se stavlja izvan grupe?**

Rj. Povećanje kohezije

**5. (1 bod) Koja su 3 osnovna tipa vidljivosti u UML-dijagramu razreda, kako ih označavamo i što ona znače?**

Rj.

Javni (public, +) – svi razredi mogu pristupiti atributu / metodi

Zaštićeni (protected, #) – svi razredi iz hijerarhije podrazreda ili iz samog razreda mogu

pristupiti tom atributu / metodi

Privatni (private, -) – samo je moguće pristupiti iz samog razreda tom atributu / metodi

**6. (1 bod) Što je metoda u objektno orijentiranim programskim jezicima i kako se prikazuje poziv metode na UML-sekvencijskom dijagramu?**

Rj. Metoda je način izvođenja ili implementacija neke operacije. To je procedura, funkcija, rutina,proceduralna apstrakcija koja se koristi za implementaciju ponašanja razreda. Poziv metode prikazuje se strelicom s nazivom poruke (metoda je poziv ili slanje poruka).

**7. (1 bod) Definirajte alociranje odgovornosti (engl. responsibility) razredima.**

Rj. Odgovornost (engl. responsibility) je nešto što sustav mora izvršiti.

Sve odgovornosti jednog razreda moraju biti jasno povezane.

Ako jedan razred ima previše odgovornosti, razmotri podjelu toga razreda u različite razrede.

Ako razred nema odgovornosti, tada je vjerojatno beskoristan.

Ako se neka odgovornost ne može pripisati niti jednom od postojećih razreda, mora se kreirati

novi razred.

**2. (1 bod) Na koja pitanja cjeloviti proces razvoja programske potpore treba definirati odgovore?**

Rj. Proces definira TKO radi ŠTO, KADA i KAKO postići željeni cilj

**6. (1 bod) Koliko UML dijagrama objekata možemo prikazati na temelju jednog UML dijagrama razreda? Ukratko objasnite odgovor.**

Rj. Proizvoljno mnogo, budući da dijagram objekata prikazuje stanje memorije u određenom trenutku izvođenja programa.

**1. (1 bod) Navedite generičke aktivnosti inženjerstva zahtjeva.**

Rj. Studija izvedivosti, izlučivanje zahtjeva , analiza i specifikacija zahtjeva, validacija zahtjeva, upravljanje zahtjevima.

**2. (1 bod) Unificirani proces (engl. Unified process) naglašava kontinuiranu komunikaciju s korisnicima u svim fazama razvoja. Navedite kada korisnici mogu predložiti promjene i dopune?**

Rj. Na kraju svake iteracije.

**3. (1 bod) Scrum radni okvir koristi iterativni, inkrementalni pristup za optimizaciju predvidivosti i kontrole rizika, a sastoji se od Scrum timova i njihovih pridruženih uloga, događaja, artefakata i pravila. Opišite ulogu Scrum vođe (engl. Scrum Master) u timu.**

Rj. Odgovoran za razumijevanje i primjenu teorije, prakse i pravila Scruma.

**5. (1 bod) Navedite na koje ste sve načine u projektnoj dokumentaciji izrazili korisničke zahtjeve.**

Rj. Obrasci uporabe (use cases) s pripadajućim opisima scenarijima, sekv. dijagrami, lista nefunkcionalnih zahtjeva.

**7. (1 bod) Prilikom procesa razvoja dijagrama razreda, objasnite razliku između istraživačkog modela domene primjene, modela domene sustava i modela sustava.**

Rj. Istraživački model domene sustava se crta s najmanje detalja i crta se s ciljem boljeg razumijevanja domene. Model domene sustava ima više detalja i modelira čitavu domenu, ali ne sadrži ostale razrede potrebne u izgradnji cjelovitog sustava. Model sustava uključuje sve razrede u aplikaciji i najdetaljniji je.

**1. (1 bod) Navedite generičke aktivnosti programskog inženjerstva.**

Rj. specifikacije, oblikovanje i implementacija, validacija i verifikacija, evolucija.

**2. (1 bod) U inkrementalnom pristupu razvoja programske potpore sustav se korisniku ne isporučuje u cjelini, a razvoj, oblikovanje i isporuka razbiju se u inkrementalne dijelove koji predstavljaju djelomične funkcionalnosti. Što je presudno u definiranju prioriteta isporuke?**

Rj. Zahtjevi korisnika.

**3. (1 bod) Scrum radni okvir koristi iterativni, inkrementalni pristup za optimizaciju predvidivosti i kontrole rizika, a sastoji se od Scrum timova i njihovih pridruženih uloga, događaja, artefakata i pravila. Opišite značajke razvojnog tima (engl. development team).**

Rj. Samoorganizirajući profesionalci koji isporučuju potencijalno isporučiv inkrement

**4. (1 bod) Koja je temeljna značajka agilnog modela (engl. agile model) razvoja programske potpore?** Rj. razvoj se odvija iterativno i inkrementalno, ponavljaju se aktivnosti oblikovanja, implementacije i ispitivanja.

**5. (1 bod) Navedite kojim ste sve dijagramima u projektnoj dokumentaciji prikazali arhitekturu sustava.**

Rj. dijagrami razreda, komponenti, razmještaja

**1. (1 bod) Navedite klasifikaciju zahtjeva programske potpore prema razini detalja.**

Korisnički zahtjevi, zahtjevi sustava, specifikacija programske potpore.

**2. (1 bod) Navedite UML dijagrame pomoću kojih ste u projektnoj dokumentaciji opisali funkcionalne zahtjeve.**

Obrasci uporabe, sekvencijski dijagrami

**3. (1 bod) U kakvoj su vezi arhitektura i modeli u Unificiranom procesu (engl. Unified process) razvoja programske potpore?**

Modeli su prijenosnici za vizualizaciju, specifikaciju, konstruiranje (oblikovanje, implementacija) i dokumentiranje arhitekture. / UP promiče oblikovanje programske potpore zasnovano na modelima - arhitektura sadrži skup pogleda u modele.

**4. (1 bod) Objasnite odnos pojmova: odgovornost, operacija i metoda (engl. responsibility, operation, method) u kontekstu razreda.**

Svaki razred ima barem jednu odgovornost (nešto što treba izvršiti) koja se ostvaruje skupom operacija. Operacije se implementiraju metodama u programskom jeziku.

**5. (1 bod) Koliko tipično traje jedan sprint u Scrumu?**

2-4 tjedna

**6. (1 bod) Navedite četiri CASE alata koje ste koristili na projektu.**

Alati za komunikaciju: Teams, WhatsApp, Slack...

Alati za verzioniranje: Git, Gitlab ...

Alati za modeliranje: Astah, Visual paradigm ...

Alati za oblikovanje sustava: Eclipse IDE, Visual Studio ...

**7. (1 bod) Na koji način ste koristili princip oblikovanja Povećaj uporabu postojećeg (engl. Increase reusability where possible) u izradi projekta. Navedite primjer.**

Npr. Korištenje gotovih radnih okvira tipa Spring Boot, Bootstrap, Flutter, korištenje knjižnica, korištenje vlastitih razvijenih razreda na više mjesta u kodu, predložaka i sl....

**1. (1 bod) Navedite četiri dionika na Vašem projektu.**

Razvojni tim, asistenti, korisnici aplikacije, administratori sustava ...

**2. (1 bod) Navedite klasifikaciju zahtjeva s obzirom na sadržaj.**

Funkcionalni, nefunkcionalni, domene primjene

**3. (1 bod) Navedite UML dijagrame pomoću kojih ste u projektnoj dokumentaciji opisali arhitekturu programske potpore**.

Dijagram razreda, komponenti.

**4. (1 bod) Kakvu ulogu imaju obrasci uporabe u Unificiranom procesu (engl. Unified process) razvoja programske potpore?**

Obrasci uporabe su pokretači iteracija/ .aktivnosti u životnom ciklusu i sinkroniziraju sadržaj različitih modela.

**5. (1 bod) Usporedite razinu detalja UML dijagrama razreda u prvoj i drugoj inačici projektne dokumentacije.**

U prvoj prevladavaju konceptualni i specifikacijski, a u drugoj implementacijski.

**6. (1 bod) Što je još uz vizualizaciju rada glavna praksa Kanban metodologije?**

Ograničavanje količine rada u tijeku.

**7. (1 bod) Navedite jedan od 5 SOLID principa koji ste najviše primjenjivali na projektu i ukratko obrazložite način na koji ste ga koristili.**

Single responsibility, Open-closed, Liskov, Interface segregation, Dependency inversion