

Riesenie otazok na skusku z PSI 2010/2011

rw link: <http://is.gd/psi11>

stare, ale vyzeraju dobre materialy: bit.ly/jv12SV

pravidla:

poznámky do poznámok -> pre lenivých CTRL+ALT+m

do textu každý iným písmom ideálne otázky ČERVENÝM, meniť sa dá aj pozadie písma takže farieb je dosť.

zachovať formátovanie bude fajn, otázky bold + podčiarknuté.

keď niečo doplníte, odpoveď, zmeníte farbu prvého písmena v otázke(?lol možno blbosť ale tak nejak aby sa dalo rozoznať čo sa zmenilo)

// pd: čo tak changelog na konci dokumentu? **ak nikto nie je proti, prečo nie, ale zatiaľ som veľa zmien teda nezaznamenal**

*** - znamená že daná otázka je vypracovaná v dokumente od starších kolegov na fiitkarovi, v tom obraze.**

Teória

Správne odpovede z testových otázok - doplniť kto vie, vypracovávam podľa fiitkara,

RT0809

1. **b)** Sekvenčný diagram je dynamický diagram ktorým modelujeme v UML časovú následnosť správ
- e)** Entitno-relačný diagram sa používa na datové modelovanie
- f)** Hlavná entita v diagrame tokov údajov je proces

a)F

správne má byť:

Pri tvorbe funkčného modelu sa softvérový systém chápe ako množina funkcií, resp. služieb, ktorými sa naplňuje požadovaný účel, resp. požadovaným spôsobom sa transformujú údaje (alef)

b)T

c)F

správne má byť:

Systém sa modeluje ako sieť procesov, ktoré spracúvajú údaje v systéme. (alef)

d)F

e)T

f)T

v diagrame tokov údajov sú 4 hlavné entity: proces, tok údajov, externá entita, úložisko údajov (alef)

F T F F T T

2. UML sa rozlišuje pomocou

- Ohranicenia
- Oznacene body
- Stereotypy

mechanizmus rozširovania: stereotypy, ohraničenia (angl. constraints), označené hodnoty (angl. tagged values) (alef)

F T T F T F F

3. - Testovanie biela skrinka používa testovacie kritérium založené na analýze štruktúry programu

- Testovanie technikou biela skrinka je dynamická technika

4. - Scenár demom **toto je čo?**

- Agregácia predstavuje slabšiu väzbu medzi dvoma dátovými entitami ako kompozícia
- Scenár definujúci prípad použitia môžeme opísať diagramom činnosti
- scenár definujúci prípad použitia môžeme opísať sekv. diagramom

a)F

UML sa nepoužíva na tvorbu softvéru

b)F -> **FALSE, vyrok nízšie platí pre UML 1.X , pri UML 2.0 už NEPLATÍ**

diagram činností je špeciálny druh stavového diagramu, ktorý znázorňuje tok riadenia medzi činnosťami systému(prednášky)

c) T

agregácia(biely kosoštvorec) je slabšia väzba ako kompozícia(čierny kosoštvorec) (prednášky)

d)F

znázorňuje interakcie medzi objektami v časovej postupnosti -> sekvenčný diagram(alef)

e) //mnicky: podľa mňa nie, dal by som false

f)g) - zatiaľ som to nikde nenasiel(ale sekvenčný je určite) //mnicky: aj diag. činnosti

5. Statické testovanie sa v životnom cykle softveru využíva

- najmä pri analýze a návrhu
- vo všetkých etapách

6.Spravnosť softverového produktu nepostacuje a nie je nevyhnutná

7.Vlastnosti modelov využívajú sa pri ich tvorbe :

- model možno ľahšie zostaviť ako skutočný systém
- môžeme sa rozhodnúť čo budeme v modeli reprezentovať(co zanedbáme)

8.Tok údajov sa môže viesť v diagrame tokov údajov medzi:

- procesom a externou entitou
- dvoma procesmi

9. Uvedte ktorá etapa súvisí v V-modeli životného cyklu soft. s etapou *podrobný návrh*

- a) ak uvažujeme aspekt naslednosti** nasleduje implementácia, predchádza architektonický návrh
- b) ak uvažujeme aspekt abstrakcie** testovanie súčiastok

10) Vymenujte a jednou vetou opíšte každú charakteristiku procesu tvorby SW označovanu ako Unified Process

Unified Process je:

- iterative & incremental - the Elaboration, Construction and Transition phases are divided into a series of timeboxed iterations
- vedený prípadmi použitia- use cases are used to capture the functional requirements and to define the contents of the iterations
- postavený na architektúre - architecture sits at the heart of the project team's efforts to shape the system
- risk focused - focus on addressing the most critical risks early in the project life cycle.

(src: http://en.wikipedia.org/wiki/Unified_process#Unified_Process_Characteristics)

☀ 11) Stručne charakterizujte princíp regresného testovania a uveďte dôvod pre ktorý sa používa

☀ 12) Vymenujte typy zmien softwaru ktoré sa realizujú v etape údržba. Vysvetlite jednou vetou PRECO je dôležité zmeny kategorizovať

- **oprava** - odstraňovanie chýb, ktoré v systéme ešte sú
 - **prispôbenie** - zmeny v časti SW v závislosti podľa zmien HW alebo inej časti SW
 - **zlepšenie** - pridávanie nových vlastností (features)
 - **prevencia** - modifikácie s cieľom zlepšenia ďalšej údržby
- (podľa knihy, str. 92 ďalej)

Zmeny je dôležité kategorizovať kvôli určeniu priorít. Najprv sa robia opravy a potom menej dôležité veci.

☀ 13) Požiadavky v špecifikácii požiadaviek by mali spĺňať vlastnosti ako Spravnosť jednoznačnosť a úplnosť. Vymenujte ďalšie 3 a charakterizujte každú 1 vetou.

- správna
- jednoznačná
- verifikovateľná
- úplná
- konzistentná
- sledovateľná
- modifikovateľná
- zoradená podľa dôležitosti

14) Stručne porovnajte vodopádový model s V-modelom vyvoja sw:

//mnicky: ja by som povedal, že je to to iste, len inak znázornené. Ma niekto **iny názor?**

//Azures: V-model je špeciálny prípad vodopádového modelu

☀ 15) Vysvetlite čo znamená neviditeľnosť sw a aké problémy spôsobuje:

- nevizualizovateľnosť, nemožnosť opísať vlastnosti sw predtým ako ho máme, dôsledok => **syndrom 90% hotovo**

☀ 16) Uveďte cieľ mutačného testovania

cieľom je vytvoriť doveryhodné testovacie vstupy (kniha, s. 87)

PSI skuska 2007.2008A1

1 a,b totožne ako hore

2. -Hrany v diagrame prípadov použitia reprezentujú interakcie
-Diagram komunikácií je semanticky ekvivalentný so sekv. diagr.
- 3 -Testovanie technikou biela skrinka používa testovacie kritérium založené na analýze štruktúry programu

4 model životného cyklu softveru

- definuje jednotlivé činnosti, ktoré treba vykonať
- stanovuje odporúčania pre tvorbu softveru

5 vlastností dobrého návrhu

- volne viazané súčiastky, konzistentné mená súčiastok
- veľká súdržnosť, mála závislosť súčiastok

6. označenie prechodu v stavovom diagrame

udalosť [podmienka]/akcia

7. Rozhodovacia tabulka je technika

- na specifikáciu algoritmov
- semanticky ekvivalentna s rozhodovacím stromom

8. Stručne charakterizuj regresné testovanie(princíp) a uveď účel, pre ktorý sa používa.

Princíp: Testovanie všetkých už predtým testovaných častí softvérového produktu po každej vykonanej zmene.

Účel: Zistenie či vykonaná zmena bola pre systém neškodná.

RT0809

15. Vysvetlite čo znamená neviditeľnosť softveru a ako táto vlastnosť spôsobuje problémy

Neviditeľnosť: nevieme jednoznačne opísať SW predtým, ako ho máme tak, aby sme vyjadrili všetky jeho vlastnosti

Neviditeľnosť: Neexistuje akceptovateľný spôsob reprezentácie sw výrobku tak, aby sa pokryli všetky jeho aspekty, dokonca ani nevieme čo z príslušnej reprezentácie chýba.

Problémy: Spôsobuje to problém z celkovým pochopením systému, vznikajú nezrovnalosti medzi jednotlivými členmi vývojového tímu aj zákazníkmi, každý si môže určitú časť softvérového systému vyložiť iným spôsobom.

16. Jednou vetou uveďte cieľ mutacného testovania

Mutacné testovanie sa používa pri návrhu správnych testovacích vstupov(správne testovacie vstupy sú také, ktoré odhalia chyby v systéme)

17. Zaradte do metódy tvorby softvéru metodológiu SSADM(uveďte z akého prístupu vychádza a jednou vetou to objasnite)

http://en.wikipedia.org/wiki/Structured_Systems_Analysis_and_Design_Method

vychádza z vodopádového modelu

- Structure software analysis and design method
 - metodológia štruktúrovanej analýzy a návrhu systémov
 - oddeľuje fyzický a logický model systému
- základné modely systému ktoré sa vytvárajú :
- model údajov (diagram modelu údajov)
 - funkčný model (diagram tokov údajov)
 - dynamický model (graf životného cyklu entity)

18. Porovnajte tieto typy testovania produktu : akceptačné testovanie, alfa testovanie, beta testovanie (ale práve to, že bieliková chce striktne tie veci, kde sú rozdiely čiže v tomto prípade v čom sa líši akceptačné, alfa a beta teda)

☀ 19a. Uvedte a vysvetlite pristupy k verziovaniu sofveru. Vyjadrite sa k vyhodam a nevychodam jednotlivych pristupov

Distribuvany (Git)

- *uzivatelia si stiahnu vsetky verzie (cely repozitar) do svojho PC a s nimi nasledne pracuju*
- **vyhoda:** uzivatel moze pracovat aj ked nie je pripojeny k internetu
- **nevychoda:** uzivatel moze nevedomky pracovat so starsiou verzou, ako je ta, ktora je na serveri

Centralizovany (SVN)

- *existuje jeden globalny server, v ktorom su ulozene vsetky verzie (jeden repozitar). Uzivatel nasledne moze tieto verzie upravovat, a ked chce zmeny publikovat, tak ich hodi naspat na server.*
- **vyhoda:** uzivatelia vzdy pracuju s najaktualnejšou verzou dokumentu.
- **nevychoda:** vyššie riziko zničenia servera, čím by sa znicili aj všetky súbory na ňom
vyššie riziko kolízií (pretože sa commituju veľké časti)

19b. Vymenujte a jednou vetou opiste kazdu charakteristiku procesu tvorby sofveru oznacovaneho ako RUP

Rational Unified Process má 3 základné princípy:

- model je iteratívny a inkrementálny
 - je vedený/riadený prípadmi použitia
 - je postavený na architektúre; sústreďuje sa na architektúru
- del.**

chce vedieť charakteristiky RUP vid' Unified Process Characteristics http://en.wikipedia.org/wiki/Unified_Process

☀ 20(cast2). Vymenujte techniky UML 2.0 ktore sa pouzivaju na modelovanie spravania stavovy diagram, diagram cinnosti, interakcne diagramy: d sekvencii, d komunikacii, d prehladu interakcii, casovy d, d pripadov pouzitia

☀ 21(cast2). Vystizne charakterizujte tieto pojmy, uvedte o co ide(metoda,technika,entita), vyznam a pouzitie

DIAGRAM KOMUNIKACII

-zobrazuje komunikáciu(posielanie správ) medzi objektami

ROZHODOVACI STROM

- je ekvivalentný s rozhodovacou tabuľkou
- používa sa pri analýze algoritmov

22(cast2).Porovnajte techniky Stavový diagram a Diagram činnosti

23(cast2). Vysvetlite použitie stereotypov <<include>> a <<extend>> pri tvorbe diagramu prípadov použitia na jednoduchom Vami vytvorenom príklade použitia.

<<include>> -- prípad použitia je zahrnutý v inom prípade použitia; vykonáva sa vždy

<<extend>> -- prípad použitia sa vykonáva spolu s druhým prípadom použitia, ale iba niekedy, pri splnení istej podmienky

☀ 25. Vysvetlite prečo vytvárame modely softveru, uveďte ako dimenzie sa sledujú pri modelovaní softverových systémov a diskutujte(??) potrebu viacerých dimenzií pri modelovaní soft. systémov.

Modely vytvárame za účelom

- lepšej reprezentácie systému
- návrhu systému
- reprezentácie systému zákazníkovi
- odstránenie problémov pri špecifikácii systému

ZDROJE

<http://www2.fiit.stuba.sk/~bielik/courses/msi-slov/kniha/2008/essays/msipapersource13-varga.pdf> - statické vs dynamické testovanie

//Azures: Niečo z mojich poznámok z prednášok, pravdepodobne sa vyskytnú na skúške, mám to podčiarknuté v zošite:

Štúdiá vhodnosti: je to rýchla a krátka analýza požiadaviek, ktorej cieľom je urobiť rozhodnutie áno/nie. Má nám zistiť, či sme schopný daný SW vytvoriť.

Aká má byť požiadavka? - zoradenie podľa dôležitosti:

1. zoradená podľa požiadaviek (podľa Bielikovej najpodstatnejšia vec) - je nutné priradiť každej požiadavke prioritu
2. až 4. (približne rovnako podstatné, kúsok za prvou)
správna - vyjadruje sa k tomu zákazník

verifikovateľná - merateľná; bola/nebola splnená?

konzistentná - nie sú tam protirečenia

5. až 8. (menej podstatné, ale sú tam)

jednoznačná - veci sa nedajú vysvetliť inak

úplná

sledovateľná - počas životného cyklu viem, kde sa mi premieta daná požiadavka

modifikovateľná

Pri modelovaní systému máme viacero dimenzií/pohľadov na systém, čo modelujeme; 4 základné pohľady:

1. - statický model

- dynamický model

2. - fyzický model

- logický model

3. model podľa etapy životného cyklu SW

4. pohľad na systém, čo z neho modelujem

- funkcionálny model, model údajov, model správania, objektový model

Odberatelia dokumentu (tí ľudia, čo budú pracovať s dokumentáciou k projektu, bolo to na prednáške z Unicorn-u) – vývojári, tester, odberateľ (už neviem, prečo som si toto zapísal, asi sa tým myslí zákazník), analytici, projektový manažér

Prípady použitia popisujú GUI; treba mať ku každému prípadu použitia hlavný a alternatívny tok udalostí, taktiež vstupné a výstupné podmienky GUI.

Keď bude niekde na teste otázka "Porovnajte vec A a vec B" tak treba napísať:

Spoločné: v odrážkach vymenovať spoločné znaky; stačí aj iba jedna spoločná vec

Rozdielne: vypísať rozdielne veci

Keby bola taká nejaká otázka, pravdepodobne testová, tak: Fázy vývoja SW nemôžu byť paralelné. (bolo to spomínané v súvislosti s UP)

Dobrý návrh systému: voľne zviazané súčiastky a veľká súdržnosť.

Rozhodovacie pravidlo:

Pri rozhodovacej tabuľke: jeden stĺpec tabuľky

Pri rozhodovacom strome: 1 cesta od koreňa k listu v strome