

1. Čo je pre systém dôležitejšie a prečo – spoľahlivosť alebo výkonnosť (dependability vs. performance)?

Spoľahlivosť, lebo výkonný ale nespoľahlivý systém nebude akceptovaný/používaný

2. Aká je najčastejšia/hlavná príčina zlyhania systémov?

Chyba v hardvéri

Chyba v softvéri

Chyba obsluhy

3. So zvyšovaním úrovne spoľahlivosti narastajú aj náklady na vývoj takéhoto spoľahlivého systému. Aké sú dve hlavné príčiny za týmito nákladmi?

Dražší vývojári, techniky vývoja

Extra validácia

4. *Vymenujte tri základné modely softvérového procesu.

Waterfall (vodopádový)

Incremental (iterative) development

Integration and configuration (z existujúcich komponentov)

5. Pri inkrementálnom vývoji je menšie riziko zlyhania celého projektu. Aké sú dva hlavné dôvody?

Používanie – najžiadanejšia funkcionálna ide prvá a už ju zákazník používa od začiatku

Testovanie – najžiadanejšia/najdôležitejšia funkcionálna sa teda aj najviac otestuje

6. Aké sú tri hlavné aktivity projektového riadenia?

Plánovanie projektu

Spĺňanie požiadaviek/specifikácie

Riadenie rizík

Odhady nákladov

Zabezpečenie techniky (HW)

Riadenie ľudí

7. Ktoré tri základné veci musí obsahovať projektový plán?

Míľniky

Časový plán (kedy)

Analýza rizík

Rozdelenie úloh(čo)

Náklady

Zdroje (kto)

8. Aké sú dve hlavné plánovacie aktivity pri agilnom vývoji z pohľadu projektového plánovania?

Iteration planning (plánovanie najbližšieho kola)

Release planning(plánovanie dodávky)

9. Aké sú hlavné riziká v projektovom riadení vývoja softvéru?

Projektové

Produktové

Podnikateľské

10. Čo popisuje dokument s používateľskými požiadavkami a dokument so systémovými požiadavkami?

User reqs – špecifikácia pre zákazníka

System reqs – detailná (rozšírenie user reqs) špecifikácia pre vývoj

11. Uved'te aké sú hlavné kategórie nefunkcionálnych požiadaviek pri ich delení a uved'te príklad z každej kategórie

Product (produktové)

Organizational (organizačné – zákazník aj vývojár)

External (externé)

12. Ktorým UML diagramom sa dá zakresliť architektúra systému?

Diagram tried

Nie UML: box and lines, entito-relačný

13. Uved'te možné architektonické spôsoby ukladania a zdieľania údajov (OVER RIT!!)

Centrálne úložisko

Distribúované (každý syst. má vlastné) úložisko

14. Uved'te možné architektonické prístupy na delenie systémov (na menšie časti)

Objektový model

Data-flow / pipeline / pipe and filter model

15. Uved'te možné spôsoby centrálného riadenia subsystémov

Call-return model (sekvenčné riadenie)

Manager model (paralelné riadenie)

16. Uved'te možné spôsoby riadenia sub-systémov pri riadení prostredníctvom udalostí

Broadcast model (rozposielanie udalostí)

Interrupt-driven model (prerušenia)

17. Vymenujte aspoň dva z troch generických modelov softvérového procesu

Vodopádový model (The waterfall model)

Evolučný vývoj (Evolutionary development)

Vývoj založený na existujúcich komponentoch (Componentbased software engineering)

18. Ktoré pohľady (perspektívy) treba zohľadniť pri návrhu architektúry systému?

logical

process

physical

development

19. Vymenujte štyri základné aktivity softvérového procesu

špecifikácia

návrh a implementácia

validácia

evolúcia

20. Spárujte nasledovné príklady systémov podľa typu ich vážnosti

Systém na správu účtov v banke - Business-critical system

Navigačný systém vesmírnej lode - Mission-critical system

Riadiaci systém v jadrovej elektrárni - Safety-critical system

21. Ktoré fundamentálne aktivity sú spoločné pri každom softvérovom procese?

Requirements analysis and definition

System and software design

Implementation and unit testing

Integration and system testing

Operation and maintenance

Software specification

Software design and implementation

Software validation

Software evolution

22. Označte nasledovné požiadavky podľa ich typu:

F Systém musí archivovať originály fotografií, ktoré používateľ zmení

F Systém musí umožniť používateľovi vyhľadávať fotky podľa miesta odfotenia

N Systém musí spĺňať požiadavky paragrafu 7 zákona 428 / 2002 Z.z.

F/N Systém musí poskytnúť komfortný prehliadač fotiek

N Systém musí byť spustiteľný na počítačoch aj mobilných zariadeniach ← *je to správne???*

Ano, pozri <https://goo.gl/41fXq2> // mas viacero takychto opravenych testov??, mozes ich pridať hore na začiatok docu
nemam

23. Ktoré diagramy patria do ktorého systémového modelu? Poprepájajte.

Kontextový model: Diagram architektúry

Model údajov: Entitno-relačný diagram

Model správania: Stavový diagram

Diagram použitia (use-case)

Diagram toku údajov (data-flow)

Sekvenčný diagram

Diagram aktivít

24. Čo je to model? Čo je model softvérového procesu?

Model= súbor objektov reprezentovaných PC systémom spolu s ich vlastnosťami a vzťahmi

Model softvérového procesu= Zjednodušená reprezentácia procesu softvéru, vychádzajúca (vyplývajúca) z určitého špecifického pohľadu.

25. Vymenujte základné atribúty, ktoré by mal mať dobrý softvér.

Maintainability(udržiavateľnosť), Dependability (spoľahlivosť), Efficiency (účinnosť), Acceptability (akceptovateľnosť)

26. Ktoré problémy existujú pri použití prirodzeného jazyka v špecifikácii požiadaviek?

Nedostatok jasnosti, Zmätok v požiadavkách, Zlúčenie požiadaviek

27. Aké diagramy (UML alebo iné, uveďte aspoň po jednom príklade) sa používajú na popis sémantického dátového modelu?

Abstract Hierarchy diagram

28. Čo je to proces? Čo je to softvérový proces?

Proces= inštancia programu vrátane všetkých hodnôt premenných a stavu

Softver.proces= Súbor činností, ktorých cieľom je rozvoj alebo vývoj softvéru

29. Vymenujte 4 hlavné vlastnosti (atribúty) softvérového systému spadajúce pod spoľahlivosť.

Spoľahlivosť, bezpečnosť, ochrana, dostupnosť, chybová tolerancia, opraviteľnosť, udržiavateľnosť, pružnosť

30. Vymenujte alternatívy k prirodzenému jazyku, ktoré slúžia na špecifikáciu systémových

Ways of writing a system requirements specification



Notation	Description
Natural language	The requirements are written using numbered sentences in natural language. Each sentence should express one requirement.
Structured natural language	The requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement.
Design description languages	This approach uses a language like a programming language, but with more abstract features to specify the requirements by defining an operational model of the system. This approach is now rarely used although it can be useful for interface specifications.
Graphical notations	Graphical models, supplemented by text annotations, are used to define the functional requirements for the system; UML use case and sequence diagrams are commonly used.
Mathematical specifications	These notations are based on mathematical concepts such as finite-state machines or sets. Although these unambiguous specifications can reduce the ambiguity in a requirements document, most customers don't understand a formal specification. They cannot check that it represents what they want and are reluctant to accept it as a system contract

požiadaviek.

30/10/2014

Chapter 4 Requirements Engineering

43

31. Aké diagramy sa používajú na popis architektúry SW? Uved'te aspoň 1 UML a aspoň 1 nie-UML diagram.

UML: Class Diagram

nie UML :Box a line diagramy

Doplniť:

1. Vymenujte, ktoré základné vlastnosti musí spĺňať spoľahlivý (dependable) system (4)

Maintainability

Dependability and security

Efficiency

Acceptability

//nemalo by to byť?// to je to iste skoro

- Dostupnosť
- Spoľahlivosť
- Bezpečnosť
- Zabezpečenie
- Životaschopnosť

2. Aký je rozdiel medzi dvomi spoľahlivosťami (dependability vs reliability)?

reliability - pravdepodobnosť že system korektne doruci zakaznikom pozadovany softver

dependability - skladá sa z ďalších vecí, patrí tam napr. (reliability, safety, security,...)

Skúška z pred 2 rokov

1. Aký je rozdiel medzi dvomi oblasťami spoľahlivých systémov?

- **Security:**

- schopnosť ochrániť systém pred náhodným alebo zámerným vonkajším útokom

- **Safety:**

- Zameriava sa na ochranu pred náhodnými chybami, ktoré by mohli viesť k ohrozeniu zdravia alebo majetku. Ide o to, aby systém nezlyhal spôsobom, ktorý by mohol poškodiť ľudí alebo prostredie.

2. Aké sú dve hlavné výhody implementácie najžiadanejšej/najprioritnejšej funkcionality ako prvej pri inkrementálnom vývoji?

- hlavná funkcionality softvéru je implementovaná ako prvá a softvér vieme aj reálne využívať
- hlavná funkcionality sa tým pádom aj najdlhšie testuje

3. Ktoré tri základné veci musí obsahovať projektový plán?

- časový plán (kedy)
- zdroje (kto)
- rozdelenie úloh (čo)

4. Aké tri triedy projektových rizík poznáte?

- projektové
- produktové
- podnikateľské (obchodné, biznis)

5. Prepíšte cieľ na materiálnu požiadavku: Vyhľadávanie v údajoch má byť rýchle

- transakcie/s, reakčný čas
- napr. odozva na vyhľadávanie do 3s pri 1000 súčasných požiadavkách

6. Aký je rozdiel medzi používateľskými a systémovými požiadavkami?

- **používateľské:**
 - sú písané v prirodzenom jazyku
 - diagramy služieb ktoré poskytuje systém a ich operačné obmedzenia
- **systémové:**
 - štruktúrovaný dokument popisujúci detailný opis systémových funkcií, služieb a operačných obmedzení
 - Definuje čo má byť implementované, takže môže byť časť kontraktu medzi klientom a dodávateľom

7. Aké dva architektonické štýly poznáte na centralizované riadenie?

- manager model
- call-return model

8. Aké dva architektonické štýly poznáte na riadenie prostredníctvom udalostí?

- broadcast model
- interrupt-driven model

9. Uved'te aspoň 3 princípy agilných metód

- Zamerajte sa na kód, skôr než navrhovanie;
- Sú založené na iteratívnom prístupe k vývoju softvéru;
- Sú určené na rýchlu dodávku pracovných programov a vyvíjať ich tak rýchlo, aby spĺňali meniace sa požiadavky.

10. Čo je to refaktORIZÁCIA?(refactoring)

- Je proces zlepšovania kódu, kde je kód reorganizovaný a prepísaný, aby bol efektívnejší

11. Akým spôsobom sa rieši testovanie v XP?(stačí názov metódy)

- Test-first vývoj
- Písanie testov pred kódom upresňuje požiadavky, ktoré majú byť realizované.

12. Aký je rozdiel medzi konfiguráciou softvéru v čase dodania a v čase návrhu?

- **Deployment-time configuration:** všeobecné (generic) systém je nakonfigurovaný vložení znalosti požiadaviek zákazníka a obchodných procesov(?). Softvér sám v sebe sa nezmení
- **Design-time configuration:** obyčajný generic kód je spravený a zmenený podľa požiadaviek jednotlivých zákazníkov

13. Čo popisuje a načo sa dá využiť návrhový vzor?

- je to znovu použitie abstraktných vedomostí a problému(?) a jeho riešení
- vzor by mal byť dostatočne abstraktný aby mohol byť použitý v rôznych situáciach
- vzor sa často spoliehaako dedičnosť a polymorfizmus //pičovina
- predstavuje všeobecné riešenie problému , ktoré sa používa pri návrhu počítačového programu(nie je knižnica ,ani časť kódu , je to abstraktný postup)

14. Aký je rozdiel medzi validovaním softvéru a testovaním defektov?

- **validačné testovanie:** demonštrovať tvorcom a zákazníkom, že softvér splňuje požiadavky
- **chybové (defekt) testovanie:** zistenie chýb a kazov v softvéri, kde správanie programu nie je správne, alebo nie je v súhlade so špecifikáciou

15. Správne priradiť kategóriu testov (A-whitebox, B-blackbox, C-iné)

- Release testing - C
- Štruktúrne testovanie - A
- Partition testing - B
- Path testing - A

Skúška spred x rokov

1. Ako môže operátor ovplyvniť systém tak, že sa systém stane nebezpečným pre zdravie alebo unsafe, napriek tomu, že systém certifikovaný ako spoľahlivý (reliable)?

- Nesprávne používa

2. Prečo je dôležité počas fázy špecifikácie požiadaviek spísať používateľskú aj systémovú špecifikáciu?

- Aby sa predišlo tomu, že zákazník dostane produkt, ktorý nespĺňa jeho požiadavky. Požiadavky by mali byť súčasťou kontraktu pre vývoj softvéru. Preto je podstatné, aby tieto požiadavky boli tak úplné, ako sa dá. (aj používateľská aj systémová špecifikácia)

3. Ktoré tri základné veci musí obsahovať projektový plán?

- Rozdelenie úloh
- Zdroje
- Časový plán

4. Konfiguračný manažment zahŕňa:

- dokumentovanie systémových nastavení
- periodickú údržbu
- baseline systémy

5. Kategorizujte požiadavky na funkcionálne (F), nefunkcionálne, organizačné (NF-org), nefunkcionálne produktové (NF-prod), a nefunkcionálne externé (NF-ext)

- Systém musí byť naprogramovaný v Java - NF-org
- Systém musí umožniť používateľovi vyhľadávať fotky podľa miesta odfotenia - F
- Systém musí spĺňať požiadavky 7. zákona 4.28/2002 2.2 - NF-EXT
- Systém musí mať dostatočne rýchlu odozvu pri vyhľadávaní - NF-prod
- Systém musí byť dostupný na počítačoch aj mobilných zariadeniach - NF-PROD

6. Chovanie systému sa dá dynamicky popísať o dvoch pohľadoch, ktoré to sú?

- Spracovanie dát
- Stavové modely

7. Kedy by ste použili na spracovanie udalosti systém rozposielania (broadcast) a kedy prerušenia (interrupts)?

- **Broadcast** – účinný pri integrácií subsystémov na rôznych počítačoch v sieti
- **Interrupts** – používa sa v real-time systémoch, kde rýchla odozva na udalosť je nevyhnutná

8. Akými metódami sa bežne identifikujú objekty zo špecifikácie pri objektovo-orientovanom návrhu?

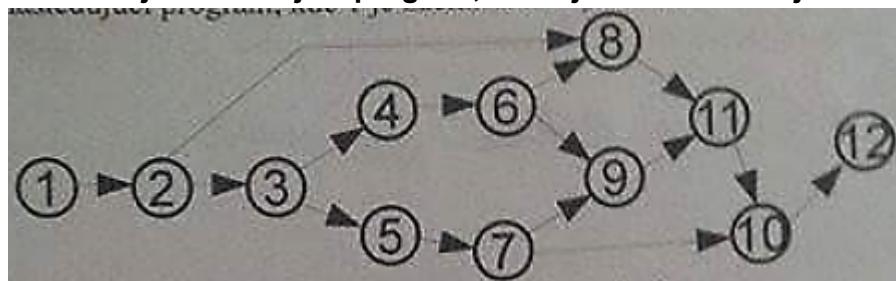
Objekty sa identifikujú podstatnými menami (neviem ako sa volá tá metóda oficiálne)

9. Ako je vyriešené testovanie v XP, kde neexistuje špecifikácia a teda ani testovacie prípady z nej vyplývajúce? Napíšte názov takéhoto testovania aj ho opíšte.

??

asi zapojenie užívateľov v oblasti vývoja a validácie testu

10. Otestujte nasledujúci program, kde 1 je začiatok a 12 je koniec:



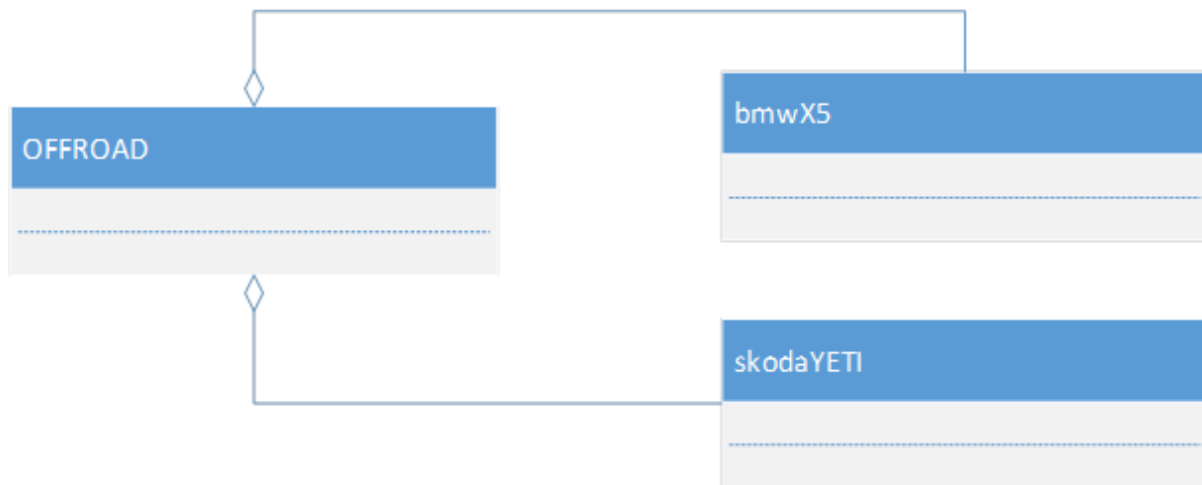
- a) Identifikujte cesty pre metódu testovania ciest (path testing)
 - 1,2,3,5,7,10,12
 - 1,2,3,4,6,9,11,10,12
 - 1,2,8,11,10,12
 - 1,2,3,5,7,9,11,10,12
 - 1,2,3,4,6,8,11,10,12
- b) Označte (podčiarknite) tie cesty identifikované v časti (a), ktoré by sa testovali v prípade použitia metódy štruktúrovaného testovania (structure testing)
-nemali by sa označiť cesty tak, aby sme otestovali každú "nodu" aspoň raz

12. použite gramaticky správnu UML notáciu a zobrazte:

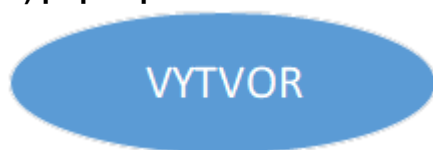
- a) kompozíciu tried srdce a človek ("vlastní")



- b) agregáciu tried bmwX5 a skodaYETI do triedy OFFROAD ("patrí")



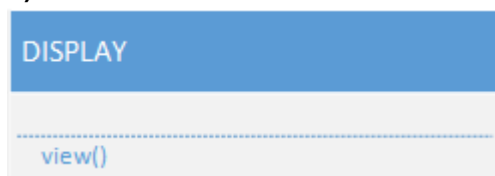
- c) prípad použitia VYTVOR



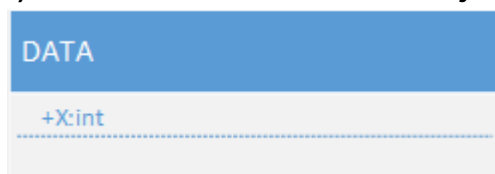
- d) objekt FEI v triede STU



- e) triedu s názvom DISPLAY obsahujúcu metódu view()



- f) triedu s názvom DATA obsahujúcu verejný atribút X typu integer



13. Nakreslite UML diagram tried a zobrazte:

- človek má meno, email, telefónne číslo
- Učiteľ a študent sú ľudia (špecifické typy človeka)
- adresa má ulicu, číslo domu, mesto, PSČ
- človek má práve jednu adresu, ale na jednej adrese môže žiť niekoľko ľudí
- učiteľ učí maximálne 2 predmety
- predmet môže navštevovať ľubovoľný počet žiakov

Použite UML notáciu. Orámujte a pomenujte diagram. Zobrazte všetky násobnosti (multiplicity)
Pomenujte všetky asociácie v diagrame