

## Предавање 1

1. Што означува правилото “3-year half life” во знањата за развој на софтвер?  
A) Пола од тоа што треба да го знаете денес ќе биде застарено за 3 години  
B) Играњето на half-life во времетраење од 3 години ќе ве направи подобар софтвер инженер  
C) Секои три години како софтверски инженер ви го кратат животниот век за половина  
D) За 3 години ќе знаете пола од тоа што го знаете денес
2. Согласно извештајот на Global Information Technology Industry, глобалната вредност на ИТ индустријата надминува 5 трилиони \$. Споредена со земјоделието, ИТ индустријата вреди:  
A) Повеќе од двојно  
B) Приближно еднакво  
C) Скоро двојно повеќе  
D) Половина од вредноста на земјоделието
3. Што од следното е најмалку веројатно да предизвика пад на сигурноста (веродостојноста) при одржувањето на софтвер  
A) Немасна интервенција  
B) Чести измени  
C) Добивање на тежина  
D) Недостаток на движење
4. Што означува кратенката CASE во софтверското инженерство?  
A) Copy and Steal everything  
B) Софтверски системи наменети да обезбедат автоматизирана поддршка за софтверските процесни активности  
C) Вообичаени елементи за апликациските сервиси  
D) Проценка на усогласеноста на сигурносните аспекти
5. Кои се атрибутите на добар софтвер?  
A) Одржливост на софтвер  
B) Функционалност на софтвер  
C) Развој на софтвер  
D) Одржливост и функционалност на софтвер
6. Кое од овие НЕ е точно?  
A) Софтверското инженерство е дел од науката за компјутери  
B) Софтверско инженерство е дел од поопшта форма на системското инженерство  
C) Компјутерската наука е дел од Софтверско инженерство  
D) Софтверското инженерство се занимава со деловите на развивање и доставување на корисен софтвер
7. Кое од следните е точно?  
A) Генерирачки и костумизирани продукти се видови на софтверски производи  
B) Генерирачки продукти се произведени од организација и се продаваат на отворен пазар  
C) Костумизирани продукти се нарачани од одреден клиент  
D) Сите наведени

8. Што од следното не претставува поделба на CASE алатките во софтверското инженерство?
- A) LOWER - CASE
  - B) UPPER – CASE
  - C) TEST – CASE
  - D) INTEGRATED – CASE
9. Што е заедничко за резервација на хотелски соби, контрола на летање, и јавни библиотеки од аспект на апликација наменета за поддршка на нивното работење?
- A) Сите се клиент/сервер системи
  - B) Сите се системи за алокација на ресурси
  - C) Сите се системи за забава
  - D) Сите се системи за обработка на јазици

## Предавање 2

10. Кои се четирите клучни професионални обврски
- A) Злоупотреба на компјутерите, право на интелектуална сопственост, право на приватност
  - B) Стручност, доверливост, злоупотреба на компјутерите
  - C) Стручност, право на приватност, право на безбедност
  - D) Стручност, злоупотреба на компјутерите, право на интелектуална сопственост
11. Објасни што се подразбира под ПРОИЗВОД ако се референцира на еден од 8те принципи во ACM/IEEE етички код
- A) Продуктот треба да е лесен за користење
  - B) Софтверските инженери обезбедуваат нивните производи и сродните измени да ги исполнуваат сите највисоки можни професионални стандарди.
  - C) Софтверските инженери обезбедуваат нивните производи и сродните измени да го задоволат клиентот
  - D) Дека дизајнираниот продукт треба да биде лесно достапен
12. Идентификувај го ТОЧНОТО дополнување на “Софтверските инженери треба”
- A) Постапува на начин што е во најдобар интерес на неговата експертиза и корист
  - B) Да постапува доследно со јавниот интерес
  - C) Да се осигураат дека нивните производи ги исполнуваат само SRS.
  - D) Сето наведено
13. Софтверските инженери не треба да ги користат своите технички вештини за да ги злоупотребуваат компјутерите на другите луѓе”. Ова е термин што се однесува на:
- A) Неавторизиран пристап до компјутерски материјали
  - B) Неавторизирана модификација на компјутерски материјали
  - C) Десеминација на вируси или друг малicioзен софтвер
  - D) Сето наведено
14. Кои се 8те принципи на Software Engineering Code of Ethics and Professional Practice
- Јавност, Клиент и Работник, Производ, Судање, Управување, Професија, Колеги, SELF
15. ISO стандардот ги гарантира квалитетот, безбедноста и ефикасноста на продуктите, сервисите и системите. Кои од наведените ISO стандарди не се насочени само кон софтвер?
- A) ISO/IEC 15504-5

- B) ISO 9000 family
- C) ISO/IEC 90003:2014
- D) ISO 25010

16. Изберете ги надворешните својства на квалитет

- A) Реискористливост
- B) Скалабилност
- C) Исправност
- D) Робусност
- E) Прилагодливост

### Предавање 3

17. Која од следните НЕ е фундаментална активност кај софтверските процеси во софтверското инженерство?

- A) Еволуција
- B) Верификација
- C) Валидација
- D) Дизајн и имплементација

18. Во рамките на работите кои ги специфицираат, опишете на процеси не специфицираат:

- A) Рољи кои ги отслукуваат оговорностите на лицата инволвирани во процесот
- B) Вештини – потребни за изведување на активност
- C) Редослед на активности
- D) Продукти кои се резултат на процесните активности

19. Кој е излез од секој инкремент во инкременталната методологија на развој на софтвер?

- A) Употреблив систем
- B) Документација на инкремент
- C) Компоненти за тестирање на софтвер
- D) Прототип

20. Спиралниот модел е развиен од страна на

- A) Victor Bisili
- B) Berry Boehm
- C) Bev Littlewood
- D) Roger Pressman

21. Една од рамковните активности е особено прецизно дефинирана кај спиралниот модел, што не е случај кај другите модели.

- A) Поставување на јасни цели
- B) Комуникација
- C) Проценка и справување со ризици
- D) Конструкција

### Предавање 4

22. Што е екстремно програмирање?

- A) Процес на решавање проблеми/задачи што вклучува истражување на стотици можности пред да се избере вистинското решение
- B) Метод на програмирање што вклучува програмирање кое што може да трае со часови
- C) Метод на програмирање што вклучува пребаарување на многу комплицирани алгоритми за решавање на различни проблеми.

D) Множество на практики за развој на софтвер коишто се целат кон креирање на доверлив софтвер со користење на обемен фидбек, ангажирање на девелопери и мали итерации на развој.

23. Што од следново НЕ се однесува на агилноста на софтверски процес?

- A) Се продуцираат само неопходни излезни резултати
- B) Сите набројани
- C) Се елиминира потребата на планирање на проектот и тестирање
- D) Применува стратегија на инкрементална испорака

24. Која од следните е вистинска секвенца на активности кај екстремното тестирање?

- A) Напиши код, напиши тестови, рефакторирај
- B) Напиши тестови, напиши код, рефакторирај
- C) Напиши код, рефакторирај, напиши тестови
- D) Дизајнирај, напиши код, напиши тестови

25. Кој пар од наведените модели(обата) не би можеле да го дадат поакуваниот излез без инволвираност на корисникот за целото времетраење на проектот?

- A) XP и Waterfall
- B) XP и Prototyping
- C) XP и Spiral
- D) Waterfall и Spiral

26. Корисничките барања се изразени како \_\_\_\_\_ во Екстремното програмирање.

- A) Имплементациски задачи
- B) Функционалности
- C) Сценарија/приказни
- D) Никое од наведеното

27. Што од следното се испорачува на крајот на секој спринт во SCRUM?

- A) Скица на дизајнот на корисничкиот интерфејс
- B) Документ со тест случаи за тековниот спринт
- C) Архитектонски дизајн на решението
- D) Инкремент од софтвер кој работи

28. Што од следното НЕ се однесува на агилноста на софтверскиот процес?

- A) Одзивот на барањата за промени е поважно отколку следењето на план
- B) Индивидуите и интеракцијата се поважни од тестирањето
- C) Поважно е да се изгради софтвер кој работи отколку да се изработи обемна документација
- D) Поважно е да се изработи софтвер кој одговара на сегашните барања на клиентот отколку да се грижime за функционалностите кои можеби би биле потребни во иднина

29. Која е главната одговорност на тестер во агилната околина во Scrum?

- A) Креирање на тест сценарија и тест случаи
- B) Креирање на автоматизирани скрипти
- C) Наоѓање на грешки
- D) Во Scrum не постои улога на тестер

30. Колку е вообичаено траење на еден спринт во Scrum?

- A) 5-10 дена

B) 1-2 месеци

C) 2-3 недели

D) 6 месеци

31. Селектирај ја опцијата која одговара на Агилниот манифест.

A) Поединци и интеракции

B) Работен софтвер

C) Соработка со клиенти

D) Сите наведени

32. Агилниот развој на софтвер е базиран на?

A) Инкрементален развој

B) Итеративен развој

C) Линеарен развој

D) Инкрементален и итеративен развој

33. Агилните методи се чини дека најдобро функционираат кога членовите на тимот имаат релативно високо ниво на вештина.

A) True

B) False

34. Што претставува програмирање во пар?

A) Процес на развој на софтвер во кој два развивачи на софтвер работат на иста работна станица

B) Процес на програмирање во кој програмерот ја пишува програмата двапати за секој случај

C) Алгоритам кој пронаоѓа парови на броеви во множеството на цели броеви

D) Начин на поправање на грешки во програма барајќи парови на грешки

35. Кои се предности на инкременталниот развој?

A) Структурата на системот не деградира при внесување нови инкременти

B) Повратната информација за сработеното полесно се добива

C) Процесот кај инкременталниот развој е целосно видлив

D) Прилагодувањето кон променливите барања на клиентите е поефтино

36. Агилните методи се чини дека најдобро функционираат кога членовите на тимот имаат релативно високо ниво на вештина.

A) True

B) False

## Предавање 5

37. Што е разликата помеѓу корисничките и системските барања?

A) Корисничките барања се многу кратки и даваат опис на главната цел на продуктот додека системските барања го опишуваат додавањето на техничките детали.

B) Корисничките барања се јасни изрази додека системските барања се апстрактни изрази или математичко функционални

38. Кои се засегнатите страни кои што пристапуваат со кориснички и системски барања?

- A) Contractor managers
- B) Systems architects**
- C) Client managers
- D) System end-users**

39. Одредете дали следните барања се функционални или нефункционални?

- Барања на интероперабилност – нефункционални
- Административни – функционални
- Барања на околина – нефункционални
- Безбедносни барања – нефункционални

40. Кои се учесници во тимот кои истовремено имаат пристап и на кориснички и на системски барања?

- A) Client managers
- B) Systems architects**
- C) Contractor managers
- D) System's end users**

#### Предавање 6

41. \_\_\_\_\_ го спојува јазот помеѓу системското ниво и дизајнот на софтверот.

- A) Analysis model**
- B) Requirement engineering
- C) Implementation
- D) Testing

42. Моделирање на однесување треба да претставува:

- A) Како системот реагира на внатрешни настани
- B) Како системот реагира на интеракција со корисникот
- C) Како системот реагира на тестирање
- D) Како системот реагира на надворешни настани**

43. Како се дефинира принципот на поделба во анализа и моделирање?

- A) Обединување на под-проблемите во еден проблем се додека истиот не е решен
- B) Опис на оригиналниот проблем на под-проблеми се додека секој од под-проблемите не е едноставен да се разбере
- C) Поделба на оригиналниот проблем на под-проблеми се додека секој од под-проблемите не е решен
- D) Поделба на оригиналниот проблем на под-проблеми се додека секој од под-проблемите не е едноставен да се разбере.**

44. Моделите на проток(тек) претставуваат:

- A) Како податочните текови протекуваат во системот и како истите се трансформираат**
- B) Како контролните податочните протекуваат во системот
- C) Тригерираните настани што се пропуштаат низ системот
- D) Кој е редоследот на корисничките акции во едно корисничко сценарио

45. Колаборационите дијаграми се дел од:

- A) Модели на однесување
- B) Класни дијаграми**
- C) Податочни модели
- D) Модели базирани на сценарио

46. Која е кардиналност на врската помеѓу ентитетите студент и број на индекс на еден факултет?

- A) 1:M
- B) M:1
- C) M:M
- D) 1:1**

### Предавање 7

47. Репрезентацијата на архитектурата може да биде овозможувач на комуникација помеѓу засегнатите страни во проектот

- A) TRUE**
- B) FALSE

48. \_\_\_\_\_ поглед го прикажува системскиот хардвер и како софтверските компоненти се дистрибуирани низ процесорите во системот.

- A) Логички
- B) Сите набројани
- C) Физички**
- D) Процесен

49. Архитектура со централен репозиториум обично се користи:

- A) За да се овозможи директна интеракција помеѓу компонентите на системот
- B) За дистрибуција на податоците за да нема единствена точка на испад
- C) За ефикасно решавање на проблемот за споделување на голема количина на податоци помеѓу повеќе одвоени компоненти на системот**
- D) За да овозможи неограничен и несинхронизиран пристап до податоците од страна на сите компоненти истовремено

50. Најдобрата репрезентација на системската архитектура е прототип на софтвер кој работи

- A) TRUE
- B) FALSE

51. Што од следното е точно за слоевата архитектура

- A) Секој слој обезбедува услуги на слојот над него и ги користи услугите на сервисите обезбедени од слојот под него**
- B) Дозволено е секој слој да зависи од слојот над него ако истиот е присутен и коректен
- C) Даден слој може да ги повикува слоевите под и над него се додека ги користи
- D) Секој слој може да ги користи услугите од сите слоеви над него

52. Кој поглед во архитектонскиот дизајн ги прикажува клучните апстракции во системот како објекти или класи од објекти?

- A) Развоен

- B) Процесен
- C) Логички
- D) Физички

#### Предавање 8

53. Ако квалификуваната компонента ги исполнува сите критериуми за интеграција, која техника на адаптација ќе ја користите
- A) White – box wrapping
  - B) No adaptation
  - C) Black – box wrapping
  - D) Gray – box wrapping
54. Класите што се спакувани заедно треба да ја адресираат истата функционална област.
- A) The Common Closure Principle
  - B) The Liskov Substitution Principle
  - C) The interface Sagregation Principle
  - D) The Release Reuse Equivalency Principle
55. Ако успееме да направиме проширување на функционалностите на една компонента без да ја промениме постоечката имплементација, тогаш сме го следеле \_\_\_\_\_ во процесот на дизајн.
- A) The Open Closed Principle
  - B) The Liskov Substitution Principle
  - C) The Interface Sagregation Principle
  - D) The Comon Closure Principe
56. Системот ќе продолжи да работи нормално, ако некои од класите во компонентата се променети со нова изведена класа доколку го следиме:
- A) The Open Closed Principle
  - B) The Liskov Substitution Principle
  - C) The Interface Sagregation Principle
  - D) The Comon Closure Principe
57. Кои активности се важни за инженерството на доменот
- A) Анализа
  - B) Конструкција
  - C) Тестирање
  - D) Имплементација
58. Квалификацијата на компоненти гарантира:
- A) Дека компонента-кандидат ќе ги исполни сите побарани критериуми за перформанси на апликацијата
  - B) Дека компонента-кандидат ќе ги исполни сите побарани критериуми за Апликацијата
  - C) Дека компонента-кандидат ќе ги изврши функциите коишто се потребни за комуникација со надворешен систем.
  - D) Дека компонента - кандидат ќе ги изврши функциите коишто се потребни за апликацијата
59. Кои критериуми треба да се земат во предвид за квалификацијата на компоненти?



- A) Можност за тестирање
- B) Побарување за складирање
- C) Справување со исклучоци
- D) Програмерскиот интерфејс на апликацијата

## Предавање 9

60. Кое од следните не е проблем при дизајн на кориснички интерфејс?
- A) Response time
  - B) Error handling
  - C) Internationalization
  - D) Memory usage
61. Кое од следните НЕ Е златно правило за дизајн на кориснички интерфејси?
- A) Креирање на конзистентен интерфес
  - B) Давање контрола на корисникот
  - C) Намалување на оптовареноста на меморија на корисникот
  - D) Креирање на интерфејс едноставен за користење
62. Информациите на корисничкиот интерфејс треба да бидат презентирани \_\_\_\_\_.
- A) Прогресивно
  - B) Комплетно
  - C) Офтомена целост
  - D) Агресивно
63. Кои од следните се дел од процесот на дизајн на кориснички интерфејси?
- A) Валидација на интерфејс
  - B) Дизајн на интерфејс
  - C) Сите други наведени
  - D) Документација на интерфејс
64. Анализата на интерфејсот треба да ни даде информации за:
- A) Задачите кои корисниците ќе ги изведуваат во системот
  - B) Комплексноста на имплементацијата на интерфејсот
  - C) Луѓето што ќе го користат системот
  - D) Библиотека што ќе се користи за претставување на информациите на front end

## Предавање 10

65. Што е точно за шаблонот на дизајн на софтвер Набљудувач?
- A) Го обвиткува објектот и му дава поинаков интерфејс
  - B) Го поедноставува интерфејсот на множеството класи
  - C) Овозможува да се известат објектите во случај на промена на состојба
  - D) Ги енкапсулира однесувањата базирани на состојби и користи делегирање за да премине од едно на друго однесување

## Предавање 11 и 12

66. Изберето го точниот одговор:

- A) Тестови што гарантираат дека сите независни контролни патеки во модулот ќе бидат извршени барем еднаш се карактеристични за: **white box testing**
- B) Тестирање во кое внатрешната структура на компонентата или системот не мора да биде позната е: **black box testing**
- C) Тестовите кои ги извршуваат сите циклуси земајќи ги во предвид самите гранични вредности и во рамките на предвидениот опсег се карактеристични за: **black box testing**
- D) Тестовите кои ги извршуваат сите логички одлуки на нивната вистинита и невивистинита страна се карактеристични за: **white-box testing**
- E) Се обидува да најде грешки во интерфејсот: **Black-box testing**
- F) Го дефинира бројот на независни патеки во програмата и ја дава горната граница на тестови што треба да се изведат за да се обезбеди секоја наредба да биде извршена барем еднаш: **цикломатска комплексност**.
- G) Квантитативна мера на логичка комплексност на програма: **цикломатска комплексност**.

67. Изберете го точниот одговор:

- A) Обезбедува дека софтверскиот производ се прави според барањата: **валидација**
- B) Обезбедува дека софтверскиот производ навистина ги задоволува потребите на клиентот: **Верификација**
- C) Се тестираат индивидуални компоненти на софтверот: **Unit testing**
- D) Повторно извршување на подмножество на веќе спроведени тестови со цел да се потврди дека промените во кодот не предизвикале несакани ефекти: **Перспективно тестирање**
- E) Валидира комплетен и целосно интегриран софтверски производ: **Системско тестирање**.

68. Тестерот или тој што го развива софтверот го проверува кодот за одредено поле на веб-страницата, ги одредува сите легални (валидни, невалидни) и нелегални влезови и ги верификува резултатите во однос на очекуваните исходи што исто така се одредува со проверка на кодот. Ова е пример за:

- A) **Интеграциско тестирање**
- B) Тестирање базирано на црна кутија
- C) Тестирање базирано на бела кутија
- D) Дебагирање

69. Со тестирањето на софтвер може да се открие и присуство и отсуство на грешки.

- TRUE
- **FALSE**

70. Алфа тестирање се изведува од страна на:

- A) Тој што го развива софтверот и корисникот
- B) **Тој што го развива софтверот**
- C) Ниту еден од наведените
- D) Корисникот

71. \_\_\_\_\_ го изведуваат тестирањето на ниво на единица (unit testing)

- A) Независна група
- B) Клиентите
- C) Бизнис аналитичарите
- D) **Тие што го развиваат софтверот**

72. Анализа на гранични вредности спаѓа во:

- A) Ниту еден од останатите одговори

**B) Тестирање базирано на црна кутија**

C) Тестирање базирано на бела кутија и тестирање базирано на црна кутија

D) Тестирање базирано на бела кутија

73. Техниката на адаптирање Black – box wrapping изведува?

A) Промена на API-та на компонентите согласно нашите промени

**B) Пост-процесирање на податоци**

**C) Пред-процесирање на податоци**

D) Промена на внатрешниот изворен код на компонентата согласно нашите потреби

74. Што од наведеното е валидно ниво на тестирање?

A) Дебагирање

**B) Unit testing**

**C) System testing**

**D) Интеграциско тестирање**

75. Која техника за адаптација на компоненти дава целосна контрола на интерниот дизајн и изворниод код?

A) Black-box wrapping

B) Glass-box wrapping

**C) White-box wrapping**

D) Gray-box wrapping

76. Изберете ги чекорите потребни за процесот за интеграција кај интеграциското тестирање базирано на стратегија одгоре надолу во правилен редослед(TOP-DOWN)

A) Главниот контролен модул се користи како драјвер за тестирање и сите компоненти директно потредени на главниот контролен модул се заменуваат со стабови

B) Во зависност од избраниот пристап за интеграција(Длабочина или ширина), подредените стабови се заменуваат еден по еден со вистинските компоненти.

C) Со интегрирање на секоја следна компонента, се спроведуваат тестови

D) Со завршување на секое множество тестови, друг стаб се заменува со реална компонента

E) Потоа може да се спроведе регресиско тестирање со цел да се обезбеди дека не се воведени нови грешки

77. Изберете ги чекорите потребни за процесот за интеграција кај интеграциското тестирање базирано на стратегија правилен редослед(Bottom-up)

A) Кластери се формираат со спојување или комбинирање на модули/компоненти од најниско ниво.

Овие кластери се познати како (builds) и се одговорни да извршуваат одредена секундарна функција на софтверот

B) Се пишува контролна програма за тестирање наречени драјвери или модули на високо ниво.

C) Потоа се тестира кластерот

D) Се отстрануваат драјверите, а кластерите се комбинираат во насока нагоре во програмерската структура

78. Што е Sandwich testing?

- Комбинација на top down и bottom up методите за тестирање и ги користи добрите страни и на двете тестирања

79. Кои се чекорите на Sandwich testing?

A) Тестирање на кориснички интерфејс изолациските користејки stubs.

- В) Тестирај ги најниските нивоа (најниските функции) користејќи драјвери
- С) Кока целиот систем е интегриран само main средното ниво останува да се тестира последно.

80. Оперативност – релативно неколку грешки ќе го блокираат извршувањето на тестовите  
Набљудување – внатрешните грешки автоматски се откриваат и пријавуваат  
Контролирање – сите можни излези се генерираат преку одредена комбинација на влез, а I/O форматите се конзистентни и структурирани  
Разградување – Изолација на проблеми и независните модули може да се тестираат независно  
Едноставност –  
Стабилност – софтверот добро се обновува после неуспеси, промените се ретки и контролирани  
Разбирливост – тех док веднаш достапна, зависностите помеѓу надворешни внатрешни компоненти се добро разбрани, промените се соопштуваат на тестерите

### Предавање 13

81. Кои од следните припаѓаат на софтверската еволуција
- А) Прибирање на барања за промена, кое директно влијае врз промената на грешките
  - В) Планирање на новото издание кое е во интеграција со кодирањето на промените
  - С) Промена на анализата како последица на планирање на новото издание
  - Д) Прибирање на барања за промена и анализа на нивното влијание
82. Кои се клучните проблеми кога при еволуцијата се применуваат агилните методи?
- А) Тимовите што ја применуваат агилната методологија создаваат детални спецификации
  - В) Тимовите што ја применуваат агилната методологија не се запознаени со автоматизирано регресионо тестирање
  - С) Тимовите за развој и еволуција даваат различна предност на агилните методологии
  - Д) На тимовите што ја применуваат агилната методологија им е неопходен плански пристап
  - Е) Тимовите за развој имаат различно искуство со агилните методологии
83. Одредете го типот на одржување во следниот случај: Корисниците можат да го проверат статусот на нивниот авионски лет онлајн, но постоечкиот софтвер често погрешно го прикажува времето на летот. Грешките се корегираат во согласност со корисничките барања.
- А) Откривање на недостатоците: Корективно
  - В) Отстранување на недостатоците: Корективно
  - С) Промени што произлегуваат од оперативниот систем, хардверот и системот за управување со базите: Адаптивно
  - Д) Промени што потекнуваат од барањата на корисниците: Перфективни
  - Е) Промени што се прават за да се зголеми одржливоста на производот: Превентивни

1. Како не може да се пресмета циклусната сложеност(cyclomatic complexity)?

- a. Со пребарување на сите ребра и јазли во графот на протокот(flow graph)
- b. Со пребарување на регионите во графот на протокот(flow graph)
- c. Со пребарување на сите предикати во графот на протокот(flow graph)
- d. Со пребарување на сите разгранувања во графот на протокот(flow graph)**

2. Кога софтверскиот систем е употреблив и притоа во него не се додаваат нови функционалности, тогаш производот е во фаза на:

- a. тестирање
- b. постепено укинување
- c. сервисирање**
- d. еволуција

3. Доколку сте ја создале графовската матрица(graph matrix), тогаш врз основа на оваа матрица, што НЕ може да направите:

- a. да ја процените контролната структура на програмата
- b. да ја пресметате матрицата на сврзаност(connection matrix)
- c. да ја пресметате кружната сложеност(cyclomatic complexity)
- d. да ги проверите логичките услови во програмскиот модул**

4. Во која активност на процесот на објектно-ориентираниот дизајн е клучно да се разберат односите меѓу софтверот што се дизајнира и надворешната средина?

- a. дизајн на системската архитектура
- b. идентификација на главните објекти на системот
- c. контекст на системот и интеркациите**
- d. развој на моделите на дизајнот

5.Кога тестирањето се реализира на таков начин што главниот модул се тестира со опушоци(stubs), при што тие се заменуваат еден по еден, тогаш се применува:

a. bottom up integration

b.sandwich testing

**c.top down integration**

d.smoke testing

6.Какво тестирање на јамките правите кога јамките ги разгледувате прво како прости, а потоа како вгнездени?

a.Тестирање на неструктурираните јамки(unstructured loops)

**b.тестирање на вгнездените јамки(nested loops)**

c.тестирање со минимум предуслови(minimum conditions)

d.поврзано тестирање(concatenated loops)

7.Кои се предностите на експлицитно направената архитектура?

a.Архитектурата никогаш не се користи во комуникацијата со засегнатите страни

**b.архитектурата се користи како центар за дискусија на засегнатите страни**

**c. може да се искористи за проценка дали системот ќе може да ги исполни нефункционалните барања**

d.дадената архитектура не може да се употреби кај повеќе системи(пр. Цела колекција системи)

8.Што ги карактеризира компонентите во конвенционалниот поглед?

**a.Компонентата е функционален елемент на програмата**

b.Компонентите се извлекуваат од каталог

c.Компонентата ги вклучува сите атрибути и релации релевантни за нивното реализирање

d. Компонентата содржи множество класи што соработуваат

9. Што значи принципот на замена на Лисков (LSP)?

a. Сите подкласи на основната класа(суперкласа) можат да бидат меѓусебно заменети без да се промени однесувањето на програмата.

b.Сите подкласи на основната класа(суперкласа) можат да бидат заменети со други подкласи без да се промени однесувањето на програмата

c.Сите основни класи(суперкласа) можат да бидат заменети со други суперкласи без да се промени однесувањето на програмата

**d.Сите подкласи на основната класа(суперкласа) можат да бидат заменети со нивната суперкласа без да се промени однесувањето на програмата.**

10. Кога одново се извршува некое подмножество тестови кои веќе биле извршени, со цел да се утврди дали промените не предизвикале споредни несакани ефекти, тогаш се применува?

a. bottom-up integration

b.smoke testing

**c.regression testing**

d.top down integration

11. Одредувањето на definition use (DU) веригите за секоја од променливите во програмата е составен дел на:

a.пресметувањето на условните оператори

b.пресметувањето на циклусната сложеност(cyclomatic complexity) на програмата

c пресметувањето на логичките услови во програмата.

**d.тестирањето на протокот на податоците**

12. Откривањето на профилот на корисникот, неговите искуства, способности и стравови се дел од:

a. Анализата на задачите и моделирањето

**b. анализата на корисникот**

c. анализата на прикажаната содржина

d. анализата на интерфејсот

13. Во која фаза софтверските инженери го вршат тестирањето?

a. Секогаш кога ја користат (употребуваат) програмата

**b. Во тек на интеграцијата**

c. Штом програмата ќе може да извршува

d. Во тек на кодирањето

14. Кога софтверскиот производ работи и притоа се сугерираат и додаваат нови барања, тогаш производот е во фаза на:

a. постепено укинување

b. одржување

**c. еволуција**

d. сервисирање

15. Во која фаза специјалистите за тестирање го вршат тестирањето?

a. во најраните фази на тестирањето

b. секогаш кога се извршува програмата

**c. во моментот кога тестирањето е во тек**

d. во тек на кодирањето

16. Што не претставуваат шаблоните на архитектурата?

a. Стилизиран опис на добри дизајнерски практики

b. Стилизиран опис на веќе применети и тествани практики



**c. Средства за прикажување, споделување и реискористување на кодот**

d. Средства за прикажување, соделување и реискористување на знаењето

17. Што означува принципот на сегрегација на интерфејсот (ISP)?

a. Подобрo е да се направи само еден специфичен интерфејс во рамките на секој интерфејс за општа намена.

**b. Подобрo е да се направат повеќе специфични интерфејси отколку само еден интерфејс за општа намена.**

c. Подобрo е да се направи само еден специфичен интерфејс кој ќе биде интерфејс за општа намена.

d. Подобрo е да се направи само еден специфичен интерфејс отколку повеќе интерфејси за општа намена.

18. Откривањето што треба да направи корисникот во специфични услови се дел од:

a. анализа на корисникот

b. анализа на интерфејсот

**c. анализа на задачите и моделирањето**

d. анализа на прикажаната содржина

19. Кога се прави дизајн на компонентите од аспект на проблемот што се решава (process-related view) карактеристично е дека

a. Компонентите содржат множество калси што соработуваат

b. Компонентите содржат процесиращка логика

c. Компонентите се функционални елементи на програмата

**d. Се користат веќе направени софтверски компоненти и шаблони**

20.Што означува принципот за заедничко реискористување (CRP)

- a. Класите што истовремено се реискористуваат не мора да се групираат заедно
- b. Класите што нема истовремено да се реискористуваат не треба да се групираат заедно.**
- c. Класите што нема истовремено да се реискористуваат може да се групираат заедно
- d. Класите што истовремено се реискористуваат никогаш не се групираат заедно.

21.Откривањето дали корисникот може да ја прилагоди локацијата на екранот на кој ќе се појавува дадената содржина е дел од :

- a. Анализата на прикажаната содржина**
- b.Анализата на интерфејсот
- c.Анализата на задачите и моделирањето
- d.Анализата на корисникот

22. За која активност на процесот на објектно-ориентираниот дизајн е клучно да се одредат објектите и класите и да се воспостават релациите меѓу нив?

- a. Контекст на системот и интеракциите
- b. Дизајн на системската архитектура
- c.Развој на моделите на дизајнот**
- d.Идентификација на главните објекти на системот

23. Што не претставува клиент-сервер архитектура?

- a.Функционални трансформации кои ги обработуваат влезовите за да произведат излези**
- b. Дистрибуиран модел на системот
- c. Мрежа што им овозможува на клиентите пристап до серверот
- d. Збир самостојни сервери кои обезбедуваат специфични услуги

Кое прашање повеќе не се однесува на современиот софтверски инженер?

- Зошто компјутерскиот хардвер чини толку многу?

Софтверот е производ и може да се произведува со користење на истите технологии што се користат за други инженерски артефакти.

- False

Софтверот се влошува наместо да се истроши затоа што

- Повеќе барања за промена воведуваат грешки во интеракцијата со компонентите

WebApps се мешавина од објавување печати и развој на софтвер, што го прави нивниот развој надвор од доменот на практиката на софтверско инженерство.

- False

Која од предметите наведени подолу не е еден од слоевите на софтверското инженерство?

- Производство

Активностите за чадор за софтверско инженерство се применуваат само во почетните фази на проектите за развој на софтвер.

- False

Кои од овие се 5-те општи активности на рамки за инженерство на софтвер?

- комуникација, планирање, моделирање, изградба, распоредување

Планирањето напред за повторна употреба на софтверот ги намалува трошоците и ја зголемува вредноста на системите во кои тие се вградени.

- True

Суштината на практиката на софтверско инженерство може да се опише како да се разбере проблемот, да се испланира решение, да се спроведе планот и да се испита резултатот за точност.

- True

Во агилните процесни модели, единствениот производ што може да се испорача е работната програма.

- False

Најмногу проекти за развој на софтвер се иницирани за да се обиде да задоволи деловна потреба.

- True

Општо, софтверот успева само ако неговото однесување е во согласност со целите на неговите дизајнери.

- False

Кое од следниве се признати типови на проток на процес?

- Привремен проток на процеси
- Проток на линеарен процес

Софтверските процеси можат да бидат конструирани од постојните модели на софтвер за најдобро да се задоволат потребите на софтверскиот проект.

- True

Кои од овие се стандарди за проценка на процесите на софтвер?

- Зачин (SPICE)
- ISO 9001

Водопадниот модел (waterfall model) на развој на софтвер е

- Разумен пристап кога барањата се добро дефинирани.

Постепен модел (incremental model) на развој на софтвер е

- Дobar пристап кога брзо се бара производ за основно дејство.

Модели на процеси на софтверски процеси

- Дали се итеративни по природа
- Лесно може да се прилагоди на промените во барањата на производот
- Обично не произведувајте системи за фрлање отпад
- **Сите горенаведени**

Моделот за прототипирање на развој на софтвер е

- Корисен пристап кога клиентот не може јасно да ги дефинира барањата.

Спиралниот модел (spiral model) на развој на софтвер

- Вклучува евалуација на ризици на проектот за време на секоја повторување

Истовремениот модел за развој е

- Друго име за истовремено инженерство.
- Дефинира настани што предизвикуваат транзиции во областа на инженерската активност.

Развојниот модел заснован на компонентата е

- Зависност од предметни технологии за поддршка.

Официјалниот модел на методи за развој на софтвер го прави користењето на математички методи

- Дефинирајте ги спецификациите за компјутерски системи
- Развијте системи засновани без компјутерски дефекти
- Потврдете ја исправноста на системите базирани на компјутер
- **Сите горенаведени**

Кое од овие не е едно од фазните имиња дефинирани со моделот Унифициран процес за развој на софтвер?

- Фаза на валидација

Која од овие не е карактеристика на процесот на личен софтвер?

- Вежбачот бара внимателен надзор од страна на раководителот на проектот

Кои од овие се целите на тимскиот софтверски процес?

- Дозволете подобро управување со времето од високо обучени професионалци
- Изградете тимови за само-режија на софтвер

Алатките за технологија на процеси им овозможуваат на софтверските организации да ги компресираат распоредите прескокнувајќи ги неважните активности.

- False

Дали е општо е прифатено дека не може да има слаби софтверски процеси и да се создадат крајни производи со висок квалитет.

- True

Агилноста не е ништо повеќе од можноста проектниот тим да реагира брзо за промена.

- False

Кое од следниве не е неопходно да се примени агилност на софтверски процес?

- Елиминирајте ја употребата на планирање и тестирање на проекти

Како создавате агилни процеси за управување со непредвидливоста?

- Зголемувањата на софтверот мора да бидат доставени во кратки периоди
- Софтверските процеси мора постепено да се прилагодуваат на промените

Во агилни софтверски процеси најголеми приоритети е да се задоволи клиентот преку рано и континуирано доставување на вреден софтвер.

- True

Која од следниве карактеристики треба да постои меѓу членовите на агилен тим за софтвер?

- Компетентност

- Способност за донесување одлуки
- Взаемна доверба и почит
- **Сите горенаведени**

Во агилен развој поважно е да се изгради софтвер кој одговара на потребите на клиентите денес отколку да се грижите за карактеристиките што може да бидат потребни во иднина.

- True

Кои се четирите рамковни активности што се наоѓаат во моделот на процеси на екстремно програмирање (XP)?

- планирање, дизајн, кодирање, тестирање

Сите модели на агилни процеси се во согласност со поголем или помал степен со принципите наведени во „Манифестот за развој на агилниот софтвер“.

- True

Кои се трите рамковни активности за процесниот модел на Адаптивен развој на софтвер (АСД)?

- шпекулации, соработка, учење

Кое не е едно од клучните прашања на кое одговара секој член на тимот на секој дневен состанок со „Скрам“?

- Која е причината за проблемите со кои се соочувате?

Методот за развој на динамички системи (ДСДМ) сугерира филозофија која се заснова на принципот Парето (80% од апликацијата може да се испорача за 20% од времето потребно за изградба на целосна апликација).

- True

Во развојот управувано со одлики (FDD) карактеристика ценета од клиент е функција чија вредност е клиент, која може да се испорача за две недели или помалку.

- False

Агилното моделирање (AM) им дава насоки на лекар за време на која од овие софтверски задачи?

- Анализа
- Дизајн

Agile Unified Proces ги користи класичните UP-фазни активности (почеток, елаборација, градба, транзиција) за да му помогне на тимот да го визуелизира целокупниот проток на процеси.

- True

Принципите за софтверско инженерство имаат околу три години полуживот.

- False

Кое од следниве не е еден од основните принципи на практиката на софтверско инженерство?

- Принцип Парето (20% од кој било производ бара 80% од напорите)

Секоја комуникациска активност треба да има олеснувач за да се осигури дека клиентот не смее да доминира во постапките.

- False

Агилниот поглед на итеративната комуникација и соработка на клиентите е применлив за сите практики за софтверско инженерство.

- True

Една од причините за вклучување на сите луѓе во софтверот за планирање е да

- Ги натерате сите членови на тимот да се „регистрираат“ на планот

Плановите на проектот не треба да се менуваат откако ќе бидат усвоени од тим.

- False

Моделите на побарувања го прикажуваат софтверот во кој три домени?

- информации, функција, однесување

Дизајнот модел (design model) треба да биде во трага по моделот на побарувања?

- True

Тимови кои користат практични софтверски практики обично не создаваат модели.

- False

Кое од следниве не е еден од принципите на добро кодирање?

- Напишете код за само-документирање, а не за програмска документација

Успешен тест јас што откривам барем една не-откриена грешка.

- True

Кое од следниве е валидна причина за прибирање повратни информации од клиентите во врска со испорачаниот софтвер?

- Им дозволува на развивачите да направат измени во дадениот прилог
- Распоредот на испорака може да се ревидира за да ги одрази промените
- Програмерите можат да идентификуваат промени за да се вклучат во следниот прирачник
- **Сите горенаведени**

Барања инженеринг е генерички процес кој не варира од еден софтверски проект до друг.

- True

За време на започнувањето на проектот, треба да се утврди намерата на задачите

- основно разбирање на проблемот
- природата на потребното решение
- луѓе кои сакаат решение

Три работи што ја отежнуваат елиминацијата на барањата се проблеми на

- Обемот
- Разбирање
- Нестабилноста

Засегната страна е секој кој ќе го купи комплетниот софтверски систем во развој.

- False

Релативно е вообичаено за различни клиенти да предложат спротивставени барања, секој тврди дека неговата или нејзината верзија е вистинската.

- True

Кое од следниве не е едно од прашањата без контекст што би се користеле за време на започнувањето на проектот?

- Кој ќе плати за работата?

Во услови на соработка се собира фасилитаторот

- го контролира состанокот

Кое од следниве не е една од барањата за класификација што се користи во распоредувањето на функциите за квалитет (QFD)?

- Задолжително

Работните производи произведени за време на избирање на барања ќе варираат во зависност од

- големината на производот што се гради

Програмерите и клиентите создаваат случаи на употреба за да му помогнат на софтверскиот тим да разбере како различните класи на крајните корисници ќе ги користат функциите.

- True

Актерите за употреба се секогаш луѓе, никогаш системски уреди.

- False

Резултатот од инженерската задача на барања е модел за анализа што дефинира кој од следниве области (а) на проблемот?



- информации
- функционален
- однесувањето
- **Сите горенаведени**

Моделите на анализа ја олеснуваат трансформацијата на моделот за анализа во модел на дизајнирање предлагајќи сигурни решенија за вообичаени проблеми.

- True

Во понудената преговори, потребите на клиентот се исполнети иако потребата на инвеститорот не може да биде.

- False

Во валидацијата на барањата, моделот на барања се разгледува за да се обезбеди техничка изводливост.

- False

Кој од овие не е елемент на модел на барања?

- Елементи на податоци

Кое од следниве не е цел за градење на модел на барања?

- развијте скратено решение за проблемот

Објектно-ориентираната анализа на домен се занимава со идентификација и спецификација на можностите за еднократно користење во рамките на доменот на апликација.

- True

Во структурните анализи моделите се фокусираат на структурата на часовите дефинирани за систем заедно со нивните интеракции.

- False

Создавање и рафинирање на случаи на употреба ако важен дел од моделирање базирано на сценарио.

- True

Важно е да се земат предвид алтернативни актерски интеракции при креирање на случај со претходна употреба.

- False

Истражување на мозокот е една техника што може да се користи за да се добие целосен пакет исклучоци од случајот.

- True

Во многу случаи, нема потреба да се креира графичко претставување на сценарио за употреба.

- True

Дијаграмите за активност на UML се корисни за претставување на кои елементи на моделот за анализа?

- Елементи засновани на сценарија

Еден или повеќе атрибути на податочен предмет мора да бидат дефинирани како клуч за да се овозможи локацијата на пример на податочниот предмет.

- True

Дијаграм на врските со ентитетот

- ги прикажува односите помеѓу предметите на податоците

Кое од следниве треба да се смета за објекти на кандидат во простор за проблеми?

- настани
- луѓе
- структури
- **Сите горенаведени**

Атрибути се избираат за еден предмет со испитување на изјавата за проблемот и идентификување на субјектите за кои се чини дека се поврзани.

- False

Која од следниве не е една од широките категории што се користат за класификација на операциите?

- Трансформатори

Која од следниве ставки не се појавува на CRC картичка?

- сигурна класа

Класичните одговорности се дефинирани со

- и неговите атрибути и операциите

Пакет за анализа вклучува категоризација на елементите на моделот за анализа во корисни групации.

- True

Дијаграм за проток на податоци (data flow)

- прикажува функции што го трансформираат протокот на податоци

- означува како податоците се трансформираат од системот

Контролни дијаграми на проток се

- потребни за моделирање на системи управувани од настани

Контролната спецификација го претставува однесувањето на системот користејќи UML секвенца и државни дијаграми.

- True

Дијаграмот за проток на податоци мора да се надополни со мини-спецификации што можат да послужат како водич за дизајнирање на софтверската компонента што ќе го спроведе процесот.

- True

За целите на моделирање на однесување, настан се случува секогаш кога

- системот актер разменува информации.

За целите на моделирање на однесување е секој

- набљудувачки начин на однесување.

Дијаграм на државна транзиција

- означува реакции на системот на надворешни настани

Дијаграмот на низата UML го покажува редоследот по кој се обработуваат системските настани.

- False

Откриени се моделите на анализа, тие не се експлицитно создадени.

- True

Не е можно да се оправда времето потребно за анализа на барањата на WebApp.

- False

Која не е една од аналитичките активности што се користи за создавање целосен модел на анализа?

- Анализа на пазарот

Предмети со содржина се извлечени од случаи на употреба со испитување на описот на сценарио за директни или индиректни референци за содржината.

- True

Кои се елементите на моделот на интеракција преку WebApp?

- случаи на употреба, дијаграми за низа, дијаграми на состојба, прототип на интерфејс

UML-дијаграмите за активност може да се користат за да ја претставуваат функционалноста на набудувачот на корисникот испорачана од WebApp, како и операциите содржани во секоја класа на анализа.

- True

Анализата на конфигурацијата се фокусира на архитектурата на околината на прелистување на корисникот.

- False

Кои од следниве се области на загриженост во моделот на дизајнирање?

- архитектура
- податоци
- интерфејси

Важноста на дизајнот на софтверот може да се сумира со еден збор

- Квалитет

Кои од овие се карактеристики на добар дизајн?

- ги спроведува сите барања во моделот на анализа
- обезбедува целосна слика за софтверот

Кое од следниве не е карактеристика заедничка за сите методи на дизајнирање?

- управување со конфигурацијата

Кои видови на апстракција се користат при дизајнирање на софтвер?

- контрола
- податоци
- процедурална

Кое од следниве може да се искористи за да претставува архитектонски дизајн на парче софтвер?

- Динамички модели
- Функционални модели
- Структурни модели
- **Сите горенаведени**

Моделите за дизајнирање не се применливи за дизајнот на софтвер ориентиран кон предмет?

- False

Бидејќи модуларноста е важна проектна цел, не е можно да имате премногу модули во предложениот дизајн.

- False

Сокривањето на информации го олеснува одржувањето на програмата со прикривање на податоците и процедурата од непогодените делови на програмата.

- True

Кохезијата е квалитативна индикација за степенот кој модул

- се фокусира на само една работа.

Спојувањето е квалитативна индикација за степенот кој модул

- е поврзан со други модули и надворешниот свет.

При користење на структурни методологии за дизајн, процесот на чекор-рафинирање е непотребен.

- False

Софтверските дизајни се преработени за да се овозможи создавање на софтвер кој е полесен за интегрирање, полесен за тестирање и полесен за одржување.

- True

Кое од следниве не е еден од петте типови на класи на дизајн

- Часови на ентитет

Кои елементи за модерен дизајн се користат за да се прикаже модел на информации претставен од гледиштето на корисникот?

- Елементи за дизајн на податоци

Кој дизајн е еквивалентен на подот план на една кука?

- Архитектонски дизајн

Кој модел на дизајн е еквивалентен на деталните цртежи на точките за пристап и надворешните комунални услуги за кука?

- Дизајн на интерфејс

Кој модел на дизајн е еквивалентен на збир на детални цртежи за секоја просторија во кука?

- Дизајн на ниво на компоненти

Елементите за дизајн на распоредување ја одредуваат наредбата за градење на компонентите на софтверот.

- False

Најдоброто претставување на системската архитектура е оперативен прототип на софтвер.

- False

Архитектонските репрезентации можат да бидат алатка за комуникација помеѓу заинтересираните страни во проектот.

- True

Архитектонскиот опис честопати се документира со помош на образец за архитектура.

- False

Архитектонски жанр честопати ќе го диктира архитектонскиот пристап што може да се користи за да се изгради структурата.

- True

Архитектонски стил опфаќа кој од следниве елементи?

- Ограничувања
- збир на компоненти
- семантички модели

За да се одреди архитектонскиот стил или комбинацијата на стилови што најдобро одговара на предложениот систем, инженерството на барања се користи за откривање

- карактеристики и ограничувања

Пред да може да се избере архитектонски модел за употреба во одреден систем, мора да има имплементација на код за да се олесни повторната употреба.

- False

Критериумите што се користат за проценка на квалитетот на архитектонскиот дизајн треба да се засноваат на систем

- Контрола
- Податоци

За време на процесот на моделирање на системот во контекст, системите што комуницираат со целиот систем се претставени како

- Системи на ниво на врсници
- Подредени системи
- Супериорни системи

Откако ќе бидат избрани, архетипите секогаш треба да се рафинираат понатаму како што се одвива архитектонскиот дизајн.

- True

Кое од следниве не е пример на компоненти на инфраструктурата што може да треба да бидат интегрирани во архитектурата на софтверот?

- Компоненти на интерфејс

Во методот на размена на архитектура, архитектонскиот стил треба да се опише со употреба на

- преглед на проток на податоци
- преглед на модул
- преглед на процеси

Квантитативни методи за проценка на квалитетот на предложените архитектонски дизајни се лесно достапни.

- False

Корисна техника за проценка на целокупната сложеност на предложената архитектура е да се погледне во компонентата

- зависности од проток
- споделување на зависности

Кога целокупниот проток во сегмент од дијаграмот за проток на податоци е во голема мера секвенцијален и се следат линиски патеки е присутна.

- проток на трансформација

Кога една ставка што поттикнува друг проток на податоци по една од многуте патеки го карактеризира протокот на информации во сегмент од дијаграмот за проток на податоци е присутна.

- проток на трансакција

Во најопшта смисла, компонента е модуларен градежен блок за компјутерски софтвер.

- True

Во контекст на објектно-ориентираното софтверско инженерство, компонентата содржи

- збир на часови за соработка

Во традиционалните модули за софтверско инженерство мора да послужи во која од следниве улоги?

- Контролна компонента
- Инфраструктурна компонента
- Домен на проблемот компонента

- **Сите горенаведени**

Инженерите на софтвер секогаш треба да ги толкуваат компонентите од нула со цел целосно да ги исполнат очекувањата на клиентите.

- False

Кој од следниве не е еден од четирите принципи што се користат за водење дизајн на ниво на компоненти?

- Принцип на парсимоносна сложеност

Употребата на стереотипи може да помогне да се идентификува природата на компонентите на детално ниво на дизајнирање.

- True

Класите и компонентите што покажуваат функционална, слој или комуникациска кохезија се релативно лесни за спроведување, тестирање и одржување.

- True

Спојувањето на софтвер е знак на слаб архитектонски дизајн и секогаш може да се избегне во секој систем.

- False

Во составувањето на дизајнот на компонентите бара кој од следниве елементи детално да се опише?

- Атрибути
- Интерфејси
- Операции

Во дизајнот на ниво на компоненти, упорните извори на податоци се однесуваат на

- Бази на податоци
- Датотеки

Дизајн на содржина на WebApp на ниво на компонента се фокусира на содржините и начинот на кој тие комуницираат.

- False

Функционална архитектура на WebApp ги опишува клучните функционални компоненти и како тие комуницираат едни со други.

- True

Која од овие конструкции се користи во структурно програмирање?

- Состојба



- Повторување
- Секвенца

Која од овие е графичка нотација за прикажување на деталите за процедурата?

- Дијаграм

Треба да се користи табела за одлуки

- кога комплексен сет на услови и активности се појавува во компонентата

Јазик за дизајнирање на програма (PDL) е честопати

- комбинација на програмски конструкции и наративен текст

Во софтверското инженерство базирано на компоненти, тимот за развој ги испитува барањата за да се види кои се подложни на композицијата, отколку за градежништвото, пред да започнат деталните задачи за дизајнирање.

- True

Кое од следниве не е една од најголемите активности на доменското инженерство?

- Валидација

Кој од следниве фактори не би се сметал за време на квалификацијата на компонентата?

- потребна опрема за тестирање

Кое е следново е техника што се користи за завиткување на компонентите?

- завиткување со јасни кутии

Кое од следниве не е едно од проблемите што претставува основа за дизајн за повторна употреба?

- програмирање ориентирана кон објекти

Во околината за повторна употреба, пребарувањата во библиотеките често се карактеризираат со користење на елементот на 3C моделот (3C Model).

- Контекст

Кој од следниве принципи за дизајнирање на интерфејс не дозволува корисникот да остане во контрола на интеракцијата со компјутер?

- обезбедете само еден строго дефиниран метод за извршување задача

Кој од следниве принципи за дизајнирање интерфејс го намалува оптоварувањето на меморијата на корисникот?

- дефинирајте интуитивни кратенки
- откриваат информации на прогресивен начин

- воспостави значајни стандардни вредности

Причината за намалување на оптоварувањето на меморијата на корисниците неговата / нејзината интеракција со компјутерот побрзо да се заврши.

- False

Конзистентноста на интерфејсот го подразбира тоа

- влезните механизми остануваат исти во текот на целата апликација
- визуелните информации се организираат според стандарден дизајн

Ако минатото интерактивни модели создадоа одредени очекувања од корисниците, воопшто не е добро да се направат промени во моделот.

- True

Кој модел го прикажува профилот на крајните корисници на компјутерски систем?

- кориснички модел

Кој модел ја прикажува сликата на системот што крајниот корисник го создава во неговата глава?

- перцепција на системот

Кој модел го прикажува изгледот и чувството на корисничкиот интерфејс заедно со сите придружни информации?

- модел на имплементација

Која од овие рамковни активности вообичаено не е поврзана со процесите за дизајнирање на корисничкиот интерфејс?

- проценка на трошоците

Кој пристап (и) за анализата на задачите на корисниците може да биде корисен во дизајнот на кориснички интерфејс?

- проучете ги постојните решенија засновани на компјутер
- набудувајте ги корисниците кои извршуваат задачи рачно

Објективно-ориентирани техники за анализа може да се користат за идентификување и рафинирање на предмети и активности на задачи на корисниците, без потреба да се однесуваат на гласот на корисникот.

- False

Можностите за прикажување на компјутерот се основната одредница на редоследот по кој се завршени активности за дизајнирање на кориснички интерфејс.

- False

Понекогаш е можно дизајнерот на интерфејс да биде ограничен од фактори на животната средина кои ублажуваат против леснотијата на користење за многу корисници.

- True

Едно средство за дефинирање на предметите и дејствата на корисничкиот интерфејс е да се спроведе граматичка анализа на сценариото на корисникот.

- True

Моделите за дизајн на интерфејсот обично вклучуваат комплетен дизајн на ниво на компоненти (дизајн класи, атрибути, операции и интерфејси).

- True

Неколку вообичаени проблеми со дизајнот имаат површина за скоро секој кориснички интерфејс, вклучувајќи

- пакување со грешки
- време на одговор на системот

Поважно е да се привлече вниманието на корисникот со светкави карактеристики отколку ергономски распоред на звук на екранот при градење на WebApp.

- False

Неколку мерки за употребливост можат да се соберат при набудување на корисниците во интеракција со компјутерски систем, вклучувајќи

- број на грешки на корисникот
- поминато време во потрага по материјали за помош

Кое од следниве не е еден од елементите на шемата за дизајн?

- Решение

RubberNecking е пример за класична генеративна шема.

- False

Рамката со рамки е еднократно мини-архитектура што служи како основа што можат да се применат и другите модели на дизајнирање?

- True

Наоѓањето на модели изградени од други кои се однесуваат на проблеми со дизајнот е често потешко од тоа што препознавањето на моделите во апликацијата што треба да се гради.

- True

Јазик на шема

- опфаќа збирка на обрасци

Концептите и техниките дискутирани за можат да се користат во комбинација со пристап заснован на шема.

- Архитектонски дизајн
- Дизајн на ниво на компоненти
- Дизајн на кориснички интерфејс
- **Сите горенаведени**

Важно е да се намали спојувањето помеѓу моделите на дизајнирање за да може да се третираат како независни субјекти.

- False

Решенијата за дизајн на реалниот живот не може секогаш да се позајмуваат на пристап од горе надолу.

- True

Кој од следниве типови на проблеми се користи за означување на колоните во табелата за организирање шема?

- База на податоци
- Инфраструктура

Повеќето грешки во дизајнот базиран на модели можат да се избегнат со разумна употреба на техники за преглед.

- True

Пред да изберете архитектонски дизајн, мора да се процени за соодветноста на апликацијата и целокупниот архитектонски стил.

- True

За разлика од архитектонските модели, моделите за дизајнирање на ниво на компонента може да се применат за решавање на подпроблеми без оглед на контекстот на системот.

- False

Повеќето модели за дизајнирање на кориснички интерфејс спаѓаат во една од категориите на модели.

- 10

Моделите на дизајнирање на веб-апликации може да се класифицираат со оглед на која од димензиите наведени подолу?

- Дизајн фокус

- Грануларност

Кое од следниве се нивоата на фокусирање на дизајнот што можат да се користат за категоризирање на моделите на WebApp?

- Функционални обрасци
- Начини на навигација

Кое од нивоата на грануларност што може да се искористи за да се опишат моделите на WebApp?

- Архитектонски обрасци
- Модели на компоненти
- Модели на дизајнирање

Која од следниве карактеристики не треба да се користи за проценка на квалитетот на една веб-програма?

- Естетика

Кои од следниве се целите за дизајн за секоја веб-програма?

- Едноставност
- Конзистентност
- Завидливост
- Визуелна жалба

Која од следниве не е дел од пирамидата за дизајн за веб-дизајн?

- Дизајн на деловни случаи

Со содржината на WebApps е се, слабо дефиниран кориснички интерфејс и брзо ќе биде занемарен од честите корисници.

- False

Кој од овие се механизми за интеракција преку WebApp?

- Графички икони
- Графички слики
- Менија за навигација
- **Сите горенаведени**

Дизајнот на изгледот на екранот има неколку широко прифатени стандарди засновани на истражување на човечки фактори,

- False

Графичкиот дизајн го смета секој аспект на изгледот како чувство на WebApp.

- True

Дизајн на содржина го спроведува

- Писатели и графички дизајнер
- Веб-инженери

Предметите на содржината имаат и атрибути со информации што се дефинирани при специфични атрибути за анализа и имплементација, специфицирани за време на дизајнот.

- True

Објектите со содржина не се одвиваат нормално на веб-страниците се додека не започнат со имплементација.

- False

Архитектурата на содржини и архитектурата на WebApp се приближно иста работа за многу WebApps?

- False

Кое од следниве не е една од содржинските архитектонски структури што ги користат веб-инженерите?

- Паралелно

MVC е трислојна архитектура која содржи

- модел, преглед и контролер

Дизајн на веб-навигација вклучува создавање на семантичка навигациска единица за секоја цел поврзана со секоја дефинирана улога на корисникот.

- True

За да му дозволите на корисникот да се чувствува под контрола на WebApp, добра е идејата да се мешаат и хоризонталните и вертикалните механизми за навигација на истата страница.

- False

Дизајн на ниво на компоненти за WebApps е многу сличен на дизајнот на ниво на компоненти за други средини за испорака на софтвер.

- True

Која од овие не е една од дизајнерските активности поврзани со дизајн-ориентирана хипермедија?

- дизајн на содржина

UML нема никакви шеми за застапеност кои се корисни во градењето на модели за дизајнирање на веб-апликации.

- False

Квалитетот на сообразноста се фокусира на степенот до кој имплементацијата на дизајнот ги исполнува неговите барања и цели на изведба.

- True

Кој од следниве не е еден од атрибутите за квалитет на софтверот?

- Отстранувањата треба да ги земат предвид проблемите со перформансите

Квалитетот на производот може да се процени само со мерење на факторите на тврд квалитет.

- False

Многу софтверски метрика можат да бидат индиректни мерки.

- True

Кои од следниве се ISO 9126 фактори за квалитет на софтверот?

- Функционалност
- Преносливост
- Доверливост

Програмерите треба да создадат колекција на насочени прашања за да го проценат секој фактор на квалитет.

- True

Софтверските метрики претставуваат директни мерки за одредена манифестација на квалитет.

- False

Дилемата за квалитет може да се сумира како избор помеѓу брзо градење работи или правилно градење работи.

- False

Доволно добар софтвер обезбедува високо квалитетни софтверски функции, заедно со специјализирани функции кои содржат грешки во знаењето.

- True

Кое од следниве најверојатно ќе биде најскапата цена за квалитет?

- Надворешни трошоци за неуспех

Лошиот квалитет доведува до ризици на софтвер што можат да станат сериозни?

- True

Кора системот не успее да ги испорача потребните функции, тоа е затоа што клиентот ги менува барањата?

- False

Програмерите мора да започнат да се фокусираат на квалитетот во текот на фазата на дизајнирање, со цел да градат безбедни системи.

- True

Која од следниве решенија за управување има потенцијал да влијае на квалитетот на софтверот?

- Одлуки за проценка
- Одлуки ориентирани кон ризик
- Заказни решенија
- **Сите горенаведени**

Проектниот план треба да вклучува експлицитни техники за управување со и?

- Промена
- Квалитет

Контролата на квалитетот опфаќа низа активности на софтверско инженерство кои помагаат да се осигура дека секој работен производ ги исполнува своите квалитетни цели.

- True

Целта на обезбедување на квалитет е да се осигура дека софтверски проект е без грешки.

- False

Целта на прегледите на софтверите да се откријат грешките и дефектите во работните производи, за да може да се отстранат пред да преминат во следната фаза на развој.

- True

Во принцип, порано е откриен дефект на софтвер и коригиран на помалку скапиот за севкупниот буџет на проектот.

- True

Моделите за засилување на дефектите можат да се користат за да се илустрираат трошоците поврзани со користење на софтвер од првичното распоредување до нејзиното пензионирање.

- False

Методите за преглед може да се користат за да се процени ефикасноста на секоја активност за преглед.



- True

Густината на дефектите може да се процени за кој било софтверски инженерски производ на работа.

- True

Агилните развивачи на софтвер се свесни дека прегледите на софтверот секогаш одземаат време, без да заштедите.

- False

Нивото на формалност на прегледот е одредено со кое од следниве?

- количина на подготовка
- следење на рецензент
- структура на преглед

Кое од следниве може да се состои од неформален преглед?

- обична средба
- проверка на биро

Кои од следниве се цели за формални технички прегледи?

- Откријте грешки во софтверски производи за работа

На крајот на официјалниот технички преглед, сите присутни можат да одлучат

- прифатете го работниот производ без модификација
- отфрлете го производот поради сериозни грешки

Краток преглед на извештајот одговара на кои три прашања?

- што беше разгледано, кој го разгледа, кои беа наодите

Во кој било вид на технички преглед, фокусот на прегледот е на производот, а не на производителот.

- True

Прегледите управувани со примерок имаат смисла само за многу мали проекти за развој на софтвер.

- False

Квалитетот на софтверот може да се дефинира како усогласеност со експлицитно наведените барања и стандарди, ништо повеќе и ништо помалку.

- False

Луѓето кои вршат гаранција за квалитет на софтверот, мора да го погледнат софтверот од гледна точка на клиентот.

- True

Елементите за обезбедување на квалитет на софтвер се состојат од прегледи, ревизии и тестирање.

- False

Која од овие активности не е една од активностите што се препорачува да ги изврши независна група на SQA?

- служат како единствен тим за тестирање за кој било произведен софтвер

Метриците можат да се користат за да се означи релативната моќност на атрибутот за квалитет на софтверот.

- True

Обидите да се примени математички доказ за да се докаже дека програмата е во согласност со нејзините спецификации се осудени на неуспех.

- False

Статистичкото обезбедување на квалитет вклучува

- трагнување на секој дефект на основната причина, изолирање на „виталните неколку“ причини и движење за нивно коригирање

(Six Sigma) Шест сигма методологија дефинира три основни чекори.

- дефинира, мери, анализира

Проблемите со сигурноста на софтверот скоро секогаш можат да се следат со

- грешки во дизајнот
- грешки во спроведувањето

Безбедност на софтверот активност за обезбедување квалитет, кој се фокусира на опасностите што се

- може да предизвика да не успее целиот систем

Стандард ISO за обезбедување на квалитет што се однесува на софтверското инженерство е

- ISO 9001

Кое од следниве не е дел во стандардот за плановите на SQA препорачани од IEEE?

- Буџет

Во работата за обезбедување на квалитет на софтверот не постои разлика помеѓу верификација на софтвер и валидација на софтвер.

- False

Најдобрата причина за користење независни тимови за софтверски тест е

- конфликтот на интереси помеѓу развивачите и тестерите е намален

Кој е нормалниот редослед на активности во кои се организира традиционалното тестирање на софтвер?

- тестирање на интеграција
- тестирање на системот
- тестирање на единици
- тестирање на валидација

Со собирање на софтверски метрика и употреба на постојни модели за доверливост на софтвер, можно е да се развијат значајни упатства за утврдување кога е направено тестирање на софтвер.

- True

Кое од следниве стратешки проблеми треба да се реши во успешен процес на тестирање на софтвер?

- спроведе формални технички прегледи пред тестирање
- наведете барања на квантитативен начин

Кое од следниве треба да се процени за време на тестирањето на единицата?

- ракување со грешки
- патеки за извршување

Единици и никулци не се потребни за тестирање на единиците, бидејќи модулите се тестираат независно еден од друг.

- False

Тестирањето од интеграција од горе надолу има најголема предност и за тоа

- големите точки на одлучување се тестираат рано
- не треба да се пишуваат возачи

Тестирањето на дното на интеграција има најголема предност а за тоа

- не треба да се пишуваат никулци

Регресивното тестирање треба да биде нормален дел од тестирањето за интеграција затоа што како нов модул се додава нов систем

- контролната логика се повикува

- воспоставени се патеки за проток на податоци

Тестирањето на чад најдобро може да се опише како

- тестирање на тркалање на интеграција

При тестирање на софтвер-ориентиран софтвер, важно е да се тестира секоја класа операција одделно како дел од процесот на тестирање на единицата.

- False

Интеграција стратегија за тестирање OO вклучува тестирање

- групи класи кои соработуваат или комуницираат на некој начин

Бидејќи многу WebApps се развиваат постојано, процесот на тестирање мора да биде во тек, исто така.

- True

Фокусот на тестирање на валидација е да открие места за кои корисникот ќе може да забележи неуспех на софтверот да ги исполни неговите барања.

- True

Валидацијата на софтверот се постигнува преку серија тестови што ги прави корисникот откако софтверот ќе се распореди во неговата или нејзината работна околина.

- False

Прегледите за конфигурација не се потребни ако тестирањето за регресија е строго применето за време на интеграцијата на софтверот.

- False

Тестовите за прифаќање обично се спроведуваат од страна на

- крајните корисници

Тестирањето за обновување е системски тест што го принудува софтверот да не успее на најразлични начини и потврдува дека софтверот во состојба да го продолжи извршувањето без прекини.

- False

Безбедносните тестови се обидуваат да потврдат дека механизмите за заштита, вградени во системот, го штитат од неправилно навлегување.

- True

Тестирањето на стрес ги испитува притисоците поставени врз корисникот за време на употребата на системот во екстремни околина.

- False

Тестирањето на перформансите е важно само за системи во реално време или вградени системи.

- False

Дебагирање не е тестирање, но секогаш се јавува како последица на тестирање.

- True

Кое од следниве е пристап кон дебагирање?

- одвраќање
- брутална сила
- да предизвикаат елиминација

Со темелно тестирање, можно е да се отстранат сите дефекти од програмата пред испораката до клиентот.

- False

Кое од следниве се карактеристиките на тестираниот софтвер?

- набудување
- едноставност
- стабилност
- **Сите горенаведени**

Техниката за тестирање што бара осмислување тест случаи за да се докаже дека секоја програма функција е оперативна, се нарекува

- тестирање на црни кутии

Техниката за тестирање што бара осмислување тест случаи за вежбање на внатрешната логика на софтверски модул се нарекува

- тестирање во бела кутија

Кои видови на грешки се пропуштаат со тестирање на црни кутии и можат да се откријат со тестирање во бела кутија?

- логички грешки
- типографски грешки

Графиконите на проток на програмата се идентични со табелите за проток на програмата.

- False

Метрика за цикломатска сложеност му дава на дизајнерот информации за бројот на

- независни логички патеки во програмата

Цикломатската сложеност на програмата може да се пресмета директно од PDL застапеност на алгоритам без да се подготви графикон на проток на програма.

- True

Тестирањето на состојбата е техника на тестирање на контролна структура каде што критериумите што се користат за дизајнирање на случаи на тестирање се дека тие

- ги вежбаат логичките услови во програмски модул

Тестирање на проток на податоци е техника на контролна структура за тестирање каде што критериумите што се користат за дизајнирање на случаи на тестирање се дека тие

- избираат патеки за тестирање врз основа на локацијата и употребата на променливите

Тестирање на јамка е техника на контролна структура за тестирање каде што критериумите што се користат за дизајнирање на случаи на тестирање се дека тие

- се фокусираат на тестирање на валидноста на конструкциите на јамки

Тестирањето со црни кутии се обидува да пронајде грешки во која од следниве категории

- неточни или исчезнати функции
- грешки во интерфејс
- грешки во работењето

Методите за тестирање врз основа на графикони можат да се користат само за системи ориентирани кон објекти

- False

Тестирањето на еквивалентност го дели влезниот домен во класи на податоци од кои може да се изведат тест случаи за да се намали вкупниот број тест случаи што мора да се развијат.

- True

Анализата на граничната вредност може да се користи само за да се направи тестирање во бела кутија.

- False

Ортогналното тестирање на низи му овозможува на дизајнерот на тестот да го зголеми покриеноста на тест-случаите направени за релативно мали влезни домени.

- True

Тестот добиен од моделите на класи на однесување треба да се заснова на

- дијаграм на состојба на транзиција

Архитектурата на клиентот / серверот не може правилно да се тестира затоа што мрежното оптоварување е многу променливо.

- False

Апликациите во реално време додаваат нов и потенцијално тежок елемент во мешавината за тестирање

- време

Не е можно да се тестира објектно-ориентиран софтвер без да се вклучат техники за откривање грешки, применети на системот OOA и OOD модели

- True

Коректноста на моделот OOA и OOD се постигнува со употреба на формални технички прегледи од тимот за обезбедување на квалитет на софтверот.

- False

Конзистентноста на моделите ориентирани кон објектите може да се оцени со прегледување на моделот на картички CRC.

- True

Дизајнот на тест случаи за OO софтвер е воден од алгоритмички детали за поединечните операции.

- True

Тестирање на интеграција на софтвер-ориентиран софтвер може да се постигне со која од следниве стратегии за тестирање?

- Тестирање на кластери
- Тестирање врз основа на навој
- Користете-базирани тестирање

Валидацијата на софтверот ориентирана кон предмет се фокусира на корисничките видливи активности и излези од системот.

- True

Екапсулацијата на атрибутите и операциите во објектите го олеснува добивањето информации за состојбата на објектот за време на тестирањето.

- False

Случаите за употреба може да обезбедат корисен придонес во дизајнот на црни кутии и државни тестови на OO софтвер.

- True

Тест-базиран на грешки е најдобро резервиран за

- операции и класи кои се критични или сомнителни

Тестирањето на операциите на OO класа е потешко од

- наследство
- полиморфизам

Тест врз основа на сценарио

- се концентрира на интеракција меѓу актерот и софтверот
- испуштање грешки во спецификациите

Тестирањето на длабока структура не е дизајнирање

- вежба структура што може да се набудува од страна на корисникот

Тестовите за случаен редослед се спроведуваат за да се применат различни историски животни истории.

- True

Која од овие техники не е корисна за тестирање на партиции на ниво на класа

- партиционирање на класи на еквивалентност

Повеќе тестирање на класи е премногу сложено за да се тестира со случаен тест случаи.

- False

Кое од следниве не е една од димензиите на квалитет што се користи за проценка на WebApp?

- Одржливост

WebApps бараат посебни методологии за тестирање затоа што грешките во WebApp имаат неколку уникатни карактеристики.

- True

Бидејќи WebnApps се развива постојано, процесот на тестирање е тековна активност, спроведена од персоналот за поддршка на веб користејќи регресивни тестови.

- True

Планирањето на тестот не се користи во тестирањето на WebApp.

- False

Бидејќи е конструирана архитектурата WebApp, кои типови тестирање се користат како тестови за интеграција?

- Тестирање на компонентата
- Тест за навигација

Кое од следниве не е една од целите на тестирањето на содржината на WebApp?

- Идентификувајте ги грешките во поврзувањето



Тестирањето на базата на податоци е многу ретко дел од тестирањето на содржината на WebApp.

- False

Целокупната стратегија за тестирање на интерфејс е да открие грешки

- во семантиката за навигација
- поврзани со специфични механизми за интерфејс

Кое од следниве не е механизам за интерфејс на WebApp?

- Форми (Forms)

При тестирање на семантиката за интерфејс на WebApp, секој случај се користи како влез за дизајнирање на секвенца за тестирање.

- True

Тестовите за употребливост треба да бидат дизајнирани и извршени од наменетите корисници за дадена веб-програма.

- False

Тестирањето за компатибилност преку WebApp е спроведено за да бидеме сигурни дека корисничкиот модел за сценарио за употреба одговара на категоријата на корисници доделена на даден корисник.

- False

Која техника (и) за дизајнирање на случајот е соодветна за тестирање на ниво на компоненти на WebApp?

- Анализа на гранична вредност
- Поделба за еквивалентност
- Тестирање на патеката
- **Сите горенаведени**

Целта на синтаксичкото тестирање за навигација преку WebApp е да се обезбеди правилен изглед на секој механизам за навигација.

- False

И веб-инженерите и нетехничките корисници вршат тестирање за семантика на навигација за веб-апликации.

- True

Кој од следниве не е еден од елементите што треба да се земат предвид при конструирање на тестови за конфигурација на страната на серверот на WebApp?

- Компатибилност на прелистувачот

За да се дизајнираат тестови за конфигурација од страна на клиентот, секоја категорија на корисници се проценува за да се намали бројот на варијабли за конфигурација на управувачки број.

- True

Кое од следниве не е докажлив безбедносен елемент на WebApp?

- Пенетрација

Тестовите за изведба на WebApp се дизајнирани да

- симулирајте ситуации во оптоварување во реалниот свет

Тестирањето на оптоварување вклучува утврдување на влезот од кои 3 варијабли?

- N, T, D

Тестот за стрес WebApp е тестирање за оптоварување кое продолжува.

- True

Стратегијата за чистење простории е заснована врз моделот на софтверски процеси.

- Постепено (incremental software process model)

Стратегијата за чистење простории се потпира на

- тестови што го практикуваат софтверот како што навистина се користат

Употребата на официјални докази за точноста на програмата, како дел од процесот на чистење простории, ја елиминира потребата да се изврши тестирање за дефекти на софтверот.

- False

Во софтверското инженерство за чистење простории, „кутија“ опкружува одреден системски аспект на одредено ниво на детали.

- True

Која спецификација на кутијата опишува апстракција, стимули и одговор.

- Црна кутија

Која спецификација на кутијата го опишува архитектонскиот дизајн за некоја компонента на системот.

- државна кутија

Која спецификација на кутијата е тесно усогласена со процедуралниот дизајн и структурното програмирање.

- јасна кутија

Во софтверот за чистење простории се користи структурниот програмски пристап

- рафинирање на дизајн на податоци
- дизајн на функција за рафинирање

Со користење само структурирани програмирани конструкции, додека креирате процедурален дизајн, ја олеснувате работата за докажување на точноста на дизајнот.

- True

Кое од следниве не е предност за користење на строга верификација на исправноста на секоја префинетост на дизајнот на чистата кутија.

- ги подобрува перформансите на кодот

Тестирањето за статистичка употреба се потпира врз дистрибуција на веројатност врз основа на

- начинот на кој всушност ќе се користи софтверот

Сертификацијата на прирачниците завршена откако ќе го помине официјалниот процес на верификација.

- False

Кој од следниве модели е дел од процесот на сертификација на просторијата за чистење?

- компонента модел
- модел на земање мостри

Инваријанта на податоците е збир на услови кои се вистинити за време на извршувањето на која било функција.

- False

На некои формални јазици, зачуваните податоци до кои пристапува системот и ги менува се нарекуваат

- Држава

Во работата на формалните методи, акција што чита или пишува податоци во состојба се нарекува

- Работење

Што ги дефинира околностите во кои одредена операција е валидна?

- Предуслов

Заедничка нотационална конвенција во многу формални методи е да ја напишете променливата со премиер во пост-условот за оператор.

- True

Кои од овие се компоненти на формален јазик за спецификација?

- семантика што ги дефинира предметите што се користат за опишување на системот

- збир на односи што ги дефинираат правилата на предметот
- синтакса што ја дефинира нотацијата
- **Сите горенаведени**

(OCL) OCL користи неколку специјализирани математички знаци.

- False

Јазикот2 употребува шеми за да ги опише состојбите во системот во однос на вредностите доделени на системските променливи.

- True

Користењето на формални методи ја елиминира потребата да се напишат коментари на природен јазик во документот за спецификации.

- False

Која од овие се валидни ставки за конфигурација на софтвер?

- алатки за случај
- документација
- извршни програми
- податоци за тестирање

Кое од следниве не се смета за еден од четирите важни елементи што треба да постојат кога се развива систем за управување со конфигурација?

- елементи за валидација

Штом производот за работа со софтверско инженерство ќе стане основна, тој не може повторно да се промени.

- False

Кои објекти за конфигурација обично не се наоѓаат во базата на податоци на проектот?

- податоци за маркетинг
- опис на организациската структура

Современите практики за софтверско инженерство обично се обидуваат да ги одржуваат SCI во базата на податоци на проектот или складиштето.

- True

Мета-моделот на складирање на податоци се користи за да се утврди како

- информациите се чуваат во складиштето

- и може да се одржи интегритетот на податоците
- лесно постојниот модел може да се продолжи
- **Сите горенаведени**

Многу барања за складирање на податоци се исти како оние за типична апликација за база на податоци.

- True

Способноста за следење на врските и промените во објектите за конфигурација е една од најважните карактеристики на складиштето SCM.

- True

Која од следниве задачи не е дел од управувањето со софтверската конфигурација?

- статистичка контрола на квалитетот

Основен објект за конфигурација е креиран од софтверски инженер за време на некоја фаза од процесот на развој на софтвер.

- единица на информации

Системите за контрола на верзиите воспоставуваат промена поставена како дел од нивната примарна функционалност.

- False

Контролата на промената не е неопходна доколку група за развој користи автоматизирана алатка за бази на податоци за проекти.

- False

Кога управувањето со софтверската конфигурација е формална активност, ревизијата за конфигурација на софтвер ја спроведува

- група за обезбедување квалитет

Примарната цел на известувањето за статусот на конфигурацијата е да

- се осигурете дека информациите за промена им се соопштуваат на сите засегнати страни

Прашањата за конфигурацијата што треба да се земат предвид при развој на веб-апликации вклучуваат:

- содржина
- цена
- луѓе

Предметите за конфигурација на WebApp можат да се управуваат на ист начин како и конвенционалните објекти за конфигурација на софтвер, освен за:

- Содржини

Управувањето со содржината воспоставува процес со кој веб-содржината се дава на екранот на корисникот.

- False

Управувањето со промените за WebApps најдобро се постапува на агилен начин.

- True

Една од причините поради што контролата на верзијата е тешка за WebApps е тоа што во неконтролирано огжружување, можете да имате повеќе автори кои прават измени во истите датотеки од повеќе локации, без да ги реализирате.

- True

Потребно е од развивачите да ги проверат и вметнуваат елементите за веб-конфигурација и автоматски да испраќаат пораки засегнати страни од засегнатите страни, се добри начини да се справат со ревизорската конфигурација и известување за WebApps.

- True

Повеќето метрики за технички софтвер опишани во ова поглавје претставуваат индиректни мерки софтверски атрибути кои се корисни во квантитативната проценка на квалитетот на софтверот.

- True

Кои од следните се причини за користење технички мерки на производ за време на развој на софтвер?

- обезбедува софтверски инженери со објективен механизам за проценка на квалитетот на софтверот

Која мерна активност недостасува од списокот подолу?

❖ Формулација

❖ Колекција

❖ Анализа

❖ Толкување

- **повратни информации**

Парадигмата на целта / прашањето / метрика (GQM) беше развиена како техника за доделување вина за дефекти на софтверот.

- False

Еден од најважните атрибути за метрика на софтверски производи е дека треба да биде

- лесен за пресметување

Во многу случаи, метрика за еден модел може да се користат во подоцнежните активности за софтверско инженерство (на пр., Метрика за дизајнирање може да се користат во планирањето на тестот).

- True

Функциската точка метрика е пример на метрика што може да се искористи за да помогне при техничко одлучување засновано врз информациите за моделот за анализа, без да се користат историските податоци за проектот.

- False

Спецификациските метрики што ги предложи Дејвис кои две карактеристики се потребни за софтверските побарувања?

- специфичност и комплетност

Методиите на архитектонскиот дизајн се фокусираат на

- архитектонска структура
- ефикасност на модулот

Кое од следниве не е мерлива карактеристика на објектно-ориентираниот дизајн?

- Ефикасност

Длабочината на наследството на дрвото (ДИТ) метриката може да му даде на ОО дизајнерот на софтвер читање на

- потребно е време на завршување за спроведување на системот

Бидејќи класата е доминантна единица во ОО системите, нема повик за дефинирање на метрика ориентирана кон класа.

- False

Ако наидете на час со голема одговорност (големина на голема класа или CS вредност), треба да размислите

- поделба на часот

Метриките на ниво на компонентите вклучуваат мерки на

- сложеност
- спојка
- модулна кохезија

Бидејќи класата е доминантна единица во OO системите, многу малку метрики се предложени за операции што живеат во рамките на една класа.

- True

Метрика за интерфејс се користи за да се процени комплексноста на влезните и излезните односи на модулот со надворешните уреди.

- False

Повеќето WebApps лесно можат да се карактеризираат со разумна употреба на широко признати пакети на софтверски метрики?

- False

Методите на изворниот код на Halstead се засноваат на бројот на

- операнди во програмата
- оператори во програмата

Метрика за тестирање на софтвер спаѓа во две широки категории

- метрики што се фокусираат на покриеноста на тестот
- метрики што го предвидуваат бројот на потребни случаи за тестирање

Индексот на зрелост на софтверот IEEE (SMI) се користи за да се обезбеди мерка за

- стабилност на софтверски производ бидејќи е изменет за време на одржувањето

Ефективното управување со софтверскиот проект се фокусира на

- луѓе, производ, процес, проект

Организациите кои постигнуваат високи нивоа на зрелост во управувањето со луѓе имаат поголема веројатност за спроведување на ефективни процеси за софтверско инженерство.

- True

Првиот чекор во планирањето на проектот е

- утврдете ги целите и обемот.

Рамковните активности на процесите се населени со

- пресвртници
- работни производи
- QA поени
- **Сите горенаведени**



Управувањето со проекти е помалку важно за современ развој на софтвер бидејќи повеќето проекти се успешни и завршени навреме.

- False

Кое од следниве не се смета за чинител во процесот на софтвер?

- продажба на луѓе

Најдоброто лице што може да се вработи како водач на проектниот тим е најкомпетентниот достапен практичар за софтверско инженерство.

- False

Најдобриот организационен модел на проектниот тим што треба да се користи при справување со екстремно сложените проблеми е

- отворена парадигма

Кои фактори треба да се земат предвид при изборот на организациската структура за софтверски тим?

- посакуван степен на комуникација
- предвидената големина на добиената програма
- ригидност на датумот на испорака

Еден од најдобрите начини да се избегне фрустрација за време на процесот на развој на софтвер е

- дајте им на членовите на тимот поголема контрола врз процесите и техничките одлуки.

Малите агилни тимови немаат место во современ развој на софтвер.

- True

Која од овие карактеристики на софтверот не е фактор што придонесува за тешкотии во координацијата на проектот?

- Перформанси

Која од овие карактеристики на софтверот се користи за да се утврди обемот на софтверскиот проект?

- информативни цели, функција, перформанси

Главните области на разградување на проблемот за време на активностите за опфат на проектот се

- функционалност што треба да се испорака
- процес кој се користи за испорака на функционалност

Распаѓањето на производот и процесите се одвива истовремено додека се развива планот на проектот.

- True

Кога може да се избере рамковните активности да бидат изоставени за време на распаѓањето на процесите?

- никогаш активностите се непроменливи

Како менаџер на софтверски проекти треба да дејствува за да се минимизира ризикот од дефект на софтверот?

- започнете со десната нога
- следи напредокот

Принципот W5HH содржи кое од следниве прашања?

- Зошто системот се развива?
- Каде се наоѓаат организациски?
- Колку од секој ресурс е потребен?

Која од овие се критични практики за управување со проекти засновани врз перформанси?

- следење на дефекти од квалитетни цели
- емпириска проценка на трошоците
- формално управување со ризик

Која од овие се валидни причини за мерење на софтверски процеси, производи и ресурси?

- да ги карактеризираат
- да ги оцени
- да ги подобриме

Индикатори за процеси овозможуваат софтверски проект менаџер да

- ниту едно од горе наведените

Јавните метрики се користат за

- да се направат стратешки промени во процесот на софтвер.
- да се направат тактички промени за време на софтверски проект

Која од следниве артикли не се мери со метрика на софтверски проект?

- Пазари

Квалитетот и функционалноста на софтверот мора да се мерат индиректно.

- True

Кое од следниве се предностите на користењето на LOC (линиите на шифрата) како метрика ориентирана кон големината?

- LOC лесно се пресметува.

Кои од следниве се предностите на користењето на точките за функција (FP) како мерка за функционалноста испорачана од софтверска апликација?

- FP е мерка независна за јазик.
- FP може да се пресмета пред да се заврши дизајнот.

Нема потреба од усогласување на мерките LOC и FP бидејќи секоја од нив има значајно значење како проектна мерка.

- False

Проектно-ориентираните мерки на проектот може да се комбинираат со историски податоци за проектот за да се обезбедат метрики што помагаат во проценката на проектот.

- True

Метрика ориентирана кон употреба се пресметува директно од UML дијаграмите, кои често се користат како мерки за нормализација.

- False

Кое од следниве не е мерка што може да се собере од проект за веб-апликација?

- Индекс на прилагодување

Кој од следниве фактори за квалитет на софтверот веројатно ќе влијае на радикалните промени во компјутерската архитектура?

- ниту едно од горе наведените

Кое од следниве обезбедува корисни мерки за квалитет на софтверот?

- коректност, одржливост, интегритет, употребливост

Метрика за квалитет на софтвер што може да се користи и во процесот и на нивото на проектот е ефикасност на отстранување на дефекти (DRE).

- True

Зошто е важно да се измери процесот на софтверско инженерство и софтвер што го произведува?

- Да се утврди дали софтверската група се подобрува или не.

За да бидат ефективна помош при подобрување на процесот, користените основни податоци мора да бидат:

- постојано мерено преку проекти
- подготвени од слични проекти

Основните податоци мора да се соберат на постојан начин и не можат да се пресметаат со формално проучување на историските податоци за проектот.

- False

Малите софтверски организации веројатно нема да забележат никаков економски поврат од воспоставувањето програма за софтверски метрика.

- False

Софтверските метрики избрани од една организација се водени од деловни или технички цели што една организација сака да ги оствари.

- True

Бидејќи проценките на проектот не се целосно сигурни, тие можат да бидат игнорирани откако ќе започне проект за развој на софтвер.

- False

Целта на планирањето на софтверски проекти е да

- му овозможат на менаџерот да направи разумни проценки на трошоците и распоредот.

Опсегот на проектот е дефиниран како средство за врзување на системот

- функционалност
- перформанс

Изводливоста на софтверот се заснова на кое од следниве

- технологија, финансии, време, ресурси

Одреден е бројот на лица потребни за софтверски проект

- откако ќе се направи проценка на развојниот напор.

Компоненти за еднократно користење софтвер мора да бидат

- каталогизиран за лесно упатување.
- стандардизиран за лесна примена
- потврдена за лесна интеграција
- **Сите горенаведени**

Опкружувањето на софтверското инженерство (ЈИЕ) се состои од кое од следниве?

- хардверски платформи
- софтверски алатки

Техниките за проценка на софтверските проекти можат широко да се класифицираат според кој од следниве наслови?

- техники на распаѓање
- емпириски модели

Проценката на големината за да се изгради софтверски производ мора да се заснова на директна мерка како LOC.

- False

Процената заснована на проблеми се заснова на распаѓање на проблемот, на кој се фокусира

- вредности на доменот на информации
- функции на софтвер

Техниките за проценка засновани врз ЛОЦ бараат распаѓање на проблемот заснован врз

- функции на софтвер

Техниките за проценка врз основа на FP бараат распаѓање на проблеми засновано на

- вредности на доменот на информации

Техниките за проценка засновани врз процесите бараат распаѓање на проблемот врз основа на

- функции на софтвер
- процесни активности

За разлика од LOC или точката за функција, "случајот за употреба" на секоја личност е точно иста големина.

- False

Кора договорот меѓу проценките е слаб, причината може често да се проследи на несоодветно дефиниран обем на проектот или несоодветни податоци за продуктивност.

- True

Моделите на емпириска проценка обично се базираат на

- регресивни модели добиени од податоци за историски проекти

SOCOMO II е пример за пакет на современи модели на емпириска проценка за кои е потребна големина на информации изразени како:

- функционални точки
- линии на код
- објекти на објекти

- **некое од горенаведените**

Софтверската равенка на Putnam е динамичен емпириски модел кој има два независни параметра: проценка на големината и индикација за времетраење на проектот во календарски месеци или години.

- True

Функционалните точки немаат корисник во развивање проценки за софтвер ориентиран кон објект.

- False

Во агилниот развој на софтвер техниките за проценка се фокусираат на потребното време за да се заврши секоја од нив

- Прираст

Можно е да се користи модифицирана техника на точка на функција за да се направат проценки за веб-апликации.

- True

Користењето статистичка техника, како што е анализата на дрво на одлуки, може да помогне во сортирање на вистинските трошоци поврзани со одлуката за купување.

- True

Екстернализацијата секогаш обезбедува едноставно средство за стекнување софтвер со пониска цена отколку на лице место за развој на истиот производ.

- False

Софтверските проекти неизбежно доцнат и нема нешто што може да објасни зошто.

- False

Неетички е да се спроведе проект за кој знаете однапред да не може да се заврши до крајниот рок на клиентот, освен ако не го информирате клиентот за ризикот и не воспоставите план за проект кој може да го испорача потребниот систем постепено.

- True

Кое од следниве не е еден од водечките принципи на распоред на софтверски проекти:

- проценка на пазарот

Со дуплирање на големината на вашиот софтверски проект тим е загарантирано да го намали времето за завршување на проектот на половина.

- False

Со равенката на софтверот може да се искористи за да се покаже дека со малку продолжување на крајниот рок на проектот

- потребни се помалку луѓе

Правилото 40-20-40 сугерира дека треба да се потроши најмал обем на напор за развој

- Кодирање

Збир на задачи е збирка од

- инженерски работни задачи, пресвртници, испораки

Мрежата за задачи (активности) е корисен механизам за

- откривање на зависности од меѓусебна задача
- утврдување на критичкиот пат

Задачите што лежат на критична патека во работна мрежа можат да бидат завршени по кој било редослед, се додека проектот е на распоред.

- False

Две алатки за пресметување на критична патека и времиња на завршување на проектот од мрежни активности се

- CPM (CPM)
- ПЕРТ(РЕКТ)

Временските графикони им помагаат на менаџерите на проектот во одредувањето на кои задачи ќе се извршуваат во даден момент.

- True

Најдобар индикатор за напредок во софтверски проект е завршувањето

- и успешно прегледување на дефиниран софтверски производен производ

Бидејќи моделот на итеративен процес најдобро функционира за проекти ориентирани кон објекти, невозможно е да се утврди дали инкрементот ќе се заврши навреме или не.

- False

Проектите на WebApp бараат само создавање макро распоред.

- False

Целта на заработената анализа на вредности е да

- обезбеди квантитативно средство за проценка на напредокот на софтверскиот проект
- Заработената анализа на вредност е техника која им овозможува на менаџерите да преземат корективни активности пред да се развие проектна криза.

- True

Проактивното управување со ризикот понекогаш се опишува како противпожарна борба.

- False

Ризикот од софтвер секогаш вклучува две карактеристики

- неизвесност и загуба

Три категории на ризици се

- проектни ризици, технички ризици, деловни ризици

Општите ризици бараат многу поголемо внимание отколку ризиците специфични за производот.

- False

Листата за проверка на ставка на ризик ќе содржи познати и предвидливи ризици од која од овие категории?

- големината на производот
- развојна околина
- големина на персоналот
- дефинирање на процеси

Прашањата што треба да се постават за да се процени целокупниот ризик на проектот, вклучуваат:

- Дали врвните менаџери се формално посветени на поддршка на проектот?
- Дали крајните корисници се посветени на проектот и предложениот систем се градат?
- Дали барањата се целосно разбрани од развојниот тим и клиентите?

Проценката на влијанието на ризикот на софтверот треба да се фокусира на последиците што влијаат

- перформанси, поддршка, цена, распоред

Проекцијата на ризик се обидува да го процени секој ризик на два начина

- веројатност и последици

Табелите за ризици се сортираат според

- веројатност и влијание

Индивидуалните членови на тимот можат да направат своја проценка за веројатност за ризик, а потоа да развијат консензуална вредност.

- True

Кои фактори влијаат на веројатните последици доколку постои ризик?

- тајминг на ризик
- onser на ризик



Причината за рафинирање на ризиците е да се разделат во помали единици со различни последици.

- False

Ефективниот план за управување со ризикот треба да се осврне на кое од овие проблеми?

- избегнување на ризик
- следење на ризикот
- планирање на вонредни состојби
- **Сите горенаведени**

Следењето на ризикот вклучува гледање на индикаторите за ризик дефинирани за проектот и не утврдување на ефективноста на самите чекори за намалување на ризикот.

- False

Анализата на опасноста се фокусира на идентификување и проценка на потенцијалните опасности што можат да предизвикаат

- цел систем да пропадне

Информациите за ризиците (РИС) никогаш не се прифатлива замена за планот за целосно ублажување, следење и управување со ризиците (RMMM).

- False

Колку напор обично се троши од софтверска организација за одржување на софтвер?

- 60 проценти - 60%

Поддршката на софтверот не е засегната со обезбедување хардвер или инфраструктура.

- False

Реинженерингот на деловните процеси честопати е придружуван од реинженеринг на софтвер.

- True

Кое од следниве не е пример за деловен процес?

- софтвер за тестирање

Реинженеринг на деловните процеси нема почеток или крај, тоа е еволутивен процес.

- True

Која од следниве активности не е дел од моделот на процес на реинженерирање на софтвер?

- Прототипирање

Моделот на процес на реинженерирање на софтвер вклучува активности за реструктурирање за кои од следниве работни артикли?

- код
- документација
- податоци
- **Сите горенаведени**

Кое од следниве не е проблем да се разгледа кога се работи за обратна инженеринг?

- Поврзаност

Обратно инженерство на податоци се фокусира на

- структури на бази на податоци
- внатрешни структури на податоци

Првата инверзна инженерска активност вклучува барање да се разбере

- обработка

Инверзен инженеринг треба да продолжи со реинженерирање на кој било кориснички интерфејс.

- True

Која од овие придобивки може да се постигне кога се реструктурира софтверот?

- програми со повисок квалитет
- намален напор за одржување
- софтвер полесен за тестирање
- **Сите горенаведени**

Преструктурирањето на шифрите е добар пример за реинженерирање на софтвер.

- False

Кој од овие не е пример за реструктурирање на податоците?

- анализа на податоци

Инженерството за напред не е потребно ако постојниот софтверски производ произведува правилен резултат.

- False

Реинженеринг-клиент / серверски системи започнува со детална анализа на деловното опкружување што опфаќа постојен компјутерски систем.

- True

Единствено време на реинженеринг влегува во работа со наследен систем е кога компонентите ќе бидат имплементирани како објекти.

- False

Придобивките од трошоците добиени од реинженеринг се реализираат главно како резултат на намалените трошоци за одржување и поддршка за новиот софтверски производ.

- True

Подобрувањето на софтверските процеси мора да обезбеди разумна реализација на инвестицијата за да ја оправда неговата употреба.

- True

Ефективен напор за подобрување на софтверскиот процес се потпира на иста рамка за секој проект.

- False

Намерата на модел на зрелост како CCM е да се обезбеди патоказ за добра практика на софтвер.

- False

SPI е оправдан само за големи софтверски организации.

- False

Најтешкиот дел од СПИ е воспоставување консензус за започнување на процесот SPI.

- True

Како се спроведува проценка на процесот, на кое од следниве прашања треба да се фокусираме?

- Прифаќање
- Посветеност
- Конзистентност
- **Сите горенаведени**

Кое од овие лица не е вклучено во активности за едукација и обука на СПИ? SPI

- Клиенти

Честопати е тешко да се постигне консензус меѓу различните изборни единици за време на активноста за избор и оправдување на СПИ. SPI

- True

Кој не е еден од процесите што треба да се земат во предвид при инсталацијата на процесите и миграцијата?

- Како што е

- Овде-до-таму
- Да бидеш

Евалуацијата се јавува само за време на активност на СПИ по смртност. SPI

- False

СПИ често не успева затоа што ризиците не биле соодветно разгледувани и не се случило планирање на јазичноста. SPI

- True

Кој не е еден од клучните фактори за успех што водат кон успешни напори за СПИ? SPI

- Стандардизирани стратегии за СПИ

Интеграција на моделот на доспевање на способноста претставува мета-модел кој се спроведува како

- континуиран модел
- изведен модел

Моделот на зрелост на способност на луѓето сугерира практики што треба да ги следат организацијата за привлекување, развој и задржување на извонреден талент.

- True

Лесно е да се утврдат квантитативните придобивки и мерките за трошоци потребни за пресметување на повратната инвестиција за активности на СПИ. SPI

- False

За да бидат ефективни во современиот развој на софтвер, SPI-рамките мора да станат значително поагилни.

- True

Која од овие не е една од фазите на развој на технологијата?

- Иновации

Циклусот возбуда претставува реален поглед на долгорочната интеграција на технологијата.

- False

Соработката и глобализацијата се примери на меки трендови што ќе влијаат на технолошките инженеринг на софтвер.

- True

Денес ние користиме процес, методи и алатки за да ја задржиме комплексноста под контрола.

- True

Софтверот со отворен свет е друго име за софтверот со отворен код.

- False

Моделите на процесни зголемувања се дел од стратегијата за справување со појавите барања.

- True

Која од овие улоги е дел од тимскиот сон за софтверско инженерство?

- Брајан
- Собрание
- Хак
- **Сите горенаведени**

Решенијата на софтверската платформа му овозможуваат на OEM да посвети на софтверскиот напор за карактеристиките на производите што го разликуваат од другите производи на пазарот.

- True

Кога ќе се достигне точката на вртење, технолошки каскади низ заедницата за софтверско инженерство се појавуваат промени во широка основа.

- True

Инженерите на софтвер ќе стават помалку акцент на инвестицијата за враќање, бидејќи автоматските технологии за софтверски процесни процеси стануваатсе поприсутни.

- False

Успешните соработнички напори вклучуваат кое од следниве?

- Заедничка култура
- Заеднички цели
- Заеднички процес

Поим за спецификација на софтвер во инженерството на барања може да се замени со вредносни управувани барања, произлезени од засегнатите страни од употребата на производот.

- True

Развој на софтвер управувано од моделот ја елиминира потребата да се занимавате со генерирање на код.

- False

Регресивното тестирање не е потребно при развој на тестовите, бидејќи секоја под-функција се тестира како што се развива.

- False

Софтверското инженерско опкружување овозможуваат интеграција на COTS алатките од повеќе пакувања.

- False