

INFORMACIJSKI SUSTAVI – 2. KPZ

7. DIZAJN ARHITEKTURE

Fizička arhitektura je:

- a) Raspodjela fizičkih komponenti na logičke komponente
- b) Raspodjela logičkih komponenti na fizičke uređaje
- c) Brojnost fizičkih uređaja koji se koriste
- d) Raspodjeka logičkih komponenti na vrste
- e) Raspodjela tipova funkcionalnosti

Što od navedenog je arhitekturni obrazac?

- a) Model-View-View model
- b) Slojeviti obrazac
- c) Debeli klijent
- d) Događajima pokretan obrazac
- e) Cjevovodi i filtri

Sustav je implementiran tako da po bibliotekama ima raspodijeljene sljedeće funkcionalnosti:

1. pristup bazi podataka MSSQL
2. pristup bazi podataka MySQL
3. programski opis domene (poslovni objekti i pravila)
4. validacija domenskih razreda
5. web API aplikacija
6. klijentska web aplikacija

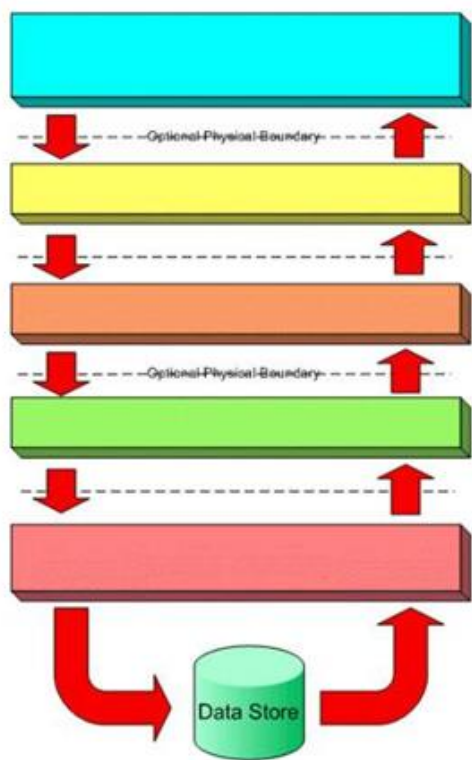
Biblioteke su raspodijeljene i pokrenute po uređajima s jedinstvenim nazivima na sljedeći način

- `MSSQL_provider` - 1)
- `MySQL_provider` - 2)
- `Main_API` - 3), 4), 5)
- `klijentsko-računalo_1` - 6)
- `klijentsko-računalo_2` - 6)

Ne uzimajući u obzir bazu podataka, koliko fizičkih slojeva ima ova arhitektura?

- a) 6
- b) 4
- c) 2
- d) 3
- e) 5

Priložena slika predstavlja petoslojnu logičku arhitekturu.



Slojevi:

- Tirkizno – Korisničko sučelje, prezentacijski sloj
- Žuto – Kontrola sučelja, kontrola radnog toka
- Narančasto – Poslovni sloj
- Zeleno – Sloj za pristup podacima
- Rozo – Sloj za upravljanje pohranom i upravljanje podacima
- Zeleni valjak – Fizička pohrana

MVC Controller može se u petoslojnoj logičkoj arhitekturi matirati kao...

- a) Poslovni sloj i sloj za pristup podacima
- b) Sloj poslovne validacije
- c) Sloj za pristup podacima
- d) Poslovni sloj
- e) Kontrola sučelja

Izbacite uljeza u petoslojnoj logičkoj arhitekturi:

- a) Aplikacijsko sučelje
- b) Poslovni sloj
- c) Sloj za upravljanje pohranom i upravljanje podacima
- d) Sloj za pristup podacima
- e) Korisničko sučelje

Što od navedenog nije vezano za sloj kontrole sučelja?

- a) Interpretacija / obrada ulaza
- b) Vođen događajima (event-driven)
- c) Generiranje izlaza
- d) Prikaz podataka i unos od strane korisnika
- e) Prihvat unosa korisnika i prosljeđivanje u BL na validaciju i obradu

Koji je od sljedećih logičkih slojeva vođen događajima, prihvaća unos korisnika i prosljeđuje podatke na validaciju i obradu?

- a) Kontrola sučelja
- b) Fizička pohrana
- c) Korisničko sučelje
- d) Sloj za pristup podacima
- e) Poslovni sloj

Što se od navedenog ne odvija u podatkovnom sloju?

- a) Inicijacija dohvata, umetanja, ažuriranja i brisanja podataka
- b) Ništa od navedenog
- c) Komunikacija s poslovnim slojem
- d) Upravljanje podacima i njihova fizička pohrana
- e) Objektno – relacijsko mapiranje

Sloj pohrane i rukovanja podacima:

- a) Ne smije sadržavati poslovna pravila
- b) Upravlja fizičkim stvaranjem podataka
- c) Sadrži skup sučelja za komunikaciju s poslovnim slojem
- d) Ništa od navedenog
- e) Sve navedeno

Objektno-relacijsko mapiranje se očekivano nalazi u...

- a) Sloju za pristup podacima
- b) Poslovnom sloju
- c) Poslovnom sloju i sloju za pristup podacima
- d) Korisničkom sučelju
- e) Fizičkoj pohrani

Ako je L = broj slojeva logičke arhitekture, a F = broj slojeva fizičke arhitekture, tada vrijedi:

- a) $L \leq F$
- b) Može vrijediti bilo koji od navedenih uvjeta
- c) $L \geq F$
- d) $L = F$
- e) $L \neq F$

MVVM ViewModel u petoslojnoj logičkoj arhitekturi može se smatrati...

- a) Fizičkoj pohrani
- b) Korisničkom sučelju
- c) Poslovnom sloju i sloju za pristup podacima
- d) Kontrolom sučelja
- e) Poslovnom sloju

Za aplikacije s relativno čvrstom krajnje korisničkom funkcionalnošću korištene u okolini gdje je dobro uspostavljeno upravljanje sustavom najbolji odabir je:

- a) Dvoslojna klijent-poslužiteljska arhitektura i tanki klijent
- b) Ništa od navedenog
- c) Sve navedeno
- d) Dvoslojna klijent-poslužiteljska arhitektura i debeli klijent
- e) Troslojna ili višeslojna poslužiteljska arhitektura

Za aplikacije koje zahtijevaju računalno zahtjevne obrade podataka (npr. vizualizacija podataka) odabir je:

- a) Dvoslojna klijent-poslužiteljska arhitektura i debeli klijent
- b) Ništa od navedenog
- c) Sve navedeno
- d) Dvoslojna klijent-poslužiteljska arhitektura i tanki klijent
- e) Troslojna ili višeslojna poslužiteljska arhitektura

Za sustave u kojima su i podaci i aplikacije promjenjivi najbolji odabir je:

- a) Dvoslojna klijent-poslužiteljska arhitektura i debeli klijent
- b) Ništa od navedenog
- c) Sve navedeno
- d) Dvoslojna klijent-poslužiteljska arhitektura i tanki klijent
- e) Troslojna ili višeslojna poslužiteljska arhitektura

Što od navedenog nije istina?

- a) Povećanje broja fizičkih slojeva može povećati složenost sustava
- b) Pretpostavlja se preslikavanje logičke arhitekture u fizičku u omjeru 1:1
- c) Prilikom kreranja aplikacije važno je odabrati logičku arhitekturu koja će omogućiti kasniji odabir fizičke arhitekture
- d) Povećanje broja fizičkih slojeva može smanjiti složenost sustava
- e) Broj slojeva logičke arhitekture je veći ili jednak broju slojeva fizičke arhitekture

U karakteristike koje se procjenjuju pri uslojavanju fizičke arhitekture spada:

- a) Kopčanje modula
- b) Standard kodiranja
- c) Lakoća održavanja
- d) Ponovna iskoristivost
- e) Skalabilnost

Postavljanjem DAL-a na aplikacijski poslužitelj, a BL na klijenta i poslužitelja:

- a) povećava se skalabilnost
- b) sve navedeno
- c) smanjuju se performanse
- d) povećava se sigurnost
- e) povećava se latencija

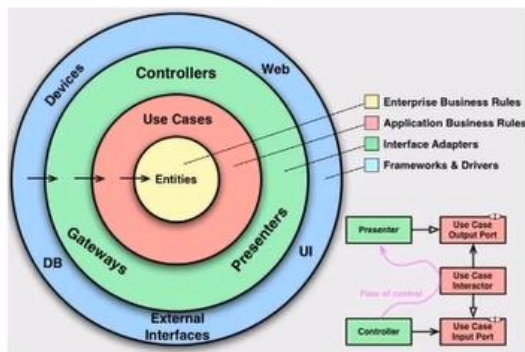
Što je aplikacijski okvir (Application Framework)?

- a) Skup osnovnih softverskih rutina za spremanje objekata te odustajanje od promjena
- b) sve tvrdnje su točke
- c) Skup osnovnih softverskih rutina koje čine temeljnu strukturu za razvoj aplikacije u koju se ugrađuju aplikacijski specifične komponente
- d) Grafičko sučelje s puno dizajnerskih mogućnosti
- e) Sve tvrdnje su točne

Što je okvir poduzeća (Enterprise Framework)?

- a) Skup osnovnih softverskih rutina koje čine temeljnu strukturu za razvoj aplikacije u koju se ugrađuju aplikacijski specifične komponente
- b) Skup osnovnih softverskih rutina za spremanje objekata te odustajanje od promjena
- c) Cjelovito okruženje za razvoj i ugradnju složenih informacijskih sustava
- d) Logika ponašanja aplikacije
- e) Grafičko sučelje s puno dizajnerskih mogućnosti

Sljedećom slikom prikazana je:



- a) Čista arhitektura
- b) Heksagonalna arhitektura
- c) Arhitektura vertikalnog reza
- d) Arhitektura lukovice
- e) CSLA.NET arhitektura

U čistoj arhitekturi (Clean architecture), poslovna pravila poduzeća se implementiraju u prstenu...

- a) Radnim okvirima i pokretačima
- b) Entiteta
- c) Prilagodnicima sučelja
- d) Poslovnih upravljača
- e) Slučajeva korištenja

U arhitekturi vertikalnog reza kopčanje ograničavamo na...

- a) Aplikaciju
- b) Korisničko sučelje
- c) Rez (slice)
- d) Domenu
- e) Kontrolu sučelja



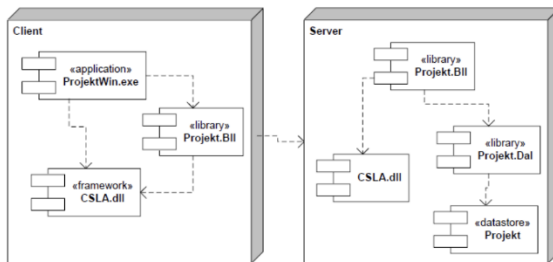
Obrascem priključaka i prilagodnika...

- propisujemo koje se baze podataka mogu koristiti za pohranu podataka
- razdvajamo poruke na upite i naredbe
- definiramo sučelja koja propisuju oblik povezivanja vanjskih komponenti i implementiramo sučelja kao komponente koje se vežu na sustav
- preusmjeravamo događaje u sustavu
- izdvajamo agregate od ostalih elemenata

U šesterokutnoj arhitekturi priključke i prilagodnike dijelimo na:

- Primarne, sekundarne i tercijarne
- Primarne i sekundarne
- Pokretačke i notifikacijske
- Aplikacijske i administrativne
- Aplikacijske i pohranjivačke

Sljedećom slikom prikazan je:



- Dijagram paketa
- Dijagram slučaja korištenja
- Dijagram imenskih prostora
- Dijagram komponenti
- Dijagram ugradnje

Što prikazuje dijagram komponenti (component diagram)?

- Sučelje glavne forme
- Bazne poslovne klase
- Bazne forme i kontrole
- Organizaciju i zavisnost softverskih komponenti
- Pomoćne klase

Što iskazuje zavisnost komponenti u dijagramu komponenti?

- a) Sve tvrdnje su pogrešne
- b) Sve tvrdnje su točne
- c) Realizaciju slučajeva korištenja
- d) Logiku ponašanja aplikacije
- e) Na koji način promjena jedne komponente može utjecati na promjenu drugih komponenti

Što se prikazuje kombiniranjem dijagrama komponenti i dijagrama ugradnje?

- a) Način na koji promjena jedne komponente može utjecati na promjenu drugih komponenti
- b) Softverske komponente
- c) Lokacije komponenti unutar (distribuiranog) sustava
- d) Modeliranje tehničke infrastrukture
- e) Detaljni dizajn konfiguracije sustava

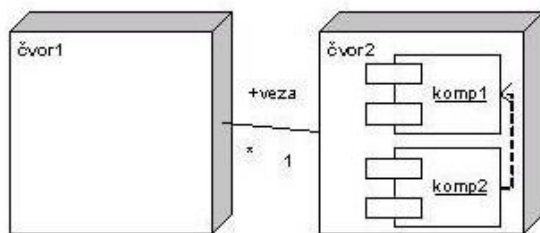
Što se prikazuje kombiniranjem dijagrama komponenti i dijagrama ugradnje?

- a) Način na koji promjena jedne komponente može utjecati na promjenu drugih komponenti
- b) Detaljni dizajn konfiguracije sustava
- c) Softverske komponente
- d) Modeliranje tehničke infrastrukture
- e) Fizička zavisnost između softvera i hardvera

Što od navedenog je svojstvo dijagrama paketa?

- a) Sve navedeno
- b) Pojedini element sadržan je samo u jednom paketu
- c) Paketi tvore imenike (namespace)
- d) Paketi mogu referencirati druge pakete
- e) Nazivi unutar paketa moraju biti jedinstveni

Što prikazuje sljedeća slika:



- a) Kombinaciju dijagrama komponenti i dijagrama ugradnje
- b) Dijagram komponenti
- c) Dijagram paketa
- d) Kombinaciju dijagrama paketa i dijagrama komponenti
- e) Dijagram ugradnje

8. IZRADA SUSTAVA

Što je osnovica plana testiranja za testiranje zahtjeva (Requirements Testing)?

- a) Dizajn sustava, testovi komponenti i integracijski testovi
- b) Prijedlog sustava, dizajn infrastrukture
- c) Sustav pomoći, postupci, priručnici
- d) Dijagrami razreda, dijagrami slijeda, dijagrami komunikacije
- e) Dizajn sučelja

Što je osnovica plana testiranja za testiranje dokumentacije (Documentation Testing)?

- a) Dizajn sustava, testovi komponenti i integracijski testovi
- b) Prijedlog sustava, dizajn infrastrukture
- c) Sustav pomoći, postupci, priručnici
- d) Dijagrami razreda, dijagrami slijeda, dijagrami komunikacije
- e) Dizajn sučelja

Što je kodiranje?

- a) Sve navedeno
- b) Faza u kojoj se obavlja izrada novog sustava i isporuka tog sustava u produkciju, to jest svakodnevnu primjenu
- c) Instalacija i provjera nabavljenih softverskih paketa (po potrebi)
- d) Pretvorba detaljnog opisa programa u stvarni program, najčešće pisanje izvornog koda nekog formalnog programskog jezika
- e) Dokument s eksplicitnim instrukcijama kako napisati kod metode

Kada je riječ o očuvanju kvalitete programskog koda, što se odnosi na kopčanje (coupling) modula?

- a) Moduli trebaju biti minimalno međusobno zavisni
- b) Moduli međusobno moraju biti interno visoko povezani
- c) Sve tvrdnje se odnose na kopčanje (coupling)
- d) Svaki modul treba obavljati jednu i samo jednu funkciju
- e) Postizanje ponovne upotrebljivosti u budućim programima

Što je testiranje jedinica (Unit testing)?

- a) Testiranje svakog procesa korak po korak
- b) Pojedinačno testiranje kod kojeg je najmanja jedinica mjere razred
- c) Testiranje mogućnosti oporavka i neautoriziranog pristupa
- d) Provjera rada sustava kao cjeline, kojom se osigurava da svi nezavisno razvijeni aplikacijski programi rade ispravno te sukladno specifikacijama
- e) Ispitivanje komponenti koje integrirane čine cijeli sustav ili neki njegov dio

Što je od navedenog nedostatak refaktoriranja?

- a) Narušavanje strukture programskog koda
- b) Povećanje razumljivosti
- c) Povećanje produktivnosti pri umanjenoj primjeni
- d) Olakšava otkrivanje bugova
- e) Smanjenje produktivnosti pri pretjeranoj primjeni

Što je od navedenog prednost refaktoriranja?

- a) Sprječava narušavanje strukture programskog koda
- b) Povećanje razumljivosti i čitljivosti programskog koda
- c) Povećanje produktivnosti
- d) Olakšava otkrivanje bugova
- e) Sve navedeno su prednosti refaktoriranja

Što je karakteristično za testiranje slučajeva korištenja (Use-Case testing)?

- a) Provjerava se svaka funkcija sučelja
- b) Provjera kako cjelina radi
- c) Provjerava prikladnost sustava za korištenje
- d) Provjera se što cjelina radi, to jest da li zadovoljava zahtjeve
- e) Provjerava se svaki slučaj korištenja

Što od navedenog pripada testiranju sustava (System Testing)?

- a) Testiranje korisničkog sučelja (User Interface Testing) i Testiranje slučajeva korištenja (Use-Case Testing)
- b) Sve navedeno
- c) Funkcionalno (black-box testing) i strukturalno (white-box, clear box testing)
- d) Testiranje interakcije (Interaction Testing) i Testiranje sučelja sustava (System Interface Testing)
- e) Testiranje zahtjeva (Requirements Testing) i Testiranje uporabivosti (Usability Testing)

Integracijsko testiranje može biti:

- a) Testiranje korisničkog sučelja (User Interface Testing) i Testiranje slučajeva korištenja (Use-Case Testing)
- b) Testiranje sigurnosti (Security Testing) i Testiranje performansi (Performance Testing)
- c) Funkcionalno (black-box testing) i strukturalno (white-box, clear box testing)
- d) Testiranje interakcije (Interaction Testing) i Testiranje sučelja sustava (System Interface Testing)
- e) Testiranje zahtjeva (Requirements Testing) i Testiranje uporabivosti (Usability Testing)

Što je testiranje sustava (System Testing)?

- a) Testiranje svakog procesa korak po korak
- b) Pojedinačno testiranje kod kojeg je najmanja jedinica mjere razred
- c) Testiranje mogućnosti oporavka i neautoriziranog pristupa
- d) Provjera rada sustava kao cjeline, kojom se osigurava da svi nezavisno razvijeni aplikacijski programi rade ispravno te sukladno specifikacijama
- e) Ispitivanje komponenti koje integrirane čine cijeli sustav ili neki njegov dio

Što je karakteristično za testiranje performansi (Performance Testing)?

- a) Provjerava se svaki proces korak po korak
- b) Provjerava se svaka funkcija sučelja
- c) Provjerava mogućnost izvođenja pod velikim opterećenjem
- d) Provjerava prikladnost sustava za korištenje
- e) Provjera se što cjelina radi, to jest da li zadovoljava zahtjeve

Što je karakteristično za testiranje dokumentacije (Documentation Testing)?

- a) Provjerava se svaki proces korak po porak
- b) Provjerava ispravnost dokumentacije
- c) Provjerava mogućnost izvođenja pod velikim opterećenjem
- d) Provjerava prikladnost sustava za korištenje
- e) Provjera se što cjelina radi, to jest da li zadovoljava zahtjeve

Što je karakteristično za testiranje zahtjeva (Requirements Testing)?

- a) Provjerava se svaki proces korak po porak
- b) Provjerava se razmjena podataka s drugim sustavima
- c) Provjerava jesu li zadovoljeni izvorni poslovni zahtjevi
- d) Provjerava prikladnost sustava za korištenje
- e) Provjera se što cjelina radi, to jest da li zadovoljava zahtjeve

Što je karakteristično za testiranje uporabivosti (Usability Testing)?

- a) Provjerava se svaki proces korak po porak
- b) Provjerava se svaka funkcija sučelja
- c) Provjerava mogućnost izvođenja pod velikim opterećenjem
- d) Provjerava prikladnost sustava za korištenje
- e) Provjera se što cjelina radi, to jest da li zadovoljava zahtjeve

Kako se nazivaju pozvane funkcije kod inkrementalnog programiranja koje se kodiraju prilikom izrade funkcije koja poziva neke druge funkcije, tako da je tijelo funkcije sadrži poruku ("Neimplementirana funkcija X") ili hardkodiranu povratnu vrijednost?

- a) Sve je pogrešno
- b) Refaktorirane funkcije
- c) Sve je točno
- d) Odresci ili okrajci (stub)
- e) Pogonitelji ili pokretači (driver)

Što je refaktoriranje?

- a) Pretvorba detaljnog opisa programa u stvarni program, najčešće pisanje zivornog koda nekog formalnog programskog jezika
- b) Sve točno
- c) Sve pogrešno
- d) Promjena interne strukture programske podrške da bi ju se bolje razumjelo i lakše održavalo, uz očuvanje vanjskog ponašanja
- e) Faza u kojoj se obavlja izrada novog sustava i isporuka tog sustava u produkciju, to jest svakodnevnu primjenu

Testiranje jedinica (unit testing) može biti:

- a) Testiranje korisničkog sučelja (User Interface Testing) i Testiranje slučajeva korištenja (Use-Case Testing)
- b) Sve navedeno
- c) Funkcionalno (black-box testing) i strukturalno (white-box, clear box testing)
- d) Testiranje interakcije (Interaction Testing) i Testiranje sučelja sustava (System Interface Testing)
- e) Testiranje zahtjeva (Requirements Testing) i Testiranje uporabivosti (Usability Testing)

***Replace conditionals with polymorphism* je primjer tehnike refaktoriranja kod kojeg se:**

- a) Dio koda izdvaja u zasebnu metodu
- b) Zamjenjuje varijabla koja poprima vrijednost nekog izraza s pozivom metode
- c) Zamjenjuje uvjet (switch) koji ispituje tip objekta s polimorfizmom, tako što se originalna metoda učini apstraktnom i u podrazredima se primijeni overriding
- d) Dodaje novo, razumljivo ime metodi
- e) Razred koji "radi puno toga" dijeli u više razreda

Kada je moguće primijeniti refaktoriranje?

- a) Pri pojavi dupliciranog koda, dugih metoda, velikih razreda.. – "bad code smells"
- b) Kada treba ispraviti neki bug
- c) Kada se dodaje nova funkcionalnost
- d) Sve točno
- e) Kada se obavlja pregled koda

Što spada u aktivnosti izrade (programiranje, testiranje, dokumentiranje)?

- a) Instalacija i provjera nabavljenih softverskih paketa (po potrebi)
- b) Ugradnja i provjera baze podataka
- c) Izgradnja i provjera mreža (po potrebi)
- d) Sve navedeno
- e) Pisanje nove programske podrške i provjera novih programa

Što je osnovica plana testiranja za testiranje interakcije (Interaction Testing)?

- a) Dizajn sustava, testovi komponenti i integracijski testovi
- b) Prijedlog sustava, dizajn infrastrukture
- c) Sustav pomoći, postupci, priručnici
- d) Dijagrami razreda, dijagrami slijeda, dijagrami komunikacije
- e) Dizajn sučelja

Što je karakteristično za funkcionalno (black-box testing) testiranje?

- a) Provjerava se svaki proces korak po porak
- b) Provjerava se razmjena podataka s drugim sustavima
- c) Provjerava jesu li zadovoljeni izvorni poslovni zahtjevi
- d) Provjerava prikladnost sustava za korištenje
- e) Provjerava se što cjelina radi, to jest da li zadovoljava zahtjeve

Što je zastoj u radu?

- a) Neispravni dio koda
- b) Dokazivanje da faza dobro provedena ili da je proizvod dobro napravljen, tj. da odgovara specifikaciji zahtjeva (slučajevima korištenja)
- c) Stanje izazvano jednim ili više kvarova
- d) Potvrda valjanosti kojom se utvrđuje da je napravljen pravi proizvod, koji odgovara namjeni te je prihvatljiv korisniku
- e) Provjera je li neki aspekt softvera ispravan

Što je od sljedećeg točno za monolitni pristup programiranju?

- a) Sve pogrešno
- b) Odgađa otkrivanje problema (pogrešava u kodu i dizajnu)
- c) Prosljeđuje probleme u primjenu i održavanje
- d) Sve točno
- e) Dugotrajno kodiranje, a zatim niz ponavljanja oblika provjera+ispravak

Kada je riječ o očuvanju kvalitete programskog koda, što se odnosi na kopčanje (coupling) modula?

- a) Postizanje ponovne upotrebljivosti u budućim programima
- b) Moduli međusobno moraju biti interno visoko povezani
- c) Svaki modul treba obavljati jednu i samo jednu funkciju
- d) Moduli trebaju biti minimalno međusobno zavisni
- e) Sve tvrdnje se odnose na kopčanje (coupling)

Što je plan testiranja?

- a) Instalacija i provjera nabavljenih softverskih paketa (po potrebi)
- b) Definira niz testova, treba ga napraviti na početku razvoja i stalno ažurirati
- c) Zasebno testiranje pojedinačnih metoda
- d) Testiranje poslovnih procesa
- e) Neispravan dio koda

9. OBLIKOVANJE I AUTOMATIZACIJA POSLOVNIH PROCESA

BPMN je skraćenica za

Score: 0.500 (=100.0%)

Id: 9775

Što od navedenog **nije** dio referentnog modela sustava za upravljanje protokom poslova

- ☐ a workflow aplikacije
- ☐ b alat za definiranje procesa
- ☒ c Sve navedeno je dio referentnog modela
- ☐ d alati za administraciju i nadgledanje
- ☐ e jezgra sustava za upravljanje protokom poslova

- ☐ a Business Process Modeling Notebook
- ☒ b Business Process Model and Notation
- ☐ c Business Process Model Notifications
- ☐ d Better Process Modeling Notation
- ☐ e Business Process Modular Notation

Score: 0.000 (=0.0%)

Id: 44302

Označite poželjne (ali ne i nužne) elemente eko-sustava za upravljanje protokom poslova.

Napomena: Pitanje ima više točnih odgovora.

Protok poslova (engl. workflow) bavi se

- ☒ a vizualno modeliranje i prikaz stanja procesa
- ☐ b heterogenost klijenata
- ☒ c nadzor obavljenih i aktivnih procesa
- ☒ d mogućnost prikaza aktivnih zadataka
- ☐ e postojanost (engl. persistence)

- ☐ a utvrđivanjem poslovnog procesa
- ☐ b analizom poslovnog procesa
- ☐ c preoblikovanjem poslovnog procesa
- ☒ d automatizacijom poslovnog procesa
- ☐ e testiranjem poslovnog procesa

Označite načine izvršavanja (engl. hit policy) po DMN-u.

- ☐ a When-And...-Then
- ☒ b Collect
- ☒ c Unique
- ☐ d When-Then
- ☐ e Hit

Što od navedenog **ne vrijedi** za sustav za upravljanje protokom poslova

- ☐ a Omogućava lakše odvajanje "kad i kojim redom nešto napraviti" od "što i kako napraviti"
- ☒ b preoblikuje poslovni proces
- ☐ c kreira i upravlja aktivnostima u procesu
- ☐ d osigurava da se određeni zadatak obavi u pravo vrijeme od prave osobe
- ☐ e interpretira formalnu definiciju procesa

Serijski zadaci koji treba obaviti kako bi se ostvario željeni rezultat je

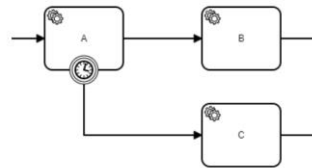
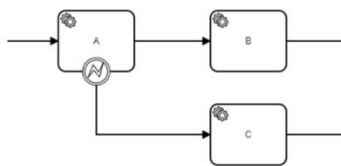
- ☐ a proces
- ☒ b procedura
- ☐ c ishod
- ☐ d aktivnost
- ☐ e automatizam

Označite vrste zadataka u BPMN modelu.

- ☒ a Service task
- ☐ b Terminate Task
- ☒ c Business Rule Task
- ☒ d Receive Task
- ☐ e Exception Task

Označite sve točne odgovore vezane uz početak i kraj procesa po BPMN standardu

- ☒ a Dolazak u bilo koji završni čvor uzrokuje završetak svih aktivnosti u procesu.
- ☒ b BPMN model može imati više završetaka različitih tipova.
- ☒ c Ovisno o vrsti završnog elementa dolazak u završni čvor može (ali ne mora) uzrokovati završetak svih aktivnosti u procesu.
- ☒ d BPMN model može imati više početaka.
- ☒ e Početak BPMN modela može biti uvjetovan različitim okidačima, ali model mora sadržavati samo jedan početak.

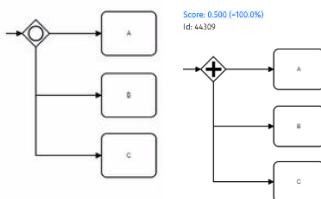


Označite **sve** točne odgovore.

- ☒ a Početak izvršavanja A šalje se signal zadatku C koji se tako (ovisno o trajanju A) može izvršiti prije ili poslije B.
- ☒ b C će se izvršiti samo ako se prilikom izvršavanja A dogodi pogreška.
- ☐ c B i C će se izvršiti samo ako se u A dogodi pogreška.
- ☒ d Ili će se izvršiti B ili C, ali ne oba.
- ☐ e B i C će se izvršiti samo ako se u A ne dogodi pogreška.

Označite **sve** točne odgovore.

- ☒ a C će se izvršiti samo ako A ne bude dovršen u određenom vremenskom periodu
- ☒ b Nakon početka izvršavanja zadatka A, ovisno o vremenu postavljenom u okidaču šalje B.
- ☐ c B i C će se izvršiti samo ako A ne bude dovršen prije isteka vremena propisanog okidaču B.
- ☐ d B i C će se izvršiti samo ako A bude dovršen prije isteka vremena propisanog okidaču B.
- ☒ e Ili će se izvršiti B ili C, ali ne oba.



Grananje kao na slici se naziva

- ☐ a Inclusive Gateway
- ☐ b Exclusive Gateway
- ☒ c Parallel Gateway
- ☐ d Event Gateway
- ☐ e Inclusive Gateway

Grananje kao na slici se naziva

- ☐ a Exclusive Gateway
- ☐ b Sequential Gateway
- ☐ c Inclusive Gateway
- ☒ d Parallel Gateway
- ☐ e Event Gateway

10. METODOLOGIJE RAZVOJA

Jedna od glavnih tehnika metodologije SSADM je

- a** oblikovanje matrice događaja - Event Matrix (EM)
- b** oblikovanje poslovnog procesa - Business Process Design (BPD)
- c** ništa od navedenog
- d** oblikovanje događaja entiteta - Entity Life Histories (ELH)
- e** oblikovanje objektnog modela - Object Model Design (OMD)

Što od navedenog **nije** osnovna praksa ekstremnog programiranja:

- a** informativno radno okružje
- b** programiranje u paru
- c** testovi se pišu nakon programiranja (kodiranja) svake iteracije
- d** rezerva (eng. Slack)
- e** tjedni ciklus

Što najbolje opisuje okolinu (Environment)?

- a** Aktivnosti potrebne da sustav bude dostupan svim krajnjim korisnicima
- b** Disciplina koja opisuje kako kontrolirati i sinkronizirati evoluciju skupa komponenti i isporuka koje zajedno čine konačni sustav
- c** Organizira dijelove metodologije koji pružaju okruženje razvojnom timu, uključujući procese i alate
- d** Uspostavlja poslovni kontekst sustava te oblik organizacije u kojoj sustav treba uvesti u primjenu
- e** Disciplina usredotočena na planiranje projekta, upravljanje rizicima, praćenje napretka i metriku

Što od navedenog **nije** karakteristika Kanban metode?

- a** višljasta metoda za upravljanje i poboljšanje ljudskog rada
- b** praćenje efektivnosti radnika
- c** poboljšanje rješavanja uskih grla na razini sustava
- d** balansiranje potreba s kapacitetom
- e** univerzalno upotrebljiva

Na što se odnosi uvođenje u primjenu (deployment) kod RUP-a?

- a** Definira kako razviti, organizirati, testirati i integrirati komponente
- b** Definira kako testirati i procijeniti kvalitetu rješenja
- c** Uspostavlja poslovni kontekst sustava te oblik organizacije u kojoj sustav treba uvesti u primjenu
- d** Aktivnosti potrebne da sustav bude dostupan svim krajnjim korisnicima
- e** Definira kako saznati i prikupiti želje zainteresiranih strana te ih pretvoriti u skup zahtjeva koji definiraju doseg sustava i potrebnu funkcionalnost

Koje komponente od nabrojanih definiraju RUP (IBM Rational Unified Process - RUP):

(više točnih odgovora)

- a** skup filozofija, ključnih praksi (core practices, best practices) te načela (process essentials) za uspješan razvoj programske podrške
- b** modeli procesa i podataka
- c** jezik za definiranje procesa
- d** metodološki okvir izgrađen od višekratno iskoristivih metoda i procesnih blokova
- e** planiranje, analiza, dizajn i konstrukcija

Općenito gledano, što je nedostatak metodologija?

- ☐ a Nepotpuna podržanost alatima
- ☐ b Nedostatak standarda (velik broj varijanti postupaka i notacija)
- ☒ c Sve navedeno
- ☐ d Neodmjerena - prekomplikirane ili prejednostavne
- ☐ e Rijetke podupiru sve faze životnog ciklusa

Id: 10016

Osoba koja se u Scrum procesu brine o procesu i koordinira zove se

- ☐ a Scrum Owner
- ☐ b Product Owner
- ☐ c Lead Developer
- ☐ d Scrum Guru
- ☒ e Scrum Master

Id: 8083

Najpoznatija i najčešće korištena strukturirana metodologija je:

- ☒ a Structured systems analysis and design method
- ☐ b Ništa navedeno
- ☐ c Unified structured software development process
- ☐ d Unified software development process
- ☒ e Information structured engineering methodology

Radna lista proizvoda u Scrum procesu naziva se

- ☐ a Requirements Log
- ☒ b Product Backlog
- ☐ c Product Specification
- ☐ d Work Items
- ☐ e Back Office

Id: 46369

Što od navedenog nije vrsta CASE alata?

- ☐ a Sve navedeno
- ☒ b Srednji
- ☐ c Integrirani
- ☐ d Gornji
- ☐ e Donji

Što je karakteristično za čvrste ili teške (hard) metodologije?

- ☐ a Sve navedeno
- ☐ b Ništa od navedenog
- ☒ c Opsežnost, složenost i velika količina dokumentacije
- ☐ d Bave se slabo definiranim, neizraženim (fuzzy edged) stvarnim situacijama
- ☐ e Naglasak na sociološkim i ljudskim aspektima

Što je karakteristično za dokumentaciju kod tradicionalnog metodološkog pristupa? Jedna od temeljnih vrijednosti ekstremnog programiranja je

a Dobro dokumentirane	a Kvaliteta
b Malo ili bez dokumentacije	b Ljudskost
c Ništa od navedenog	c Mali koraci
d Uvijek se piše točno jedan dokument	d Ekonomičnost
e Sve navedeno	e Jednostavnost

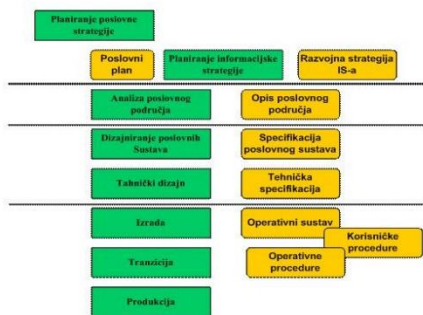
Score: -0.100 (= -20.0%)

Id: 8084

Što je karakteristično za planiranje strategije IS-a (ISP – Information Strategy Planning) unutar metodologije informacijskog inženjerstva (IEM - Information engineering methodology)?

a Izdvajanje aplikacija i definiranje prioriteta temeljem analize poslovnih područja
b Sve navedeno
c Promatranje poslovanja kao cjeline s ciljem definiranja općeg, sveobuhvatnog plana i arhitekture za slijedni razvoj informacijskih (pod)sustava
d Aplikacije postaju projekti u kojima se primjenjuju drugi postupci analize i dizajn
e Proučavanje poslovnih područja i definiranje poslovnih zahtjeva za organizirani i integrirani skup informacijskih (pod)sustava i aplikacija poslovnog područja

Na koju metodologiju se odnosi sljedeća slika?



a ISEM (Information structured engineering methodology)
b USSDP (Unified structured software development process)
c SSADM (Structured systems analysis and design method)
d USDP (Unified software development process)
e IEM (Information engineering methodology)

Što je karakteristično za dokumentaciju kod modernog agilnog metodološkog pristupa?

a Sve navedeno
b Malo ili bez dokumentacije
c Ništa od navedenog
d Uvijek se piše točno jedan dokument
e Dobro dokumentirane

Score: 0.000 (=0.0%)

Id: 10012

U slučaju da je na stabilan proizvod treba dodati inkrement dobro definirane funkcionalnosti, a tim posjeduje iskustvo kako u domeni tako i s postojećim proizvodom, koja od strategija iterativne provedbe RUP projekta će se koristiti?

- ☐ a strategija usitnjavanja
- ☒ b strategija "velikog oblikovanja"
- ☐ c inkrementalna
- ☐ d strategija velikog praska
- ☐ e evolucijska

CASE je

- ☐ a Programski alat za generiranje dijagrama
- ☒ b Računalom podržano programsko/informacijsko inženjerstvo
- ☐ c Ništa od navedenog
- ☐ d Metodologija upravljanja slučajevima korištenja
- ☐ e Metoda/tehnika upravljanja slučajevima korištenja

Jedna od temeljnih vrijednosti ekstremnog programiranja je

- ☐ a Prihvaćena odgovornost
- ☐ b Promišljanje
- ☐ c Raznolikost
- ☐ d Unaprjeđenje
- ☒ e Povratna informacija

Što od navedenog nije točno za planiranje (faza ekstremnog programiranja)

- ☐ a Rok za izdavanje prvog malog izdanja obično je unutar dva mjeseca.
- ☐ b Zatim se određuje cjelokupni vremenski raspored.
- ☒ c Postavlja prioritete na korisničke priče (tj. svojstva programskog rješenja)
- ☒ d Faza planiranja traje nekoliko tjedana do nekoliko mjeseci
- ☐ e Planira se doseg prvog malog izdanja i vrijeme za pojedinu karticu

U slučaju da je domena problema poznata i da su vrlo dobro poznati rizici, a projektni tim je iskusan, koja od strategija iterativne provedbe RUP projekta će se koristiti

- ☐ a strategija velikog praska
- ☐ b evolucijska
- ☐ c strategija usitnjavanja
- ☐ d strategija "velikog oblikovanja"
- ☒ e inkrementalna

Prekretnica zacrtavanja arhitekture (eng. lifecycle architecture milestone) je prekretnica koje faze RUP-a?

- ☒ a faze elaboracije
- ☐ b faze konstrukcije
- ☐ c nije prekretnica samo jedne faze, već može biti prekretnica bilo koje iteracije u bilo kojoj fazi
- ☐ d faze tranzicije
- ☐ e faze inepcije

Mjera programske potpore je:

- ☐ a Code coverage
- ☐ b Source lines of code
- ☒ c Function point analysis
- ☐ d Cyclomatic complexity
- ☒ e Sve od navedenog

Prekretnica s ciljevima (eng. lifecycle objectives milestone) je prekretnica koje faze RUP-a?

- ☐ a faze konstrukcije
- ☐ b nije prekretnica samo jedne faze, već može biti prekretnica bilo koje iteracije u bilo kojoj fazi
- ☐ c faze tranzicije
- ☒ d faze elaboracije
- ☒ e faze inepcije

Metodologija može biti:

- ☐ a Sve navedeno
- ☐ b Perspektivna (prospect)
- ☐ c Crna (Black)
- ☒ d Čvrsta (hard)
- ☐ e Bijela (white)

Score: -0.100 (=20.0%)

Id: 10012

U slučaju da je na stabilan proizvod treba dodati inkrement dobro definirane funkcionalnosti, a tim posjeduje iskustvo kako u domeni tako i s postojećim proizvodom, koja od strategija iterativne provedbe RUP projekta će se koristiti?

- ☐ a strategija usitnjavanja
- ☐ b strategija velikog praska
- ☒ c strategija "velikog oblikovanja"
- ☐ d evolucijska
- ☒ e inkrementalna

Što je karakteristično za fazu inepcije (počinjanja) kod RUP-a?

- ☒ a prikupljanje najvažnijih zahtjeva
- ☐ b prikupljanje detaljnih zahtjeva
- ☐ c testiranje programskog rješenja na lokaciji isporuke
- ☐ d Sve navedeno
- ☐ e upravljanje resursima

Što je karakteristično za fazu konstrukcije, kod RUP-a?

- ☐ a Izvršavanje planova uvođenja u primjenu
- ☐ b Sve navedeno
- ☐ c Opis problemskog konteksta te najvažnijih zahtjeva i ograničenja
- ☐ d Definiranje, validacija i zacrtavanje arhitekture
- ☒ e Upravljanje resursima, kontrola projekta i optimizacija procesa

Što nije jedna od glavnih karakteristika RUP-a?

- ☐ a Produktivnost ekipe
- ☐ b Vizualno modeliranje
- ☐ c Elastičnost i prilagodljivost
- ☒ d Sve su glavne karakteristike RUP-a
- ☐ e Oslonac na UML

Višljasto upravljanje projektima u programskom inženjerstvu ~~ne~~ uključuje:

- ☐ a Redukciju programskog kôda
- ☐ b Stalnu integraciju
- ☐ c Kratke iteracije
- ☒ d Korištenje isključivo jediničnih testova
- ☐ e Sudjelovanje korisnika

Višljasto upravljanje projektima sažeto je u tri koncepta:
(više točnih odgovora)

- ☐ a Raslojavanje aktivnosti
- ☒ b Identificiranje nepotrebnih aktivnosti
- ☐ c Sprječavanje podkapacitiranja resursa
- ☒ d Smanjivanje nepotrebne varijacije unutar toka
- ☒ e Sprječavanje preopterećenja ljudi ili resursa

Kakve su prakse kod tradicionalnog metodološkog pristupa?

- ☒ a Nisu definirane
- ☐ b Ima ih samo pet
- ☐ c Izvrsne
- ☐ d Temelje se na samoorganizirajućim ekipama
- ☐ e Srednje

Komponente metodologije mogu biti:

- ☐ a Definirane faze – varijanta životnog ciklusa
- ☒ b Sve navedeno
- ☐ c Procedure, zadatci, pravila, tehnike i upute za pojedine faze
- ☐ d Alati – preporuke (vodič) uporabe tehnika i pomagala
- ☐ e Dokumentacija – formati, predlošci i organizacija dokumenata

Što nije jedna od glavnih karakteristika RUP-a?

- ☐ a Elastičnost i prilagodljivost
- ☐ b Oslonac na UML
- ☐ c Vizualno modeliranje
- ☒ d Sve su glavne karakteristike RUP-a
- ☐ e Produktivnost ekipe

Što je karakteristično za proces razvoja kod modernog agilnog metodološkog pristupa?

- ☐ a Ništa od navedenog
- ☐ b "Srednji": plan se definira na sredini projekta i više se ne dorađuje
- ☒ c "Lagan": nema strogo definiranog plana, planiranje se odvija kroz cikluse, proces: nelinearan i prilagodljiv, inkrementalan
- ☐ d "Težak": plan strogo definiran na početku – planski usmjeren (eng. plan driven); linearan i predvidiv
- ☐ e Sve navedeno

Područje obrnutog inženjerstva **nije**:

- ☐ a Metrika (software metric, system metric)
- ☐ b Redokumentacija (redocumentation)
- ☐ c Restrukturiranje (restructuring)
- ☒ d Deserijalizacija (deserialization)
- ☐ e Objektifikacija (objectification)

Koji od navedenih pojmova se ne povezuje s razvojem ekipe

- ☐ a Formiranje (forming)
- ☐ b Normiranje (norming)
- ☐ c Predstavljanje, djelovanje (performing)
- ☐ d Jurišanje (storming)
- ☒ e Planiranje (planning)

Koja od sljedećih tvrdnji **nije** točna?

- ☐ a Inženjerstvo unatrag nalazi se u kontekstu kada je izvorni kôd nedostupan i cilj je doći do njega.
- ☐ b Glavna aktivnost inženjerstva unatrag je prepoznavanje dijelova sustava i njihove povezanosti, s ciljem da se identificiraju komponente sustava i dobije potpuna specifikacija.
- ☒ c Inženjerstvo unatrag podrazumijeva izmjene u postojećem sustavu.
- ☐ d Inženjerstvo prema naprijed opisuje tradicionalni postupak - s viših razina apstrakcije prelazimo na projektiranje i izvođenje.
- ☐ e Inženjerstvo unatrag nalazi se u kontekstu kada je izvorni kôd dostupan, ali je dokumentacija nepotpuna, neažurna ili je nema.

Metodologija informacijskog inženjerstva prema Jamesu Martinu je

- ☐ a usmjerena na upravljanje procesom
- ☒ b podatkovno usmjerena
- ☐ c aplikacijski usmjerena
- ☐ d agilno usmjerena
- ☐ e procesno usmjerena

Što je karakteristično za proces razvoja kod tradicionalnog metodološkog pristupa?

- ☐ a "Lagan": nema strogo definiranog plana, planiranje se odvija kroz cikluse, proces: nelinearan i prilagodljiv, inkrementalan
- ☐ b Ništa od navedenog
- ☐ c Sve navedeno
- ☐ d "Srednji": plan se definira na sredini projekta i više se ne dorađuje
- ☒ e "Težak": plan strogo definiran na početku – planski usmjeren (eng. plan driven); linearan i predvidiv

11. SUSTAVI ZA UPRAVLJANJE PODUZEĆEM

Što je upravljanje konfiguracijom i promjenama (Configuration & Change Management)?

- ☐ a Disciplina koja opisuje kako kontrolirati i sinkronizirati evoluciju skupa komponenti i isporuka koje zajedno čine konačni sustav
- ☐ b Uspostavlja poslovni kontekst sustava te oblik organizacije u kojoj sustav treba uvesti u primjenu
- ☐ c Disciplina usredotočena na planiranje projekta, upravljanje rizicima, praćenje napretka i metriku
- ☐ d Organizira dijelove metodologije koji pružaju okruženje razvojnom timu, uključujući procese i alate
- ☐ e Aktivnosti potrebne da sustav bude dostupan svim krajnjim korisnicima

Tko od navedenih nije član ekipe u elastičnom modelu ekipe?

- ☐ a Savjetnik
- ☐ b Administrator baze podataka
- ☐ c Voditelj ekipe
- ☐ d Projektant (analitičar-programer)
- ☐ e Programer

Tko od navedenih nije dio XP ekipe? Tko od navedenih nije dio XP ekipe?

U modernoj organizaciji ekipe suradnju s korisnikom obavlja:

- | | | |
|---|---|---|
| <input type="radio"/> a Administrativni koordinator | <input type="radio"/> a Trener | <input type="radio"/> a Tester |
| <input type="radio"/> b Programer-analitičar | <input type="radio"/> b Klijent, korisnik | <input type="radio"/> b Razvojniki |
| <input type="radio"/> c Sistem inženjer za komunikacije | <input type="radio"/> c Upravitelj | <input type="radio"/> c Klijent |
| <input type="radio"/> d Konzultant | <input type="radio"/> d Tester | <input type="radio"/> d Sistem analitičar |
| <input type="radio"/> e Poslovni analitičar | <input type="radio"/> e Poslovni analitičar | <input type="radio"/> e Upravitelj |

Poželjan broj osoba u dobroj ekipi je?

- | | |
|------------------------------------|---|
| <input type="radio"/> a Više od 10 | Tko od navedenih nije član ekipe u elastičnom modelu ekipe? |
| <input type="radio"/> b 8-10 | <input type="radio"/> a Upravitelj ekipe |
| <input type="radio"/> c 7 ± 2 | <input type="radio"/> b Projektant (analitičar-programer) |
| <input type="radio"/> d Manje od 5 | <input type="radio"/> c Administrator baze podataka |
| <input type="radio"/> e 5 ± 2 | <input type="radio"/> d Sistem inženjer |
| | <input type="radio"/> e Tester |

Score: 0.000 (=0.0%)

Id: 8142

Tko od navedenih nije član ekipe u elastičnom modelu ekipe?

- ☐ a Savjetnik
- ☐ b Administrator baze podataka
- ☐ c Projektant (analitičar-programer)
- ☐ d Voditelj ekipe
- ☐ e Programer