

## Test A – 1

Meno:

Osobné číslo:

1. 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.

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

Spoľahlivosť  
Bezpečnosť  
Dostupnosť  
Udržiavateľnosť

3. Aké tri hlavné oblasti rizík sa analyzujú v rámci manažmentu rizík v projektovom manažmente?

Projektové  
Podnikateľské  
Produktové

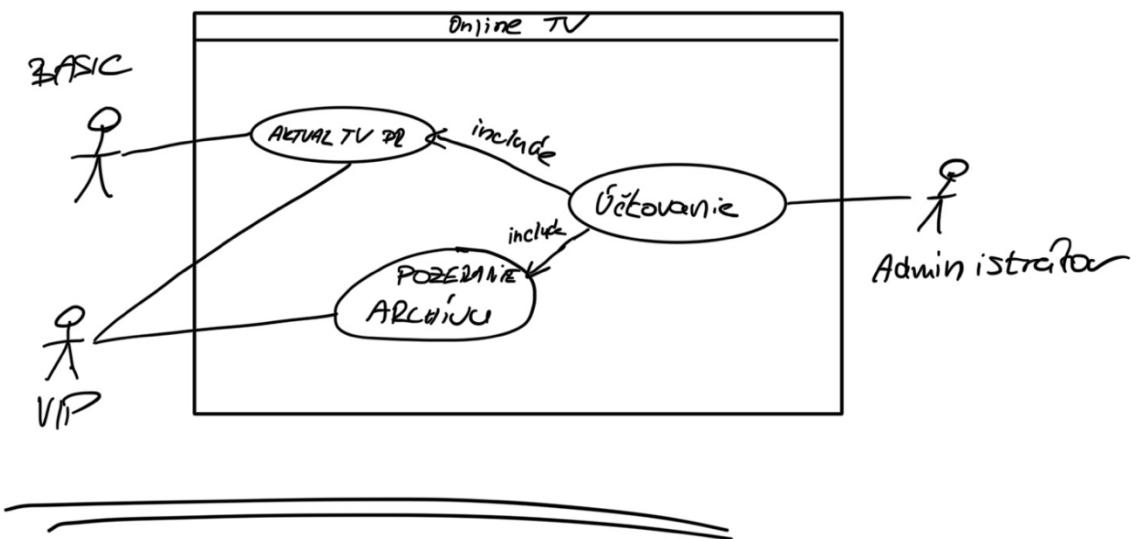
4. Ktoré problémy existujú pri použití prirodzeného jazyka v špecifikácii požiadaviek?

Nedostatok jasnosti  
Zmätok v požiadavkách  
Zlúčenie požiadaviek

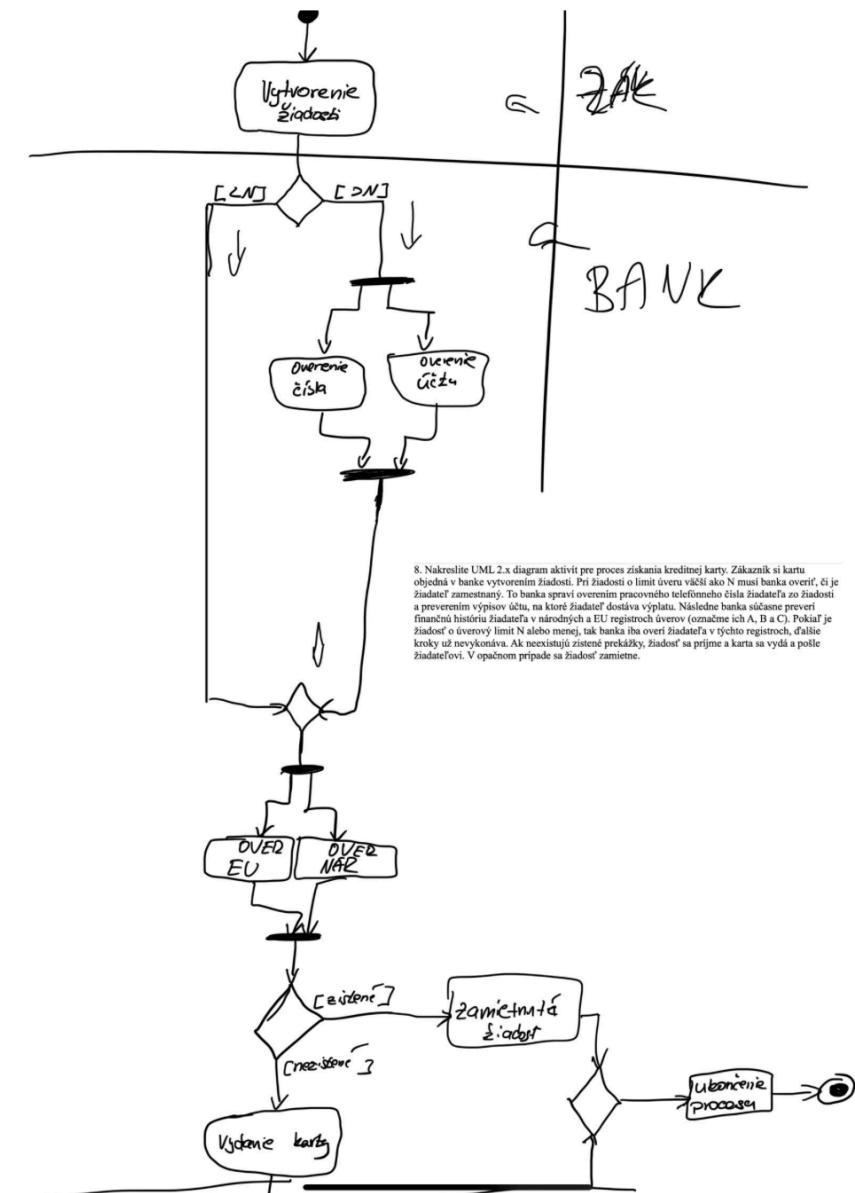
5. Aké diagramy (UML alebo iné, uvedťte aspoň po jednom príklade) sa používajú na popis sémantického dátového modelu?

Abstract Hierarchy diagram

7. Nakreslite diagram použitia pre online TV. Zákazník môže sledovať aktuálny TV program. VIP zákazník je ako zákazník (teda zdieľa rovnaké použitia), ale navyše môže pozerať aj program z archívu. Sledovanie TV aj archívu sa zaznamenáva v účtovnom systéme. Účtovný systém zaznačte ako ďalší prípad použitia v diagrame. V diagrame dodržte UML špecifikáciu verzie 2.x.



8. Nakreslite UML 2.x diagram aktivít pre proces získania kreditnej karty. Zákazník si kartu objedná v banke vytvorením žiadosti. Pri žiadosti o limit úveru väčší ako N musí banka overiť, či je žiadateľ zamestnaný. To banka spraví overením pracovného telefónneho čísla žiadateľa zo žiadosti a preverením výpisov účtu, na ktoré žiadateľ dostáva výplatu. Následne banka súčasne preverí finančnú história žiadateľa v národných a EU registroch úverov (označme ich A, B a C). Pokiaľ je žiadosť o úverový limit N alebo menej, tak banka iba overí žiadateľa v týchto registroch, ďalšie kroky už nevykonáva. Ak neexistujú zistené prekážky, žiadosť sa príjme a karta sa vydá a pošle žiadateľovi. V opačnom prípade sa žiadosť zamietne.



## Test A – 2

Verzia A	Meno:	Osobné číslo:
1. Vymenujte aspoň dve alternatívy k prirodzenému jazykú, ktoré slúžia na špecifikáciu systémových požiadaviek.		
2. Vymenujte tri základné modely softvérového procesu.		

Matematické špecifikácie

Diagramy (napr. UML)

Jazyky na opis dizajnu

Štruktúrovaný prirodzený jazyk

2. Vymenujte tri základné modely softvérového procesu.		
3. Vymenujte, ktoré základné vlastnosti musí splňať spoľahlivý (dependable) systém.		

Vodopádový model

Postupný vývoj

Integrácia a konfigurácia

3. Vymenujte, ktoré základné vlastnosti musí splňať spoľahlivý (dependable) systém.		
4. Aké sú hlavné aktivity projektového riadenia? (zakrúžkujte správne)		

Spoľahlivosť

Bezpečnosť

Dostupnosť

Udržiavateľnosť

4. Aké sú hlavné aktivity projektového riadenia? (zakrúžkujte správne)		
plánovanie projektu	spísanie požiadaviek/špecifikácie	riadenie rizík
odhady nákladov	zabezpečenie techniky (HW)	riadenie ľudi
5. Ktoré základné veci musí obsahovať projektový plán? (zakrúžkujte správne)		

odhady nákladov	zabezpečenie techniky (HW)	analýza rizík	
míľníky	časový plán	zdroje	
rozdelenie úloh	náklady	5. Ktoré základné veci musí obsahovať projektový plán? (zakrúžkujte správne)	
6. Aké sú základné veci, ktoré musia byť v rámci agilného plánovania, čo treba plánovať (aké sú plánovacie aktivity) v agilnom			

- o miľníky
  - o rozdelenie úloh
  - o Casový priebeh
  - o náklady
  - o zdroje
6. Z pohľadu projektového plánovania, čo treba plánovať (aké sú plánovacie aktivity) v agilnom prístupe k vývoju?

Plánovanie vydania (Release planning) – rozhodovanie o funkciách na niekoľko mesiacov dopredu

Iteračné plánovanie (Iteration planning) – plánovanie krátkodobých prírastkov systému

7. Aký je rozdiel medzi používateľskými a systémovými požiadavkami? Prečo je dôležité oddeliť špecifikáciu používateľských a špecifikáciu systémových požiadaviek? [2b]

- user:

**používateľské:**

- o sú písané v prirodzenom jazyku
- o diagramy služieb ktoré poskytuje systém a ich operačné obmedzenia

**systémové:**

- o štruktúrovaný dokument popisujúci detailný opis systémových funkcií, služieb a operačných obmedzení
- o Definuje čo má byť implementované, takže môže byť časť konaktu medzi klientom a dodávateľom

**Dôvod oddelenia:** Jasnejšia komunikácia medzi používateľmi a vývojármi.

8. Čo je pre systém dôležitejšie a prečo – spoľahlivosť alebo výkonnosť (*dependability vs. performance*)? [2b]

Spoľahlivosť, lebo výkonný ale nespoľahlivý systém nebude akceptovaný/používaný

## Test B – 1

Meno:

Osobné číslo:

1. Čo je to proces? Čo je to softvérový proces?

Proces - inštancia programu vrátane všetkých hodnôt premenných a stavu

Softvérový process - súbor činností, ktorých cieľom je rozvoj alebo vývoj softvéru

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

Spoľahlivosť

Bezpečnosť

Dostupnosť

Udržiavateľnosť

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

Matematické špecifikácie

Diagramy (napr. UML)

Jazyky na opis dizajnu

Štruktúrovaný prirodzený jazyk

4. Aké diagramy sa používajú na popis architektúry SW? Uveďte aspoň 1 UML a aspoň 1 nie-UML diagram.

UML: Class Diagram

nie UML: Box a line diagram

5. Aké typy modelov správania sa používajú na popis dynamickej stránky systému?

Sekvenčné diagramy

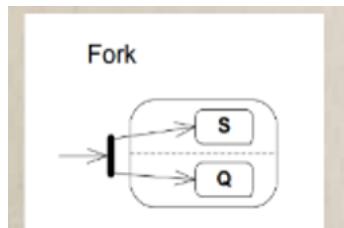
Stavové diagramy

Activity diagramy

---

6. (a) Nakreslite rozvetvenie stavov (fork) v stavovom diagrame, zo stavu A do stavov B a C.

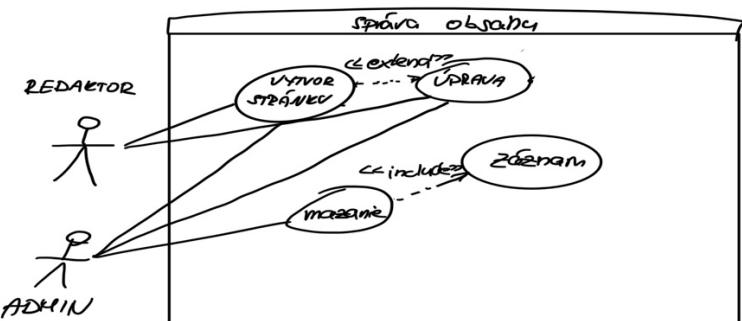
(b) Nakreslite rozvetvenie (fork) v diagrame aktivít, z aktivity A do aktív B a C.



- a) Fork v stavovom
- b) Fork v diagrame aktivít – to isté, len S a Q majú obdlžníky a nie je “rectangle” obvod

7. Namodelujte stavový diagram pre hru šach: Na začiatku je na ťahu biely, následne čierny a opakuje sa to kým nenastane jedna z nasledujúcich troch situácií: biely nepotiahne a nedá šach-mat čiernemu; čierny nepotiahne a nedá šach-mat bielemu; nastane remíza tým, že ktorýkoľvek z hráčov nepotiahne tak, že nie je možé ďalej vykonať ťah podľa pravidiel. Dodržte UML 2.x špecifikáciu.

- 
8. Nakreslite UML 2.x diagram prípadov použitia pre správu obsahu na webe: Redaktor môže (a) vytvárať nové stránky a môže (b) meniť obsah existujúcich. Administrátor je ako redaktor (jeho špeciálnym prípadom), ktorý navyššie môže stránky aj zmazať. Pri zmazaní stránky sa vytvorí informácia, kto stránku zmazal – zaznačte tento prípad použitia v diagrame. Tiež zaznačte, že menenie obsahu existujúcich stránok využíva vytvorenie novej stránky (napr. v prípade, ak stránka pred modifikovaním neexistovala).



## Test B – 2

Verzia B	Meno:		
1. Uveďte aké sú hlavné kategórie nefunkčných požiadaviek.		Osobné č	
organizačné	externé	projektové produktové	
2. Vymenujte štyri základné aktivity softvérového procesu.			
špecifikácia	development	validácia	evolúcia

Produktové

Organizačné

Externé

organizačné	externé	projektové produktové
2. Vymenujte štyri základné aktivity softvérového procesu.		
špecifikácia	development	validácia
3. Vymenujte tri principálne typy kritických (spoločenských) systémov.		evolúcia

Špecifikácia

Návrh a implementácia

Validácia

Evolúcia

specifikácia	development	validácia	evolúcia
3. Vymenujte tri principálne typy kritických (spoločenských) systémov.			
projektové safety-critical	produktové business-critical	podnikateľské mission-critical	
4. Aké triedy projektových rizík poznáte? (zakrúžkujte správne)			

Safety-critical (bezpečnosti)

business-critical (obchodne-kritické)

mission-critical (misie—kritické)

projektové safety-critical	produktové business-critical	podnikateľské mission-critical
4. Aké triedy projektových rizík poznáte? (zakrúžkujte správne)		
technologické	ľudské	podnikateľské
organizačné	produktové	projektové
5. Ktoré základné veci musí obsahovať projektový plán? (zakrúžkujte správne)		

organizačné	produktové	projektové
5. Ktoré základné veci musí obsahovať projektový plán? (zakrúžkujte správne)		
míľníky	časový plán	analýza rizík
rozdelenie úloh	náklady	zdroje

rozdeľenie úlohnákladyzdroje

6. Opište, príp. uveďte príklad, kedy je softvér použiteľný (*usable*) ale nedôveryhodný (*not trusted*).

- SW splňa použiteľské funkcionálne požiadavky, ale nesplňa nie nefunkcionálne alebo domenové požiadavky (napr. je klikateľný funkčný, no nie je za bezpečený)

Kedysi softvér plní svojú funkciu a je pre používateľov jednoduchý na ovládanie, ale existujú pochybnosti o jeho bezpečnosti, ochrane súkromia alebo spoločnosťi – napr. aplikácia na správu hesiel (slabé šifrovanie dát)

7. Aký je rozdiel medzi dvomi spoľahlivosťami (*dependability vs. reliability*)? [2b]

- *dependability*: počasťava z viacerých vlastností ako je dole uvedená *reliability*, ale aj z bezpečnosti, zabezpečenia, dostupnosti atď.
- *reliability*: pravdepodobnosť, že systém bude dodaný zadaným podľa jeho zadanej špecifikácie

**Dependability** – skladá sa z ďalších vecí, patrí tam napr. realibility, safety, security..

**Realibility** – pravdepodobnosť, že systém korektné doručí zákazníkom požadovaný softvér

8. Aký je rozdiel medzi funkcionálnymi a nefunkcionálnymi požiadavkami? [2b]

- *functional*: čo sa očakáva od funkcionality systému, ako bude spracovať použiteľské vstupy
- *non-functional*: obmedzenia systémov na rôznych úrovniach, napr.
  - organizácia (ako sú politiky danej organizácie)
  - produktová (na akých systémoch bude berieť)
  - externá (rôzne faktory)

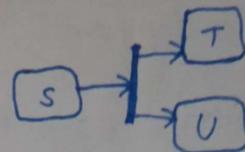
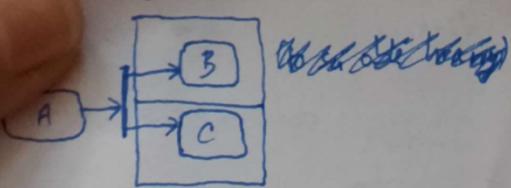
**Functional** – popisujú, čo systém robí a ako reaguje (popisujú konkrétné funkcie, úlohy a správanie softvéru) – teda popisujú konkrétné funkcionality

**Non-functional** – popisujú, ako má systém fungovať – popisujú vlastnosti ako výkonnosť, bezpečnosť, použiteľnosť... - popisujú kvalitatívne štandardy systému

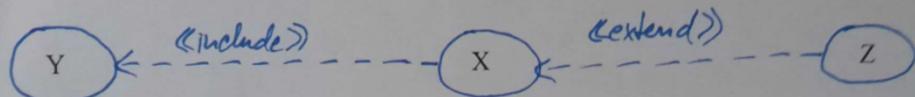
V nasledovných úlohách sa bude hodnotiť logická správnosť aj UML gramatika - správne geometrické tvary, popisy diagramov, particii, rozlišovanie objektov a tried, a/synchronných správ, použitie direktív a pomocných útvarov ako sú pseudostavy, rozvetvenia, cykly, podmienky, a pod.

9. Použite gramaticky správnu UML notáciu a zobrazte:

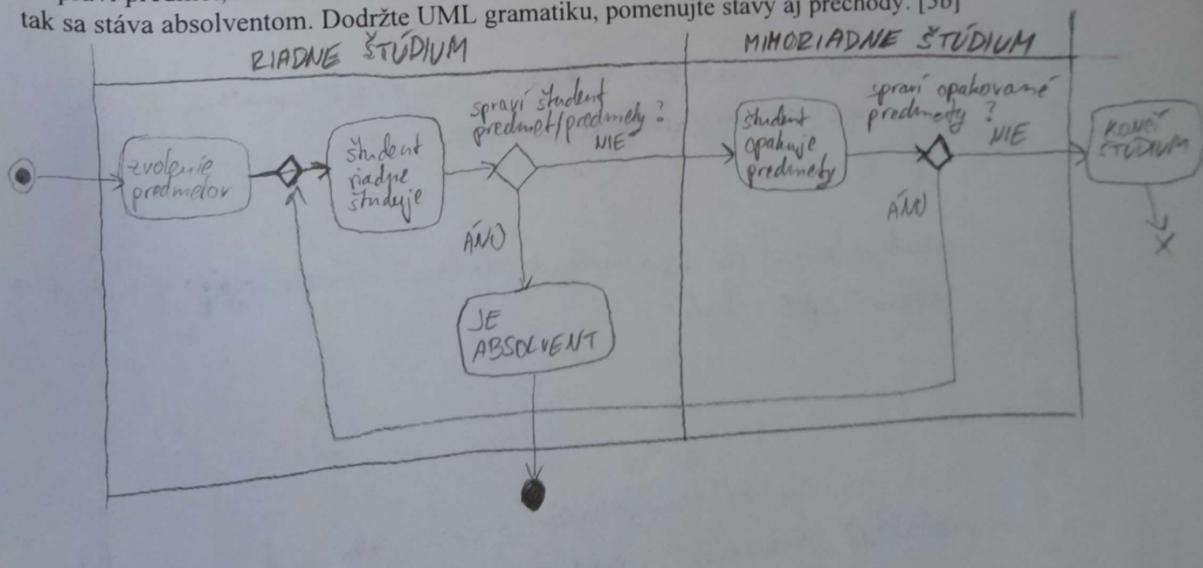
(a) rozvetvenie (fork) aktivity A do aktivít B a C | (b) rozvetvenie (fork) stavu S do stavov T a U



(c) prípad použitia X; prípad použitia Y, ktorý sa vykoná vždy keď X; a prípad použitia Z, ktorý sa nemusí vykonať vždy keď X [3b]



10. Namodelujte UML stavový diagram pre štúdium: Na začiatku, po zvolení predmetov, študent študuje riadne. Ak spraví ďalšie predmety, stále študuje riadne. Ak nejaký predmet nespráví, ale ho opakuje, tak študuje mimoriadne. Ak opäť spraví všetky predmety, študuje riadne. Ak opakovane nespráví predmet, končí štúdium. Ak spraví všetky predmety, či už na prvý alebo opakovaný pokus, tak sa stáva absolventom. Dopržte UML gramatiku, pomenujte stavy aj prechody. [5b]



## Test C

Verzia C

Meno:

Osobné číslo:

1. Vymenujte tri základné modely softvérového procesu.

Vodopádový model

Postupný vývoj

Integrácia a konfigurácia

2. Vymenujte tri principiálne typy kritických (spoľahlivých) systémov.

3. Aká je hlavná príčina zlyhania systémov? (zakružkujte správne)

Safety-critical (bezpečnosti)

business-critical (obchodne-kritické)

mission-critical (misie—kritické)

3. Aká je hlavná príčina zlyhania systémov? (zakružkujte správne)

chyba v hárveri

chyba v softvéri

chyba obsluhy

Chyba obsluhy

4. Ktoré sú dva hlavné dôvody extra nákladov pri vývoji spoľahlivých systémov (*dependable systems*)?

Aké sú hlavné aktivity projektového riadenia?

Drahé vývojové techniky

Zvyšené testovanie

Systemová validácia

5. Aké sú hlavné aktivity projektového riadenia? (zakružkujte správne)

plánovanie projektu

spísanie požiadaviek/specifikácie

odhadý nákladov

zabezpečenie techniky (HW)

riadenie rizík

riadenie ľudi

6. Ktoré základné veci musí obsahovať projektový plán?

	<u>Zložky projektového plánu (HW)</u>	<u>riadenie</u>
6. Ktoré základné veci musí obsahovať projektový plán? (zakružkujte správne)		
<input checked="" type="checkbox"/> zdroje	<input checked="" type="checkbox"/> časový plán	
<input checked="" type="checkbox"/> rozdelenie úloh	<input checked="" type="checkbox"/> náklady	analýza rizik miľníky
7. Napíšte štyri hlavné aktivity pre vývoj SW	náklady	miľníky
7. Napíšte štyri hlavné aktivity pre vývoj SW systému a čiarami ich graficky naviažte na aktivity vodopádového modelu. [2b]		
1. definovanie požiadaviek	• specifikacia	
2. návrh systému a softvéru	• vývoj	
3. implementácia	• validácia	
4. testovanie	• evolúcia	
5. prevádzka a spravovanie		

Co je softvér?

8. Čo je softvér? Aká je hlavná úloha softvérovo inžinierstva? [2b]

Softvérovo inžinierstvo je inžinierská disciplína, ktorá sa zaobrá teóriami, metódami a nástrojmi pre profesionálny vývoj softvéru

Zameriava sa na nákladovo efektívny vývoj softvéru

Zahŕňa všetky aspekty výroby softvéru, od počiatočnej špecifikácie systému až po údržbu systému po jeho uvedení do prevádzky.

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 takého spoľahlivého systému. Aké sú dve hlavné príčiny za týmito nákladmi?

Drahší 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 funkčionalita ide prvá a už ju zákazník používa od začiatku*

*Testovanie – najžiadanejšia/najdôležitejšia funkčionalita sa teda aj najviac otestuje*

**6. Aké sú tri hlavné aktivity projektového riadenia?**

*Plánovanie projektu*

*Splňanie požiadaviek/špecifikácie*

*Riadenie rizík*

*Odhady nákladov*

*Zabezpečenie techniky (HW)*

*Riadenie ľudí*

**7. Ktoré tri základné veci musí obsahovať projektový plán?**

*Mílniky*

*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. Uvedte aké sú hlavné kategórie nefunkcionálnych požiadaviek pri ich delení a uvedte 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. Uvedte možné architektonické spôsoby ukladania a zdieľania údajov (OVER RIT!!)**

*Centrálné úložisko*

*Distribuované (každý syst. má vlastné) úložisko*

**14. Uvedte možné architektonické prístupy na delenie systémov (na menšie časti)**

*Objektový model*

*Data-flow / pipeline / pipe and filter model*

**15. Uvedte možné spôsoby centrálneho riadenia subsystémov**

*Call-return model (sekvenčné riadenie)*

*Manager model (paralelné riadenie)*

**16. Uvedte 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**

*specifikácia*

*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 splňať 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 ↘

spravne???

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

**23. Ktoré diagramy patria do ktorého systémového modelu? Popredajte.**

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      model - abstraktna reprezentacia systemu pre rozne pohlady  
                          model SW procesu - abstraktna reprezentacia procesu vyvoja softveru  
                          z konkretnej perspektivy

**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 (spoloahlivosť), 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, uvedťte aspoň po jednom príklade) sa používajú na popis sémantického dátového modelu?**

*Abstract Hierarchy diagram*

UML: class diagram  
nonUML: entity-relational 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  
sw proces - strukturovany subor cinnosti potrebnych na vyvoj SW systemu

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

**Spoločnosť, bezpečnosť, ochrana, dostupnosť, chybová tolerancia, opraviteľnosť, udržiavateľnosť, pružnosť**

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

**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

**31. Aké diagramy sa používajú na popis architektúry SW? Uvedťte 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 systém korektnie doručí zakazníkom požadovaný softver  
dependability - skladá sa z ďalších vecí, patrí tam napr. (reliability, safety, security,...)

**Skúška z pred 2 rokov**

**1. Aký je rozdiel medzi dvomi oblastami 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 funkciu softvéru je implementovaná ako prvá a softvér vieme aj reálne využívať
- hlavná funkciu 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. Uvedte 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 splňali 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 vložením znalostí 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 znova použitie abstraktných vedomostí a problému(?) a jeho riešení
- vzor by mal byť dostatočne abstraktný aby mohol byť použitý v rôznych situáciach
- vzor sa často spolieha .....ako 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 tvorcovi a zákazníkovi, 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 priradte kategóriu testov (A-whitebox, B-blackbox, C-iné)**

- *Release testing - C*
- *Štrukturálne 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 nesplňa 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 zahrňa:**

- 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 Jave - NF-org
- Systém musí umožniť používateľovi vyhľadávať fotky podľa miesta odfotenia - F
- Systém musí splňať 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 popísť 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 intergrácii 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?**

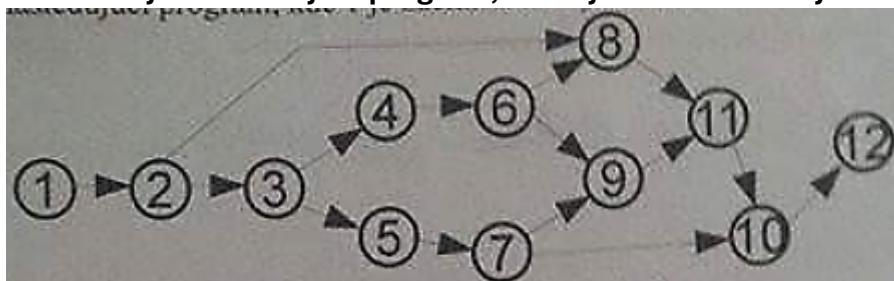
Objekty sa identifikujú podstatnými menami (neviem ako sa volá tá metóda oficiálne)

**9. Ako je vyriešené testovanie v XP, kde neexistuje špecifikácia a teda ani testovacie prípady z nej vyplývajúce? Napíšte názov takého 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 kazdu "nodu" aspon 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

## VYTvor

- d) objekt FEI v triede STU

FEI:STU

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

DISPLAY

view()

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

DATA

+X:int

### 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