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