

## 简答题

### 1. what are the content of user interface analysis?请详细描述

1. 用户分析:要想让用户的心理意象和设计模型趋同,唯一的办法就是努力去理解用户,
2. 任务分析和建模:

#### 12.4.3 Task Analysis and Modeling

- Use-cases
  - define basic interaction
- Task elaboration
  - refines interactive tasks.
  - Task analysis: 1-Understand the tasks that people currently perform (when use a manual approach)
  - Task analysis: 2-study an existing specification for a computer-based solution and derive a set of user tasks that will accommodate the user model, the design model...
- Object elaboration (相当于用户接口类的设计与细化)
  - identifies interface objects (classes)
- Workflow analysis
  - defines how a work process is completed when several people (and roles) are involved (UML swimlane)
- Hierarchical representation (层次表示)
  - A task hierarchy can be defined for each user type. The hierarchy is derived by stepwise elaboration of each task

3. 显示内容分析:数据可视化文字、表格、图形、音频、视频等
4. 工作环境分析:有时,设计会受到某些因素的约束和限制,这些因素会减弱易用性(如:空间站,机舱等)

### 2. Why are incremental process models considered by many to be the best approach to software development in a moden context?

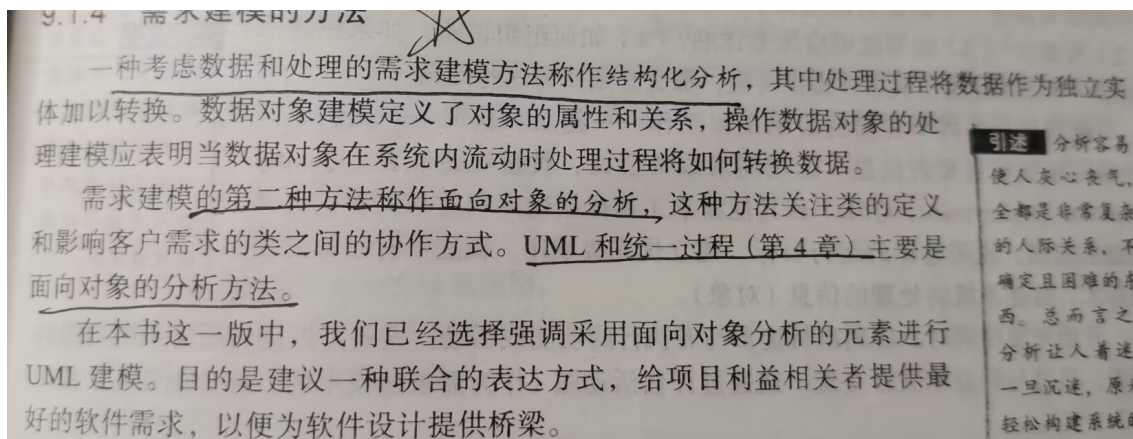
QA: 由于现代软件开发的时间线越来越短,客户越来越多样化(对需求的理解更加困难),对需求的变更也越来越多,因此整个开发过程不宜单纯采用线性模型。我们需要一种方式提供快速增量或者进行演化交付,即**能迫切需要**为用户迅速提供一套功能有限的软件产品,然后在后续版本中不断细化和扩展功能。增量模型比大多数过程模型能更好的适应不确定性,允许有序和有计划的方式交付**部分解决方案**。

### 3. How do software characteristics differ from hardware characteristics?

#### 软件硬件区别:

1. 软件是被开发的或者设计的,它不是经典意义上被制造出来的
2. 随着时间的推移,因为尘土、振动、使用不当、温度超限制、以及其他一些环境问题,硬件效率会逐渐降低,即硬件会**磨损**  
而软件不会磨损,但会退化,由于软件生命周期,会面临许多变更,每次变更都可能引入新的错误,使得是失效率像实际曲线那样陡然上升。可以说,**不断地变更时软件退化的根本原因**。
3. 尽管该行业正在转向基于组件的构建,但大多数软件仍然是**定制的 (custom-built)**。

### 4. Briefly describe the primary differences between structured analysis and object-oriented analysis.



5. how do you create agile processes to manage unpredictability?

3. How do you create agile processes to manage unpredictability ?

- a. Requirements gathering must be conducted very carefully
- b. Risk analysis must be conducted before planning takes place
- c. Software increments must be delivered in short time periods
- d. Software processes must adapt to changes incrementally
- e. Both c and d

【e】

6. Describe the differences between software construction and software deployment.

软件构造涉及软件增量的编码和测试。部署涉及到将操作软件产品交付给最终用户，在活跃期间使用支持产品，并处理有关产品有用性的反馈。

7. List the four design models required for a complete specification of a software design and the role of each.

- 场景模型：出自各种系统“参与者”观点的需求from user point of view(有用例图、活动图、顺序图)
- 面向类的模型：表示面向对象类（属性和操作）的模型，其方式为通过类的协作获取系统需求。(类图，协作图)
- 基于行为和模式的模型：描述如何将软件行为看作外部事件后续的模式（impact of event）(时序图，状态图)
- 面向流模型：描述问题信息域的模型。(数据流图，数据模型)

8. What is software architectural style? Give at least four software architectural styles

### 13.3 体系结构风格

每种体系结构风格描述包含：

- 1. 完成系统需要的某种功能的一组组件(如数据库、计算模块)
- 2. 能够使构件间实现通信、协作的一组连接件
- 3. 定义构建如何成为系统的约束
- 4. semantic models 语义模型，能够使设计者通过分析系统组成成分的已知属性来理解系统的整体性质。

- 以数据为中心体系结构
- 数据流体系结构
- 调用和返回体系结构
- 面向对象体系结构
- 层次结构

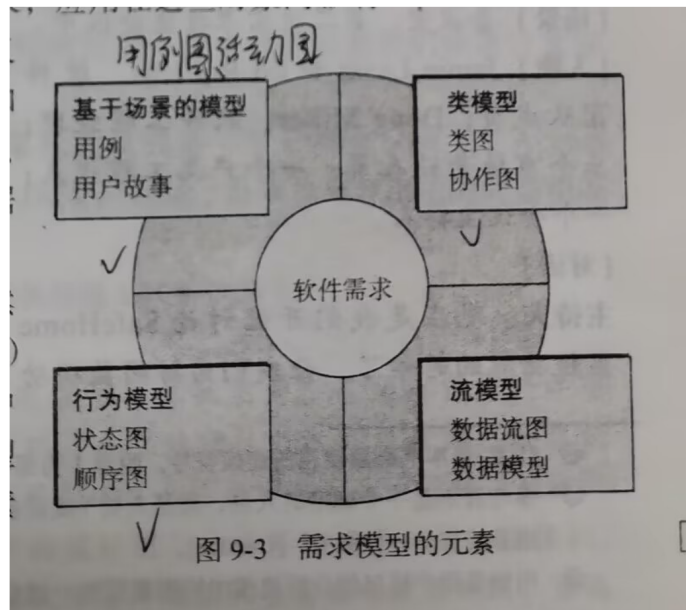
9. Describe the roles of the three sections of CRC (class responsibility collaborator) cards?

- 1. 类名唯一标识数据对象。(3分)
- 2. 职责是类的属性和操作。（3分）
- 3. 合作者是那些需要向班级提供完成责任所需的信息的类。(4分)

10. Describe the four types of models for requirements modeling and list at least two detail modeling approaches for each.

- 场景模型：出自各种系统“参与者”观点的需求from user point of view(有用例图、活动图、顺序图)
- 面向类的模型：表示面向对象类（属性和操作）的模型，其方式为通过类的协作获取系统需求。(类图，协作图)

- 基于行为和模式的模型：描述如何将软件行为看作外部事件后续模型 (impact of event) (时序图，状态图)
- 面向流模型：描述问题信息域的模型。(数据流图，数据模型)



11. Explain what is wrong with the notion that computer software does not need to evolve over time.

为什么必须改变

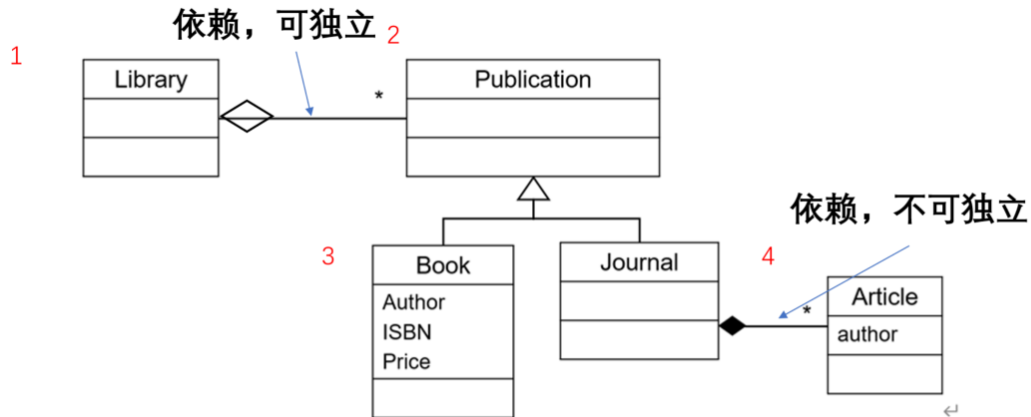
- 1、软件需要进行适应性调整，从而可以满足新的计算环境或者技术的需求
- 2、软件必须升级以实现新的商业需求
- 3、软件必须扩展以使之具有与更多新的系统和数据库的互操作能力

4、软件架构必须进行改建以使之能适应不断演化的计算环境

- Software must be **adapted to** meet the needs of new computing environments or technology.
- Software must be **enhanced** to implement new business requirements.
- Software must be **extended** to make it interoperable with other more modern systems or databases. (软件必须扩展使之具有与更多新的系统和数据库的互操作能力)
- Software must be **re-architected** to make it viable within a network environment. (软件架构必须进行改建使之能适应不断演化的计算环境)

12. According to the following description, draw the corresponding class diagram. A library has many publications. These publications include books and journals. Each book has information such as author, ISBN number, price, etc. A journal contains multiple articles, each with authors.

∴ A journal contains multiple articles, each with authors.



13. Someone says: Until I get the program running, I have no way of assessing its quality. Is that right? Why?

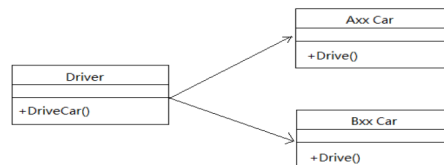
答案:

这种说法不对。(---2 分)

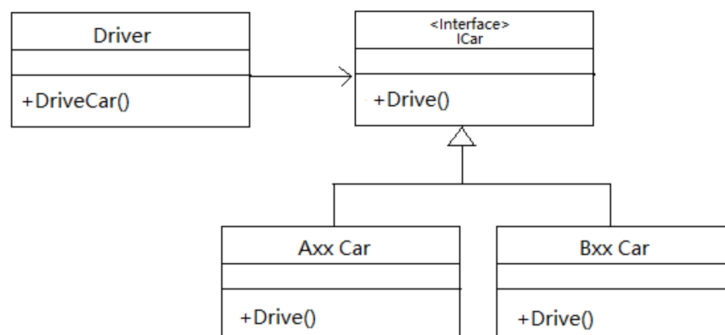
(1) 有数据分析显示: 需求阶段引入的缺陷占比最高, 其次是设计, 如果不在早期发现修复这些缺陷, 则会导致修复费用的飙升, 并影响软件的质量和交付时机, 所以不能等到后期才开始进行质量评估。(---3 分)

(2) 软件工程中提供有手段可以在项目的早期进行质量评估活动, 即“技术评审”, 该方法被称之为“质量过滤器”, 是比动态测试更为有效的一种发现缺陷的质量保证手段, 它可以发现需求和设计的缺陷。(---3 分)

14. 1. What does OCP (Open-Closed Principle) stand for?(4 分) Please modify the following design model to avoid violation of OCP. (6 分)



QA:构件对修改具有封闭性, 对扩展具有开放性



15. Descriptions of software testing has to go through several stages,each stage with what documentation.

软件测试之后是单元测试、集成测试、验证测试和系统测试  
与集成测试相关的文档是集成测试计划和软件架构设计文档(主)设计文档  
与验证测试相关的文档包括验证测试计划和软件需求规范。  
与系统测试相关的文档有系统测试计划、系统需求规范、用户手册和安装手册。

16. 什么用例? 有什么作用?

QA:

#### 5. 用例

用例是一个软件工术语,描述参与者如何使用系统来完成特定的目标。用例充当软件建模技术,定义要实现的特性以及可能遇到的任何错误的解决方案。

17. 什么是回归测试?

QA:回归测试指的是测试代码中被修改的部分以及可能因为修改而受到影响的的部分的过程,以确保在修改完成后,软件中没有引入新的错误。回归意味着某种东西的返回,在软件工领域,他是指bug的回归。

18. 什么是QFD

#### 3.QFD(质量功能部署)

质量功能部署(QFD)是一种将客户需求转换为软件技术需求的质量管理技术。QFD专注于从软件工过程中最大化客户满意度。

19. 什么是原型模型(prototyping)模型?

#### 4. 原型模型

原型模型是软件开发生命周期模型之一。当客户事先不知道确切的项目需求时,使用此模型。在这个模型中,首先开发最终产品的原型,根据客户的反馈反复测试和改进,直到获得最终可接受的原型,这形成了开发最终产品的基础。