**1. Избройте поне три линейни последователни модели на процеса на разработка на софтуер?**

- Водопаден модел

- V- Model

- CleanRoom Engineering

**2. Каква е разликата между линейните последователни и непоследователните модели на процеса на разработка на софтуер?**

- Последователни модели (sequential life-cycle models) - при тези модели се спазва една сравнително строга последователност на дейностите (фазите) при разработване на софтуер

- Инеративни модели (nonsequential life-cycle models) - тези модели (наричат се още циклични модели) допускат връщане (feedbacks) към предишни дейности (фази).

**3. Каква е разликата между инкременталния развой и гъвкавите модели за разработка на софтуер?**

- Инкременталният развой е непоследователен тип.

- При гъвкавите модели се следват принципи. Доставят се работещи версии на продукта на няколко седмици или месец; Клиентите и разработчиците работят заедно по време на целия проект;

**4. Колко и кои са нивата на зрелост в CMM?**

- Ниво 1 – начално

- Ниво 2- повторяемо

- Ниво 3- определено

- Ниво 4- управляемо

- Ниво 5- оптимизиращо

**5. Кои са двете характеристики, с които се определя софтуерния риск?**

- несигурност –няма 100% вероятен риск

- загуба –ако риска стане реален ще се появят нежелани последствия и загуби.

**6. Опишете концепцията за работа със софтуерния риск?**

- Избягване на риска- Разработва се стратегия за намаляване на реорганизацията в персонала.

- Наблюдение на риска - Риска трябва да се наблюдава и да се прецени каква е в даден момент неговата вероятност.

- Управление на риска и планиране на евентуални случаи

**7. Избройте видовете валидационни тестове?**

Alpha тестове и Beta тестове

8. Какви тестове се провеждат при тестване на най-малките компоненти на системата (unit testing)?

- Interface

- Local data structures

- Boundary conditions

- Independent paths

- Error handling paths

**9. Опишете структурата на модела за качество на типичния йерархичен модел?**

- Ниво 0 – Качество

- Ниво 1 - фактори, представящи даден аспект на качеството на софтуера от гледна точка на потребителя.

- Ниво 2 - критерии – представят характеристики на програмния продукт.

- Ниво 3 - метрики –детайли на даден критерии. Те се определят от оценъчните елементи.

- Ниво 4 – оценъчни елементи – елементарни характеристики на най-ниско ниво, коитоподлежат на количествена оценка.

**10. Кои са SOLID принципите избройте ги и ги обяснете?**

- Single Responsibility Principle (SRP). „Класът трябва да има една единствена отговорност.“

- Open-Closed Principle (OCP). “Модулът (компонента) трябва да бъде отворен за разширения, но затворен за модификации.“

- Liskov Substitution Principle (LSP). “Подкласовете трябва да бъдат заменими с техните базови класове.“

- Interface Segregation Principle (ISP).“Много специфични клиентски интерфейси са по-добре от един с обща цел.“

- Dependency Inversion Principle (DIP). “Зависимост от абстракции. Не зависимост от конкретики.”

**11. Кои са стъпките в процеса на валидация на UI?**

- Дефиниране на интерфейсните обекти и действия (операции).

- Дефиниране на събития (действия на потребителя)

- Описание на всяко състояние на интерфейса

- Определете как потребителя интерпретира състоянието на системата от информацията предоставена през интерфейса.

**12. Избройте някои от аспектите на добрия стил на програмиране (поне 4 от общо 9)?**

- коментари

- структорно програмиране

- производителност

- цялостна документация

- форматиране

- странични ефекти

- сигурни програми

- преносими програми

- избор на имена

**13. Кои са критериите за качество на една софтуерна архитектура?**

- Структурираност

- Простота

- Нагледност

**14. Как от ООА модел преминаваме към ООП модел?**

- Като се разшири ООА с класове от потребителския

интерфейс.

**15. Има ли възможност в SCRUM да се добави проектирането като стъпка – кога и как?**

Да, след получаване на изискванията от клиента и плануваната архитектура за желания продукт.

**16.Какви модели се създават по време на ООА?**

Базов, статичен и динамичен модел.

**17.Какво включва извличането на изискванията при ООА?**

- Провеждат се срещи (реални или виртуални) с участието на

софтуерните инженери и клиентите

- Установени са правила за подготовка и участие

- Предложен е дневен ред

- модератор "facilitator" (може да бъде клиент, разработчик или

външен) контролира срещата

- Използва се механизъм за дефиниране на изискванията

(могат да бъдат работни листи, чат, стикери за стена,

електронен бюлетин или виртуален форум)

Комбинация от основни концепции: описание на системата от различни изгледи.

ООА представя изискванията от гледна точка на класове и обекти, открити в речника на проблемния домейн.

**18.Клас диаграмата към кой модел на ООА принадлежи?**

- Статичния

**19.Какво представлява CRC модела?**

Class-responsibility-collaborator моделирането предоставя прост начин за идентифициране и организиране на класовете, които съответстват на системата или изискванията към нея.

CRC модел е колекция от стандартни индекс карти, които представят класовете. Картите са разделени на три секции. В най-горната секция записвате името на класа. В тялото на картата изброявате отговорностите на класа в лявата страна и сътрудничества в дясната.

**20.Каква е разликата между Sequence диаграми и Collaboration диаграми?**

В Sequence диаграмата моделираме времето на живот на обекта, а в Collabroation диаграмата - структурните връзки между обектите по време на тяхното взаимодействие.

Разликата между Sequance и collaboration диаграма е в това, че в Sequance по вертикала се изобразява протичане на времето.

**21.Какви са двата принципа за разработка на interaction диаграмите?**

Sequance и Collaboration

**22.Какви диаграми се използват за представяне на жизнения цикъл на обектите от един клас?**

- Крайни автомати

**23.Един от документите на фазата „Анализ и дефиниция“ е речник. Какво е неговото предназначение?**

Речникът дефинира общата терминология, бива използван за потребителския интерфейс, help и ръководствтвото за потребителя.

**24.Защо фазата на „Анализ и дефиниция“ се разделя на две подфази?**

Разделянето на две фази е нужно, защото може да се стигне до неразбирателство с клиента още след първата подфаза - планирането(проучването), и съответно да се прекрати проекта. Ако не се раздели на две части и се окаже, че клиента не е съгласен, или че разработчиците не са съгласни и съответно проекта пропадне, това би било много изгубено време.

**25.На какви принципи трябва да отговарят основните концепции?**

1.Независими от фазата

2.Независими от вида на приложението

3.Независими от методите на развой.

**26.Кой е основният недостатък на COCOMO (Constructive Cost Model)?**

Основният недостатък на COCOMO е, че се базира на редове първичен код. Поради това то не е подходящ за Обектно-ориентирани проекти.

Трудната му ранна определеност и липсата на единно виждане за ред първичен код.

Броя редове код.

Няма ясна структурираност, съсредоточава се върху качеството на кода, а не върху целия продукт; тясна експериментална база.; не е подходящ за ОО.

**27.Кои са двата критерия, които влияят при определяне на цената на софтуера?**

Функционалност и качество

**28.Кои са основните концепции на функционалния изглед на системата?**

Функционални дървета, Диаграми на потока от данни и Use case диаграми (бизнес процес)

**29.Какви видове връзки има в ERD?**

Семантични връзки между множества от идентичности.

30.4.Кое от изброените не е принцип на доброто писане на код:

А) Създаване на unit тестове преди писане на кода;

Б) Създаване на визуално представяне на кода, за да бъде разбран;

В) Прилагане на refactoring след написване на кода;

Г) Всички тестове трябва да бъдат проследими до клиентските изисквания

Проучването може да е основа за договор, а дефиницията – за основа на проекта.

Един от документите на фазата „Анализ и дефиниция“ е речник. Какво е неговото предназначение?

Речникът дефинира общата терминология, бива използван за потребителския интерфейс, help и ръководствтвото за потребителя.

Защо фазата на „Анализ и дефиниция“ се разделя на две подфази?

Разделянето на две фази е нужно, защото може да се стигне до неразбирателство с клиента още след първата подфаза - планирането(проучването), и съответно да се прекрати проекта. Ако не се раздели на две части и се окаже, че клиента не е съгласен, или че разработчиците не са съгласни и съответно проекта пропадне, това би било много изгубено време.

Съдържание на спецификацията на изискванията – словесно описание на изискванията към продукта:

1.Функционални изисквания: функционалност, данни (логически изглед), потребителски интерфейс

2.Изисквания на средата на приложението: например потребителски профил

3.Технически изисквания: език за разработка, операционна система, хардуер

4.Изисквания за производителност: производителност, количество данни

5.Изисквания за валидност: разработване на тестове, тестови случаи

6.Изисквания за качество: удобен за използване интерфейст, надеждност...

7.Изисквания за реализация: модел на процеса, документация, срокове, цена

7 изисквания за качество:

1.Функционалност – удобство, прецизност, възможност за интеграция, съгласуваност, сигурност

2.Надеждност – зрелост, допустими грешки, възстановимост

3.Използваемост – разбираемост, разучаемост, оперируемост

4.Ефективност – поведение към време, поведение към ресурси

5.Възможност за поддръжка – възможност за анализ, възможност за промяна, стабилност, възможност за тестване

6.Преносимост

3 Специфични изисквания

•Има няколко различни организации (подхода) в тази секция зависещи от приложението.

•Независимо от избраната организация тази секция трябва да съдържа следната информация:

-изисквания за външния интерфейс

-функционални изисквания

-изисквания за производителност

-проектни ограничения

-критерии за качество

-други изисквания

Предварителна спецификация на изискванията:

1.основни функции на продукта, 2.основни данни, 3.обща производителност, 4.важни аспекти на потребителския интерфейс, 5.важни критерии за качество

Този документ е ориентиран към потребителя и трябва да бъде четен и разбран от него, т.е. той трябва да бъде написан в терминологията на потребителя (не в тези на разработчика).

Предварителна спецификация на изискванията – основната й задача е да представи основните функции и данни на проекта, производителността за продукта, основите на потребителския интерфейс, критериите за качество. Предназначен е за четене от потребителя. м/у клиент и разработчик

Определяне на цената – Целият процес за определяне на цената помага за избор на проект за реализация, определяне на екип и оценка на работата им. Критериите за определяне на цената според Боем са определеност, точност, обективност, детайлност, предсказуемост, икономичност, и др.

Метод на Боем (COCOMO – Constructive Cost Model)

Цел: Да се определи цена и срок на изработка за всеки проект

Основна идея: Да се използва броя редове първичен код

Същност: Метриката при COCOMO е човекомесец

Спецификация на изискванията – описва две основни групи изисквания: функционални и нефункционални. Предназначен е главно за разработчиците и трябва да бъде написан в тяхната терминология. Използва се като изходна точка на проектирането или разработването.

1. В диаграмите на потока от данни може ли да има потоци от данни, които нямат

имена? да

2. В кой от документите структурния анализ се различава от традиционния анализ:

модел на продукта;

3. В коя от фазите на разработка на софтуер се разработва entity-relationship модела:

фаза на дефиницията;

4. В коя от се разработват крайните автомати:фаза на дефиниция, фаза на проектиране;

5. В коя фаза от фазите на разработка на софтуер се разработват диаграмите на потока

от данни: фаза на дефиницията

6. Вярно ли е, че рамките са общи софтуерни архитектури с общи части за изпълнение,

за конкретен клас проблеми? да;

7. Вярно ли е, че рамките са система от повторно използваеми и приспособими класове

и приспособими клас библиотеки? да;

8. Дефиницията на продукта е основа за:проекта;

9. Диаграмите на потока от данни използват ли се при обектно-ориентиран анализ? не;

10. За всяка от характеристиките на всяко ниво на йерархичния модел на качество, се

определя теглова стойност, която се променя според: експертите, които ги оценяват;

11. Зависи ли избора на използваните основни концепции при софтуерната разработка

от приложната област на софтуера, който ще се разработва? Да

12. Кое от синтактичните правила за диаграмите на потока от данни (DFD) е грешно:

между външните обекти и хранилищата има директни потоци от данни;

13. Кои документи се разработват по време на фазата на планиране от процеса на

разработка на софтуер: речник, предварителна спецификация на изискванията,

определяне на цената и план на проекта;

14. Кои от изброените модели за определяне на цената на софтуер наподобяват

COCOMO: Doty, SPQR

15. Кои от моделите на развойния софтуерен процес са линейни последователни

модели: водопаден модел, V-модел, cleanroom – engineering;

16. Кои от моделите на развойния софтуерен процес са непоследователни модели:

инкрементален, прототипиране, фонтанен;

17. Кои рискове не се оценяват? всички рискове се оценяват без значение какви са;

18. Кои са документите, които се разработват по време на под-фазата: спецификация на

изискванията, модел на продукта, потребителски интерфейс, ръководство за потребителя,

речник;

19. Кои са основните концепции на ориентирания към алгоритми изглед на системата,

според Balzert: box диаграми, диаграми на програмния поток, псевдо-код;

20. Кои са основните концепции, според Balzert, на ориентирания към правила изглед

на системата: таблици на решенията, правила

21. Кои са основните концепции, според Balzert, на ориентирания към състояния изглед

на системата: activity диаграми, крайни автомати, мрежи на Петри;

22. Кои са основните концепции при разработка на софтуер според Balzert, използвани

във функционалния изглед на системата: функционални дървета, use-case диаграми,

диаграми на потока от данни;

23. Кои са основните продукти, които се получават във фазата разработка на софтуер?

source код + коментари + обектна програма + тестови план и протокол за тестове;

24. Кои са основните стъпки при анализ на изискванията:познаване на проблема, оценка

и синтез, моделиране, спецификация, преглед;

25. Кои са основните участници от софтуерния екип за разработка във фазата на

проектиране: ръководител на проект, проектант, системен анализатор, разработчик;

26. Кои са основните характеристики, които Боем дефинира в своя йерархичен модел

на качеството на софтуера за нивото на полезност на софтуера: надеждност, ефективност

и използваемост;

27. Кои са основните цели на обектно-ориентирания анализ? и двата отговора са верни

(Разработване на модел на продукта като часто от дефиницията му и разбработване на

модел на продукта за целите на разработването);

28. Кои са участниците от екипа за разработка на софтуер във фазата “разработване”:

ръководител на проект, проектант, разработчик;

29. Кой въвежда interaction диаграмите в разработката на обектно-ориентиран софтуер:

Ивар Якобсон

30. Кой от бизнес рисковете е свързан със създаването на софтуерен продукт, от който

никой няма нужда? стратегически риск;

31. Кой от моделите за определяне на цената на софтуера е подобен на

функционалните точки? BANG;

32. Кой от моделите за определяне на цената на софтуера е подобен на COCOMO? SPQR

& Doty;

33. Кой от факторите не влия върху софтуерната архитектура: всички изброени влияят;

34. Кой тип от крайните автомати е хибриден: автомат на Harel;

35. Коя е основната и най-важна концепция, която се използва в обектно-ориентирания

анализ: class диаграми;

36. Коя от основните концепции на функционалния изглед на системата е подходяща

при обектно-ориентиран развой? use case диаграми;

37. Коя от основните концепции показва жизнения цикъл на обектите? крайни автомати;

38. Коя от основните концепции представя алгоритъм за протичане на сценарии?

Activity диаграми;

39. Мини-спецификацията не може да бъде: речник от данни;

40. Могат ли use cases да бъдат описани чрез state-transition диаграми? не;

41. Могат ли use cases да бъдат описани чрез крайни автомати? да;

42. Могат ли в тестването на обектно-ориентиран софтуер да се използват класическите

интеграционни тестове – top-down и bottom-up? Не;

43. Могат ли таблиците на решения да се оптимизират? да;

44. На кое ниво от типичния йерархичен модел на качеството се намират оценъчните

елементи: Ниво 4.

45. На най-високо ниво в класификацията за описание на основните концепции за

разработка на софтуер се намират: нотационните форми;

46. На най-ниско ниво в класификацията за описание на основните концепции за

разработка на софтуер се намират: изгледите;

1. В диаграмите на потока от данни може ли да има потоци от данни, които нямат

имена? да

2. В кой от документите структурния анализ се различава от традиционния анализ:

модел на продукта;

3. В коя от фазите на разработка на софтуер се разработва entity-relationship модела:

фаза на дефиницията;

4. В коя от се разработват крайните автомати:фаза на дефиниция, фаза на проектиране;

5. В коя фаза от фазите на разработка на софтуер се разработват диаграмите на потока

от данни: фаза на дефиницията

6. Вярно ли е, че рамките са общи софтуерни архитектури с общи части за изпълнение,

за конкретен клас проблеми? да;

7. Вярно ли е, че рамките са система от повторно използваеми и приспособими класове

и приспособими клас библиотеки? да;

8. Дефиницията на продукта е основа за:проекта;

9. Диаграмите на потока от данни използват ли се при обектно-ориентиран анализ? не;

10. За всяка от характеристиките на всяко ниво на йерархичния модел на качество, се

определя теглова стойност, която се променя според: експертите, които ги оценяват;

11. Зависи ли избора на използваните основни концепции при софтуерната разработка

от приложната област на софтуера, който ще се разработва? Да

12. Кое от синтактичните правила за диаграмите на потока от данни (DFD) е грешно:

между външните обекти и хранилищата има директни потоци от данни;

13. Кои документи се разработват по време на фазата на планиране от процеса на

разработка на софтуер: речник, предварителна спецификация на изискванията,

определяне на цената и план на проекта;

14. Кои от изброените модели за определяне на цената на софтуер наподобяват

COCOMO: Doty, SPQR

15. Кои от моделите на развойния софтуерен процес са линейни последователни

модели: водопаден модел, V-модел, cleanroom – engineering;

16. Кои от моделите на развойния софтуерен процес са непоследователни модели:

инкрементален, прототипиране, фонтанен;

17. Кои рискове не се оценяват? всички рискове се оценяват без значение какви са;

18. Кои са документите, които се разработват по време на под-фазата: спецификация на

изискванията, модел на продукта, потребителски интерфейс, ръководство за потребителя,

речник;

19. Кои са основните концепции на ориентирания към алгоритми изглед на системата,

според Balzert: box диаграми, диаграми на програмния поток, псевдо-код;

20. Кои са основните концепции, според Balzert, на ориентирания към правила изглед

на системата: таблици на решенията, правила

21. Кои са основните концепции, според Balzert, на ориентирания към състояния изглед

на системата: activity диаграми, крайни автомати, мрежи на Петри;

22. Кои са основните концепции при разработка на софтуер според Balzert, използвани

във функционалния изглед на системата: функционални дървета, use-case диаграми,

диаграми на потока от данни;

23. Кои са основните продукти, които се получават във фазата разработка на софтуер?

source код + коментари + обектна програма + тестови план и протокол за тестове;

24. Кои са основните стъпки при анализ на изискванията:познаване на проблема, оценка

и синтез, моделиране, спецификация, преглед;

25. Кои са основните участници от софтуерния екип за разработка във фазата на

проектиране: ръководител на проект, проектант, системен анализатор, разработчик;

26. Кои са основните характеристики, които Боем дефинира в своя йерархичен модел

на качеството на софтуера за нивото на полезност на софтуера: надеждност, ефективност

и използваемост;

27. Кои са основните цели на обектно-ориентирания анализ? и двата отговора са верни

(Разработване на модел на продукта като часто от дефиницията му и разбработване на

модел на продукта за целите на разработването);

28. Кои са участниците от екипа за разработка на софтуер във фазата “разработване”:

ръководител на проект, проектант, разработчик;

29. Кой въвежда interaction диаграмите в разработката на обектно-ориентиран софтуер:

Ивар Якобсон

30. Кой от бизнес рисковете е свързан със създаването на софтуерен продукт, от който

никой няма нужда? стратегически риск;

31. Кой от моделите за определяне на цената на софтуера е подобен на

функционалните точки? BANG;

32. Кой от моделите за определяне на цената на софтуера е подобен на COCOMO? SPQR

& Doty;

33. Кой от факторите не влия върху софтуерната архитектура: всички изброени влияят;

34. Кой тип от крайните автомати е хибриден: автомат на Harel;

35. Коя е основната и най-важна концепция, която се използва в обектно-ориентирания

анализ: class диаграми;

36. Коя от основните концепции на функционалния изглед на системата е подходяща

при обектно-ориентиран развой? use case диаграми;

37. Коя от основните концепции показва жизнения цикъл на обектите? крайни автомати;

38. Коя от основните концепции представя алгоритъм за протичане на сценарии?

Activity диаграми;

39. Мини-спецификацията не може да бъде: речник от данни;

40. Могат ли use cases да бъдат описани чрез state-transition диаграми? не;

41. Могат ли use cases да бъдат описани чрез крайни автомати? да;

42. Могат ли в тестването на обектно-ориентиран софтуер да се използват класическите

интеграционни тестове – top-down и bottom-up? Не;

43. Могат ли таблиците на решения да се оптимизират? да;

44. На кое ниво от типичния йерархичен модел на качеството се намират оценъчните

елементи: Ниво 4.

45. На най-високо ниво в класификацията за описание на основните концепции за

разработка на софтуер се намират: нотационните форми;

46. На най-ниско ниво в класификацията за описание на основните концепции за

разработка на софтуер се намират: изгледите;

47. Определянето на стойностите на оценъчните елементи в йерархичния модел за

качество по традиционния модел става като: стойностите се определят от експерти;

48. Основните концепции на ориентирания към данни изглед на системата, според

Balzert са:речници на данните, entity-relationship модел;

49. Основните концепции, описващи трябва да отговарят на следните принципи:

независими от фазите, приложението и метода на развой & контекста;

50. Отговорността за създаване на документите предварителна спецификация на

изискванията и речник на продукта е на: приложния специалист;

51. Отговорността за създаване на модел на продукта е на: системния анализатор;

52. По време на коя от фазите се разработват правилата? фаза на дефиниция;

53. По време на коя от фазите се разработват таблиците на решения? Няма верен

отговор (Не са: фази на пректиране и разработка, фази на проектиране и дефиниция, фаза

на проектиране); Верен отговор – Фази на дефиниция и разработка;

54. По какво се различава тестването на обектно-ориентиран софтуер от тестването на

ковенционален софтуер?по цялостната стратегия за тестване;

55. При кой от крайните автомати действията са прикрепени към преходите? на Mealy;

56. При кой от крайните автомати действията са прикрепени към състоянията? на Moor;

57. Прототипирането на софтуер може да бъде: open-ended и close-ended;

58. Речниците от данни използват ли се в обектно-ориентирания анализ? да;

59. С над колко условия се смята, че таблицата на решения става неясна:с над 5;

60. Спецификацията на изискванията е документ, състоящ се от следните основни

части:функционални изисквания, изисквания на средата на приложението, технически

изисквания, изисквания за производителност, изисквания за валидност, изисквания за

качество, изисквания за реализация;

ОТВОРЕНИ

61. Дайте дефиниция за документа предварителна спецификация на изискванията?

Предварителна спецификация на изискванията - основни функции, основни данни, обща

производителност, важни аспекти на потребтелския интерфейс, важни критерии за

качество

62. Дайте дефиниция за документа речник на продукта? Дефинира общата

терминология. Служи за потребителския интерфейс, help и ръководството за

потребителите. Синтактична структура на данните от потребителска гледна точка.

63. Дайте дефиниция за метрика? Свойството на програмния продукт да бъде оценен с

някаква обективна мярка;

64. За какво служи COCOMO (Constructive Cost Model) модела и каква е неговата

същност? Целта му е за всеки планиран софтуерен продукт да се оцени цената и срокът на

разработване,основната идея е използването на броя редове първичен код;

65. За решаване на какви проблеми е подходяща структура на екип с децентрализиран

контрол: на прости проблеми и сложни проблеми защото при малки проекти ще затрудни

разработката но при големи може проджект менажера да решава кой модел да използва

dd,cd или cc в зависимост от стила,броя и нивото на екипа,декомпозиране евентуално

може да има в две области функционалност, процес;

66. Защо activity диаграмите могат да се причислят към алгоритмичния изглед на

системата? Те представляват вариант на крайните автомати.Описват бизнес процеса чрез

състояния на действията;

67. Защо е подходящо използването на рамки (frameworks) в обектно-ориентираното

проектиране? Заради повторното използване на код;

68. Избройте десетте критерия, на които трябва да отговаря един модел за определяне

на цената на софтуера според Боем? Определеност, точност, обективност, детайлност,

устойчивост, област на приложение, конструктивност, простота на прилагане,

предсказуемост, икономичност

69. Как от обектно-ориентиран аналитичен модел се получава обектно-ориентиран

проектен модел? Разширен, модифициран, оптимизиран, адаптиран към средата. Същите

механизми за описание: Клас диаграми.

70. Какви са двете разширения на COCOMO (Constructive Cost Model) модела? -

въвеждането на три типа софтуерни проекти:разпространен, полунезависим, вграден и

преминаване към уравнение, в което се отчитат и допълнителни фактори освен редове

първичен код

71. Какво е идентичността в Entity-Relationship модела? Индивидуално и различно

представяне на обекти от реалния или абстрактен свят.

72. Какво е множеството от идентичности в Entity-Relationship модела? множество от

идентичности с еднакви характеристики (атрибути)

73. Какво е развоен софтуерен процес? Цялостният процес на поставяне на задачата,

планиране, реализиране и оценка на едно софтуерно и хардуерно приложение,

включително и използваните помощни средства, методи и необходимия персонал.

74. Какво е софтуерен инженеринг?Дисциплина, която интегрира процеси, методи и

средства за разработване на софтуер.

75. Какво означава добра софтуерна архитектура? Слабо свързване + силна

съгласуваност;

76. Какво описват концепциите от ориентирания към сценарий изглед на системата?

Описват последователност от съобщения между актьори и обекти на системата.

77. Какво представлява big-bang подхода при интеграционно

тестване?Неинкрементална интеграция

78. Какво представлява процесната метрика инфлация на услията?Метриката инфлация

на усилията в процеса на разработка на софтуер е :отношение между усилията в

ускорения процес на предвидените действия в нормални условия.отношение между

условията в ускорения процес и предвидените нормални условия

79. Какво представлява софтуерната архитектура? Състои се от системни компоненти и

връзки между тях.

80. Какво представляват дърветата на решенията? Основна концепция на ориентирания

към правила изглед на с-мата (хоризонтална подредба на правила)

81. Какво се получава като продукт след приключване на фазата на проектиране?

Софтуерна архитектура и спецификация на софтуерните компоненти

82. Кои диаграми се разработват в динамичния модел на обектно-ориентирания

анализ? Крайни автомати; activity диаграми; sequence диаграми; collaboration диаграми;

use-case диаграми;;

83. Кои основни концепции, според Balzert, спадат към ориентирания към сценарий

изглед на системата?

Sequence диаграми, Collaboration диаграми

84. Кои са двата основни модела, които се разработват по време на обектно-ориентиран

анализ? Статични и динамични.

85. Кои са недостатъците на структурното проектиране? Интерфейсите и хранилищата не

могат да се усъвършенстват. Методите за описание на софтуера се променят при прехода

от анализ към проектиране

86. Кой е основният недостатък на структурното проектиране?

софтуерната архитектура се състои от йерархия от функционални компоненти

(подпрограми), а такива архитектури не се срещат често.

87. Кои са нивата на типичния йерархичен модел на качество?

Ниво 0 – Качество

Ниво 1 - фактори, представящи даден аспект на качеството на софтуера от гледна точка на

потребителя.

Ниво 2 - критерии – представят характеристики на програмния продукт.

Ниво 3 - метрики –детайли на даден критерии. Те се определят от оценъчните елементи.

Ниво 4 – оценъчни елементи – елементарни характеристики на най-ниско ниво, които

подлежат на количествена оценка.

88. Кои са основните концепции, използвани в структурното проектиране?

Функционални дървета, DF диаграми, DD, псевдо код, дъревта на решенията и таблица на

решенията.

89. Кои са основните концепции, които се използват при структурен анализ?

Функционални дървета, dataflow диаграми, data dictionaries, таблици на решенията,

дървета на решенията и псевдокод.

90. Кои са основните типове тестове, които се провеждат в една стратегия за тестване?

Единични, интеграционни, валидационни, системни

91. Кои са основните фази на жизнения цикъл на програмния продукт? Жицнен цикъл на

програмния продукт - целия период на неговото изпозлване и създаване ; начало - възникване на идеята за създаване на продукта; край - моментът в който се

преустановява изпозлването му

92. Кои са основните цели, които си поставя Олбрихт при създаване на метода на

функционалните точки за определяне на цената на софтуера?

-да се използват външните характеристике на софтуера

-да може да се прилага в ранни фази на производствения процес

-за може да се свърже лесно с икономическа оценка

-да има независимост от редовете първичен код

93. Кои са основните цели на дисциплината Софтуерни технологии? Лесна поддръжка,

надеждност на софтуера, ефективност на софтуера, удобен и лесен за ползване, цена.

94. Кои са под-фазите в обектно-ориентираното проектиране? Проект на

архитектурата,Проект на разработката;

95. Кои са под-фазите, на които се разделя фазата на анализ и дефиниция според

Balzert? Фаза на планиране, Фаза на дефиницията

96. Кои са тестовете, които се провеждат при приемане на софтуера?Валидационни

тестове – alpha и beta

97. Към кои от основните типове тестове принадлежат stress тестовете? Системни

тестове

98. Опишете partition тестовете? Категоризират се входовете и изходите от операциите на

класовете и тестовите случаи се разработват за изпълнение на всяка категория

99. Опишете random тестовете? Тестването включа произволни вариации на

последователности на операциите.

100. Опишете top-down интеграционните тестове? Модулите се интегрират посредством

преместване надолу през контролната йерархия, като се започне от главния контролен

модул – главната програма (maincontrolmodule – mainprogram). Модулите, подчинени на

главния контролен модул, се интегрират в структурата по един от двата начина.

Първо в дълбочина (depth-first)

Първо в широчина (breadth-first)

Главният контролен модул се използва като тестови драйвер, а всички компоненти,

директно подчинени на него, се заместват временно със стубове (stubs).

В зависимост от избрания подход за интеграция заместващите стубове (subs) се заместват

последователно и поотделно от съответните актуални компоненти.

Тестовете се направляват, с интеграцията на всеки компонент.

При завършване на всяко множество от тестове, следващ стуб (stub) се замества от реален

компонент.

Регресионното тестване може да се използва за осигуряване, че няма да се въведат нови

грешки.

Процесът продължава във втора стъпка, докато се създаде цялата програма.

101. Опишете процеса на провеждане на unit тестове(при конвенционален софтуер)?

Съсредоточават се върху най-малката единица на софтуера – софтуерните компоненти

или модели;

102. Опишете същността на COCOMO модела на Боем:За изчисляване на усилията се

използва формулата:ЧМ=2.4\*ХРПК^1.05,ЧМ-човекомесеци,ХРПК-хиляди реда първичен

код,за оценка на продълителността на разработване на софтуерният проект формулата

е:B=2.5\*ЧМ^0.38,където B есрокът за разработване в месеци.

103. Основа за кой модел, при разработката на софтуер, е entity-relationship

модела?Data-orientedview(ориентирания към данни изглед)

104. С каква цел се разработват package диаграмите?Показват модулация на

разработваната система. Пакетът съдържа компоненти за големи единици, като може и

да съдържа други пакети/ управление на семинари.

105. Кои от изброените модели за определяне на цената на софтуер наподобяват

COCOMO:

Doty, SPQR

106. Кои са основните концепции, според Balzert, на ориентирания към правила изглед

на системата:

таблици на решенията, правила

107. Кой от факторите не влия върху софтуерната архитектура:

Език за разработка, целева платформа и функционални изисквания (Всички са верни);

108. Кои са основните концепции, използвани в структурното проектиране?

Структурните диаграми и спецификация на модулите

109. Кой тип от крайните автомати е хибриден:

автомат на Harel;

110. Опишете процеса на провеждане на unit тестовете?

Unit тестове – съсредоточават се върху най-малката единица на софтуера – софтуерните

компоненти или модули.

111. Кой е основният недостатък на структурното проектиране?

Софтуерната архитектура се състои от йерархия от функционални копоненти, а такива

архитектури не се срещат често;

112. Какви тестове се провеждат при unit тестовете (при конвенционален софтуер)?

Съсредоточават се в/у най-малката единица на софтуера – софтуерните компоненти или

модели.

113. Опишете (COCOMO) модела на Боем за определяне на цената на софтуера и кажете

кои са неговите разширения?

ЦЕЛ: за всеки планиран софтуерен проект да се оцени цената и срокът на разработване.

Същност- за изчисляване на усилията се използва форумалта ЧМ = 2.4\* ХРПК^1.05, ЧМ-

брой човекомесеци ; ХРПК - хиляди реда първичен код, за Оценка на продължителността

В=2.5\*ЧМ^0.28 В-срок на разработванетo.

РПК се броят без коментарите, принадлежат на крайният продукт, не включват

използваните стандартни програми; Вклюват се фазите на проектиране, програмиране и

оценка включително усилипта по управление и докумнетирането 1 ЧМ = 19 дни или 152

часа... не се правят сериозно промени в спецификацията и изискванията след

одобрението и.

114. Коя е основната концепция с най-важно значение в обектно-ориентирания анализ?

Class Диаграмите;

115. Кои са подфазите на фазата на проектиране според Balzert?

проект на архитектурата и проект на разработката

116. Кои са недостатъците на структурния анализ?

Интерфейсите на могат да бъдат усъвършенствани, хранилищата не могат да бъдат

усъвършенствани, трансормирането на модела на структурния анализ в модел на

структурно проектиране води до промяна на използваните основни концепции.

117. Кои от моделите за определяне на цената, при разработка на софтуер, не са

приложими при обектно-ориентиран софтуер:

модели свързани с броя редове код;

118. Кой е основният недостатък на структурното проектиране?

Софтуерната архитектура се състои от йерархия от функционални копоненти, а такива

архитектури не се срещат често

119. Кои основни концепции спадат според Balzert към ориентирания към сценарии

изглед на системата? описва последователност от съобщения между актьори и обекти на

системата. Използват се Sequence и Collaboration диаграми.

120. Кои са основните концепции, които се ползват в структурния анализ?

функционални дървета, data flow диаграми, data dictionary, таблица на решенията,

дървета на решенията, псевдо код.

121. Кои са основните концепции, използвани в стуктурното проектиране?

структурните диаграми

122. Кой от факторите не влияе върху софтуерната архитектура?

всички изброени влияят

123. Как от обектно-ориентирани аналитични модели се получават обектно-

ориентирани проектни модели? Като се разшири ООА с класове от потребителския

интерфейс.

124. Спецификацията на изискванията е документ, състоящ се от следните основни

части:

функционални изисквания, изисквания на средата на приложението, технически

изисквания, изисквания за производителност, изисквания за валидност, изисквания за

качество, изисквания за реализация;

125. Кои от моделте за определяне на цената, при разработка на софтуер, не са

приложени при обектно ориентирания софтуер?

Модели свързани с броя на редовете код

126. Каква е идеята за модела на качество на софтуера при типичния йерархичен

модел?

Разглеждане на качеството като йерархична структура.

127. Каква е разликата между двете методологии за оценяване на софтуерния процес

CMM и BOOTSTRAP?

Разликата е, че BOOTSTRАР за разлика от СММ се основава и на стандартите за качество

ISO 9000, 9001 и 9001-3

128. Какво представлява модела на процеса в BOOTSTRAP?

Аналогично на СММ има нива на зреост:

- ниво 0 – непълен процес;

- ниво 1 – изпълняван процес;

- ниво 2 – управляван процес;

- ниво 3 – установен процес;

- ниво 4 – предсказуем процес;

- ниво 5 – оптимизиращ поцес;

129. Какви са недостатъците на софтуерното проектиране? – използването на напълно

нова концепция за програмиране.

130. Какво представлява критерият конструктивност на моделите за установяване на

цената според Боем? - Всеки модел трябва да дава възможност да бъдат анализирани и

разбрани получените резултати. Моделът че допуска получаването на различни

резултати в зависимост от това на какъв фактор потребителят придава по-голямо

значение.

131. Какво представлява стратегията за работа със софтуерния риск? – Управлението и

анализа на риска е поредица от стъпки, които помагат на софтуерния тим да управлява

несигурността при разработка на един софтуерен продукт.

132. Кой от моделите за определяне на цената на софтуера е подобен на COCOMO?-Doty

и SPQR

133. Кой от моделите за определяне на цената на софтуера е подобен на

функционалните точки? - BANG

134. Кои са десетте основни критерия на моделите за установяване на цената според

Боем?-определеност, точност, обективност, детайлност, устойчивост, област на

прриложение, конструктивност, простота на прилагане, предсказуемост, икономичност.

135. Кои са основните недостатъци на модела на Боем за качество на софтуера?

Повторното използваемия код.

136 . Кои са основните предимства на модела на функционалните точки създаден от

Олбинт?

Основа на модела е понятието „функционална точка”. Идеята му е, че усилията в

разработка на даден софтуер се определят от неговата функционалност, която се измерва

с „фунционални точки”.

137. Посочете поне една процесна метрика? – усилия, продължителност, персонал, цена,

приходи, продуктивност, инфлация, бързина, ускорение.

138. Функционални дървета и диаграми на потока от данни –структурен анализ;

139. Диаграми на сценарии – обектно-ориентиран анализ;

140. Какво е метрика? - Всяко свойство за конкретния програмен продукт да се оцени с

някаква обективна мярка. Тази мярка се нарича метрика.

141. Каква е разликата между обектно-ориентирания анализ (ООА) и структурния

анализ в документите от „Анализ и дефиниция”? – В моделите

142. При ООА (задължително) се разработват 2 вида модели – статичен и динамичен.

143. Кога свършва живота на един софтуерен продукт? – Когато морално остарее.

144. Къде е мястото на Use case диаграмите? – в статичния и в динамичния модел.

145. Какво е софтуерна архитектура? – системни компоненти и връзката между тях.

146. Какъв е недостатъка на структурния анализ и структурното проектиране?

При обектно-ориентираното проектиране основна концепция – клас диаграмата –

основна софтуерна архитектура.

147. Какво трябва да се коментира?

- започване на нова програмна единица;

- ролята на типовете данни и променливите;

- започващ код;

148. Кога приключват тестовете?

- никога не приключват;

- когато свършат парите и времето;

- използвайки статистическо моделиране;

149. Може ли да се тестват не зависими модули? – Теорията дава решение – предлага

Overhead програмиране;

150. При Unit тестовете се тестват вече класове