

第 6 章开发方法

1. 【2011 年题 24 解析】

本题考查开发相关的一系列知识。B 选项中“自底向上地使用结构化方法开发系统”显然是错误的,因为结构化方法的一个核心特色为:“自顶向下,逐步求精”,而非自底向上。

2. 【2013 年题 21 解析】

逆向工程导出的信息可分为如下 4 个抽象层次。

实现级:包括程序的抽象语法树、符号表等信息。

结构级:包括反映程序分量之间相互依赖关系的信息,例如调用图、结构图等。

功能级:包括反映程序段功能及程序段之间关系的信息。

领域级:包括反映程序分量或程序与应用领域概念之间对应关系的信息。

【答案】A、B。

3. 【2017 年题 29 解析】

所谓软件的逆向工程就是分析已有的程序,寻求比源代码更高级的抽象表现形式。一般认为,凡是在软件生命周期内将软件某种形式的描述转换成更为抽象形式的活动都可称为逆向工程。与之相关的概念是:重构(restructuring),指在同一抽象级别上转换系统描述形式;

设计恢复(design recovery),指借助工具从已有程序中抽象出有关数据设计、总体结构设计和过程设计的信息(不一定是原设计);

再工程(re-engineering),也称修复和改造工程,它是在逆向工程所获信息的基础上修改或重构已有的系统,产生系统的一个新版本。

【答案】A、D。

6.1 软件生命周期

1. 【2011 年题 29 解析】

在软件开发中,外部设计又称为概要设计,其主要职能是设计各个部分的功能、接口、相互如何关联。内部设计又称为详细设计,其主要职能是设计具体一个模块的实现。所以本题应选 A。

2. 【2019 年题 16 解析】

本题考查的是系统开发基础知识中关于软件生命周期的内容。

按照传统的软件生命周期方法学,可以把软件生命周期划分为软件定义、软件开发、软件运行与维护三个阶段。第一空选择 A 选项。

软件定义包括可行性研究和详细需求分析过程,任务是确定软件开发工程必须完成的总目标。第二空选择 A 选项。

软件开发时期就是软件的设计与实现,可分为概要(总体)设计、详细设计、编码、测试等。

软件运行就是把软件产品移交给用户使用。软件投入运行后的主要任务是使软件持久满足用户的要求。

软件维护是对软件产品进行修改或对软件需求变化做出响应的过程,也就是尽可能地延长软件的寿命。

当软件已没有维护的价值时,宣告退役,软件生命随之宣告结束。

6.2 软件开发模型

1. 【2010 年题 25 解析】

原型模型又称快速原型。原型模型主要有两个阶段：① 原型开发阶段。软件开发人员根据用户提出的软件系统的定义，快速地开发一个原型。该原型应该包含目标系统的关键问题和反映目标系统的大致面貌，展示目标系统的全部或部分功能、性能等。② 目标软件开发阶段。在征求用户对原型的意见后对原型进行修改完善，确认软件系统的需求并达到一致的理解，进一步开发实际系统。

瀑布模型可以说是最早使用的软件生存周期模型之一。由于这个模型描述了软件生存的一些基本过程活动，所以它被称为软件生存周期模型。这些活动从一个阶段到另一个阶段逐次下降，形式上很像瀑布。瀑布模型的特点是因果关系紧密相连，前一个阶段工作的结果是后一个阶段工作的输入。

螺旋模型是在快速原型的基础上扩展而成的。这个模型把整个软件开发流程分成多个阶段，每个阶段都由 4 部分组成，它们是：① 目标设定。为该项目进行需求分析，定义和确定这一个阶段的专门目标，指定对过程和产品的约束，并且制定详细的管理计划。② 风险分析。对可选方案进行风险识别和详细分析，制定解决办法，采取有效的措施避免这些风险。③ 开发和有效性验证。风险评估后，可以为系统选择开发模型，并且进行原型开发，即开发软件产品。④ 评审。对项目进行评审，以确定是否需要进入螺旋线的下一次回路，如果决定继续，就要制定下一阶段计划。

V 模型是一种典型的测试模型。在 V 模型中测试过程被加在开发过程的后半部分，分别包括单元测试、集成测试、系统测试和验收测试。

【答案】C。

2. 【2011 年题 30 解析】

原型开发分两大类：快速原型法(又称抛弃式原型法)和演化式原型法。其中快速原型法是快速开发出一个原型，利用该原型获取用户需求，然后将该原型抛弃。而演化式原型法是将原型逐步进化为最终的目标系统。所以本题应选 C。

3. 【2012 年题 23 解析】

快速应用开发(Rapid Application Development, RAD)是一种比传统生存周期法快得多的开发方法，它强调极短的开发周期。RAD 模型是瀑布模型的一个高速变种，通过使用**基于构件的开发方法**获得快速开发。如果需求理解得很好，且约束了项目范围，利用这种模型可以很快地开发出功能完善的信息系统。

但是 RAD 也具有以下局限性：

①、并非所有应用都适合 RAD。RAD 对模块化要求比较高，如果有哪一项功能不能被模块化，那么 RAD 所需要的构建就会有问题；如果高性能是一个指标，且该指标必须通过调整接口使其适应系统构件才能获得，则 RAD 也有可能不能奏效。

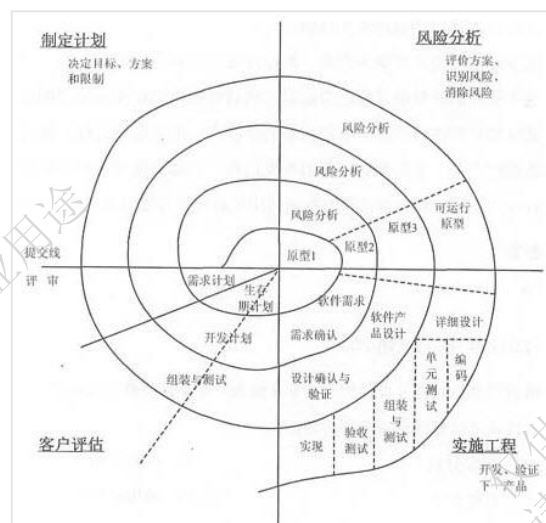
②、开发者和客户必须在很短的时间完成一系列的需求分析，任何一方配合不当，都会导致 RAD 项目失败。

③、RAD 只能用于管理信息系统的开发，不适合技术风险很高的情况。例如，当一个新系统要采用很多新技术，或当新系统与现有系统有较高的互操作性时，就不适合使用 RAD。

【答案】D、C。

4. 【2012 年题 25 解析】

螺旋模型是在快速原型的基础上扩展而成的一种生存周期模型。这种模型将整个软件开发流程分成多个阶段，每个阶段都由 4 部分组成，它们是：



① 目标设定。为该项目进行需求分析，定义和确定这一个阶段的专门目标，指定对过程和产品的约束，并且制定详细的管理计划。

② 风险分析。对可选方案进行风险识别和详细分析，制定解决办法，采取有效的措施避免这些风险。

③ 开发和有效性验证。风险评估后，可以为系统选择开发模型，并且进行原型开发，即开发软件产品。

④ 评审。对项目进行评审，以确定是否需要进入螺旋线的下一次回路，如果决定继续，就要制定下一阶段计划。

螺旋模型的软件开发过程实际是上述 4 个部分的迭代过程，每迭代一次，螺旋线就增加一周，软件系统就生成一个新版本，这个新版本实际上是对目标系统的一个逼近。经过若干次的迭代后，系统应该尽快地收敛到用户允许或可以接受的目标范围内，否则也可能中途夭折。

【答案】B。

5. 【2012 年题 26 解析】

【答案】A。

6. 【2016 年题 18 解析】

【答案】B。

7. 【2017 年题 20 解析】

软件过程模型的基本概念：软件过程是制作软件产品的一组活动以及结果，这些活动主要由软件人员来完成，软件活动主要有：

- (1) 软件描述。必须定义软件功能以及使用的限制。
- (2) 软件开发。也就是软件的设计和实现，软件工程师人员制作出能满足描述的软件。
- (3) 软件有效性验证。软件必须经过严格的验证，以保证能够满足客户的需求。
- (4) 软件进化。软件随着客户需求的变化不断地改进。

瀑布模型的特点是因果关系紧密相连，前一个阶段工作的结果是后一个阶段工作的输入。或者说，每一个阶段都是建筑在前一个阶段正确结果之上，前一个阶段的错漏会隐蔽地带到后一个阶段。这种错误有时甚至可能是灾难性的。因此每一个阶段工作完成后，都要进行审查和确认，这是非常重要的。历史上，瀑布模型起到了重要作用，它的出现有利于人员的组织管理，有利于软件开发方法和工具的研究。

【答案】D、A。

8. 【2018 年题 20 解析】

答案 A。

9. 【2018 年题 21 解析】

系统模块化程度较高时, 更适合于采用快速应用开发方法。

螺旋模型是在快速原型的基础上扩展而成的。这个模型把整个软件开发流程分成多个阶段, 每个阶段都由 4 部分组成, 它们是: ①目标设定。为该项目进行需求分析, 定义和确定这一个阶段的专门目标, 指定对过程和产品的约束, 并且制定详细的管理计划。②风险分析。对可选方案进行风险识别和详细分析, 制定解决办法, 采取有效的措施避免这些风险。③开发和有效性验证。风险评估后, 可以为系统选择开发模型, 并且进行原型开发, 即开发软件产品。④评审。对项目进行评审, 以确定是否需要进入螺旋线的下一次回路, 如果决定继续, 就要制定下一阶段计划。

答案 AC。

6.2.5 构件组装模型

1. 【2009 年题 27 解析】

基于构件的开发模型利用模块化方法将整个系统模块化, 并在一定构件模型的支持下复用构件库中的一个或多个软件构件, 通过组合手段高效率、高质量地构造应用软件系统的过程。基于构件的开发模型融合了螺旋模型的许多特征, 本质上是演化形的, 开发过程是迭代的。基于构件的开发模型由软件的需求分析定义、体系结构设计、构件库建立、应用软件构建以及测试和发布 5 个阶段组成。

2. 【2014 年题 30 解析】

检测并消除体系结构失配: 体系结构失配问题由 David Garlan 等人在 1995 年提出。失配是指在软件复用的过程中, 由于待复用构件对最终系统的体系结构和环境的假设 (assumption) 与实际状况不同而导致的冲突。在构件组装阶段失配问题主要包括:

(1) 由构件引起的失配, 包括由于系统对构件基础设施、构件控制模型和构件数据模型的假设存在冲突引起的失配;

(2) 由连接器引起的失配, 包括由于系统对构件交互协议、连接器数据模型的假设存在冲突引起的失配;

(3) 由于系统成分对全局体系结构的假设存在冲突引起的失配等。要解决失配问题, 首先需要检测出失配问题, 并在此基础上通过适当的手段消除检测出的失配问题。

6.3 统一过程

1. 【2011 年题 21 解析】

RUP 也称为 UP、统一过程, 其核心特点是: 以架构为中心, 用例驱动, 迭代与增量。该开发模型分 4 个阶段, 分别为: 初始、细化、构造、移交。其中题干所述的“确定系统的体系结构”是细化阶段的主要工作, 所以该空应填细化。

【答案】C、B。

2. 【2013 年题 23 解析】

RUP 包括四个阶段: 初始阶段、细化阶段、构建阶段、交付阶段。

初始阶段的任务是为系统建立业务模型并确定项目的边界。

细化阶段的任务是分析问题领域, 建立完善的架构, 淘汰项目中最高风险的元素。

在构建阶段, 要开发所有剩余的构件和应用程序功能, 把这些构件集成为产品, 并进行详细测试。

交付阶段。交付阶段的重点是确保软件对最终用户是可用的。

RUP 中的每个阶段可以进一步分解为迭代。一个迭代是一个完整的开发循环。

【答案】B、A。

3. 【2015 年题 25 解析】

RUP（统一软件开发过程，Rational Unified Process），RUP 的三个核心特点是：以架构为中心，用例驱动，增量与迭代。

其中增量与迭代的好处是：

降低了在一个增量上的开支风险。如果开发人员重复某个迭代，那么损失只是这一个开发有误的迭代的花费。

降低了产品无法按照既定进度进入市场的风险。通过在开发早期就确定风险，可以尽早来解决而不至于在开发后期匆匆忙忙。

加快了整个开发工作的进度。因为开发人员清楚问题的焦点所在，他们的工作会更有效率。

由于用户的需求并不能在一开始就作出完全的界定，它们通常是在后续阶段中不断细化的。因此，迭代过程这种模式使适应需求的变化会更容易些。

【答案】：C、A。

6.4 敏捷方法

1. 【2009 年题 21 解析】

敏捷方法以原型开发思想为基础，采用迭代增量式开发，发行版本小型化，比较适合需求变化较大或者开发前期对需求不是很清晰的项目。

【答案】D。

2. 【2014 年题 19 解析】

敏捷方法适合于开发团队较小的项目。

3. 【2017 年题 21 解析】

敏捷方法是面向对象的，而非面向过程。

6.5 软件重用

1. 【2010 年题 27 解析】

软件重用是指在两次或多次不同的软件开发过程中重复使用相同或相似软件元素的过程。按照重用活动是否跨越相似性较少的多个应用领域，软件重用可以区分为横向重用和纵向重用。横向重用是指重用不同应用领域中的软件元素，例如数据结构、分类算法和人机界面构建等。**标准函数是一种典型的、原始的横向重用机制。**纵向重用是指在一类具有较多公共性的应用领域之间进行软件重用。**纵向重用活动的主要关键点是域分析：**根据应用领域的特征及相似性预测软件的可重用性。

【答案】C。

2. 【2018 年题 25 解析】

软件重用（软件复用）是使用已有的软件产品（如设计、代码和文档等）来开发新的软件系统的过程。

软件重用的形式大体可分为垂直式重用和水平式重用。水平式重用是重用不同应用领域中的软件元素，如**标准函数库**。

垂直式重用是在一类具有较多公共性的应用领域之间重用软件构件。

答案 B。

6.5.2 构件技术

1. 【2017 年题 26 解析】

构件的特性是:

- (1) 独立部署单元;
- (2) 作为第三方的组装单元;
- (3) 没有(外部的)可见状态。

一个构件可以包含多个类元素,但是一个类元素只能属于一个构件。将一个类拆分进行部署通常没什么意义。

对象的特性是:

- (1) 一个实例单元,具有唯一的标志。
- (2) 可能具有状态,此状态外部可见。
- (3) 封装了自己的状态和行为。

【答案】C。

2. 【2017 年题 27 解析】

接口标准化是对接口中消息的格式、模式和协议的标准化。它不是要将接口格式化为参数化操作的集合,而是关注输入输出的消息的标准化,它强调当机器在网络中互连时,标准的消息模式、格式、协议的重要性。这也是因特网(IP, UDP, TCP, SNMP, 等等)和 Web(HTTP, HTML, 等等)标准的主要做法。为了获得更广泛的语义,有必要在一个单一通用的消息格式语境中标准化消息模式。这就是 XML 的思想。XML 提供了一种统一的数据格式。

【答案】B。

6.6 基于架构的软件设计

1. 【2009 年题 38 解析】

基于架构的软件设计(ABSD)强调由商业、质量和功能需求的组合驱动软件架构设计。使用 ABSD 方法,设计活动可以从项目总体功能框架明确就开始,并且设计活动的开始并不意味着需求抽取和分析活动可以终止,而是应该与设计活动并行。

ABSD 方法有三个基础:

第一个基础是功能分解,在功能分解中使用已有的基于模块的内聚和耦合技术。

第二个基础是通过选择体系结构风格来实现质量和商业需求。

第三个基础是软件模板的使用。ABSD 方法是一个自顶向下,递归细化的过程,软件系统的架构通过该方法得到细化,直到能产生软件构件的类。

【答案】D。

2. 【2010 年题 39 解析】

根据定义,基于软件架构的开发(Architecture Based Software Development, ABSD)强调由商业、质量和功能需求的组合驱动软件架构设计。它强调采用视角和视图来描述软件架构,采用用例和质量属性场景来描述需求。

【答案】: B、C。

3. 【2011 年题 36 解析】

ABSD 方法有三个基础:

(1)功能的分解。使用已有的基于模块的内聚和耦合技术。

(2)通过选择体系结构风格来实现质量和商业需求。

(3)软件模板的使用。软件模板是一个特殊类型的软件元素,包括描述所有这种类型的元素在共享服务和底层构造的基础上如何进行交互。软件模板还包括属于这种类型的所有元素的功能,这些功能的例子有:每个元素必须记录某些重大事件,每个元素必须为运行期间

的外部诊断提供测试点等。

4. 【2013 年题 33 解析】

基于架构的软件设计(Architecture-Based Software Design, ABSD)方法有三个基础,分别是对系统进行**功能分解**、**采用架构风格**实现质量属性与商业需求、采用软件模板设计软件结构。ABSD 方法主要包括架构需求等 6 个主要活动,其中**架构复审**活动的目标是标识潜在的风险,及早发现架构设计中的缺陷和错误;**架构演化**活动针对用户的需求变化,修改应用架构,满足新的需求。

软件架构文档应该从使用者的角度进行书写,针对不同背景的人员采用不同的书写方式,并将文档分发给相关人员。**架构文档要保持较新,但不要随时保证文档最新,要保持文档的稳定性。架构文档化的主要输出结果是架构规格说明书和架构质量说明书。**

【答案】: A、C、D、B、C。

5. 【2015 年题 32 解析】

根据定义,基于软件架构的开发(Architecture Based Software Development, ABSD)强调由商业、质量和功能需求的组合驱动软件架构设计。它强调采用**视角和视图**来描述软件架构,采用**用例和质量属性场景**来描述需求。

【答案】: B、C。

6. 【2017 年题 32 解析】

根据基于软件架构的设计的定义,基于软件架构的设计(Architecture Based Software Development, ABSD)强调由商业、质量和功能需求的组合驱动软件架构设计。它强调采用**视角和视图**来描述软件架构,采用**用例和质量属性场景**来描述需求。进一步来说,用例描述的是功能需求,质量属性场景描述的是质量需求(或侧重于非功能需求)。

【答案】B、C、C。

7. 【2019 年题 22 解析】

本题考查的是构件管理相关知识。

目前,已有的构件分类方法可以分为三大类,分别是关键字分类法、刻面分类法和超文本组织方法。

关键字分类法:是一种最简单的构件库组织方法,其基本思想是:根据领域分析的结果将应用领域的概念按照从抽象到具体的顺序逐次分解为树状或有向无回路图结构。每个概念用一个描述性的关键字表示。不可分解的原子级关键字包含隶属于它的某些构件。第一空描述的是关键字分类法,选择 A 选项。

刻面分类法:在刻面分类机制中,定义若干用于刻画构件特征的“面”(facet),每个面包含若干概念,这些概念表述构件在面上的特征。刻画可以描述构件执行的功能、被操作的数据、构件应用的语境或任意其他特征。第二空描述的是刻面分类法,选择 B 选项。

超文本组织方法:超文本组织方法与基于数据库系统的构件库组织方法不同,它基于全文检索技术,主要思想是:所有构件必须辅以详尽的功能或行为说明文档;说明中出现的重要概念或构件以网状链接方式相互连接;检索者在阅读文档的过程中可按照人类的联系思维方式任意跳转到包含相关概念或构件的文档;全文检索系统将用户给出的关键字与说明文档中的文字进行匹配,实现构件的浏览式检索。第三空描述的是超文本方法,选择 D 选项。

8. 【2019 年题 23 解析】

本题考查的是构件相关知识。

构件组装是指将构件库中的构件经过适当修改后相互连接,或者将它们与当前开发项目中的构件元素相连接,最终构成新的目标软件。

构件组装技术大致可分为基于功能的组装技术、基于数据的组装技术和面向对象的组装技术。本题 C 选项不属于构件组装技术。

9. 【2019 年题 27 解析】

本题第一空选择 D 选项。

图示展示的是类图的结构。

注意区分类图和对象图。对象图标记的是对象名,命名形式 对象名:类名,或者 :类名。这里没有出现冒号,表示的是类图。

对象图(object diagram)。对象图描述一组对象及它们之间的关系。对象图描述了在类图中所建立的事物实例的静态快照。和类图一样,这些图给出系统的静态设计视图或静态进程视图,但它们是从真实案例或原型案例的角度建立的。

类图(class diagram)。类图描述一组类、接口、协作和它们之间的关系。在 OO 系统的建模中,最常见的图就是类图。类图给出了系统的静态设计视图,活动类的类图给出了系统的静态进程视图。

本题第二空选择 B 选项。

“4+1”视图模型从五个不同的视角来描述软件架构,每个视图只关心系统的一个侧面,五个视图结合在一起才能反映软件架构的全部内容。

(1) 逻辑视图。逻辑视图主要支持系统的功能需求,即系统提供给最终用户的服务。在逻辑视图中,系统分解成一系列的功能抽象,这些抽象主要来自问题领域。这种分解不但可以用来进行功能分析,而且可用作标识在整个系统的各个不同部分的通用机制和设计元素。在 OO 技术中,通过抽象、封装和继承,可以用对象模型来代表逻辑视图,用类图来描述逻辑视图。逻辑视图中使用的风格为面向对象的风格,在设计中要注意保持一个单一的、内聚的对象模型贯穿整个系统。

(2) 开发视图。开发视图也称为模块视图,在 UML 中被称为实现视图,它主要侧重于软件模块的组织和管理。开发视图要考虑软件内部的需求,例如,软件开发的容易性、软件复用和软件的通用性,要充分考虑到由于具体开发工具的不同而带来的局限性。开发视图通过系统 I/O 关系的模型图和子系统图来描述。其中类图属于开发视图。

(3) 进程视图。进程视图侧重于系统的运行特性,主要关注一些非功能性需求,例如,系统的性能和可用性等。进程视图强调并发性、分布性、系统集成性和容错能力,以及逻辑视图中的功能抽象如何适合进程结构等,它也定义了逻辑视图中的各个类的操作具体是在哪一个线程中被执行的。进程视图可以描述成多层抽象,每个级别分别关注不同的方面。

(4) 物理视图。物理视图在 UML 中被称为部署视图,它主要考虑如何把软件映射到硬件上,它通常要考虑到解决系统拓扑结构、系统安装和通信等问题。当软件运行于不同的物理节点上时,各视图中的构件都直接或间接地对应于系统的不同节点上。因此,从软件到节点的映射要有较高的灵活性,当环境改变时,对系统其他视图的影响最小化。

(5) 场景。场景可以看作是那些重要系统活动的抽象,它使四个视图有机联系起来,从某种意义上说场景是最重要的需求抽象。场景视图对应 UML 中的用例视图。在开发软件架构时,它可以帮助架构设计师找到构件及其相互关系。同时,架构设计师也可以用场景来分析一个特定的视图,或描述不同视图的构件之间是如何相互作用的。场景可以用文本表示,也可以用图形表示。

6.6.2 基于架构的软件开发模型

1. 【2009 年题 37 解析】

软件架构需求是指用户对目标软件系统在功能、行为、性能 and 设计约束等方面的期望。需求过程主要是**获取用户需求**,**标识系统中所要用到的构件**,并进行**架构需求评审**。其中标识构件又详细分为生成类图、对类图进行分组和将类打包成构件三步。软件架构需求并不应该包括设计构件的过程。

【答案】A。

2. 【2009 年题 40 解析】

架构复审是基于架构开发中一个重要的环节。架构设计、文档化和复审是一个迭代的过程。从这个方面来说,在一个主版本的软件架构分析之后,**要安排一次由外部人员(用户代表和领域专家)参加的复审**。架构复审过程中,通常会对一个可运行的最小化系统进行架构

评估和测试。架构复审的目标是标识潜在的风险,及早发现架构设计的缺陷和错误。

3. 【2010 年题 31 解析】

在基于构件的软件开发中,**逻辑构件模型**用功能包描述系统的抽象设计,用接口描述每个服务集合,以及功能之间如何交互以满足用户需求,它**作为系统的设计蓝图以保证系统提供适当的功能**。**物理构件模型**用技术设施产品、硬件分布和拓扑结构、以及用于绑定的网络和通信协议描述系统的物理设计,这种架构用于了解系统的性能、吞吐率等许多非功能性属性。

4. 【2019 年题 30 解析】

本题考查的是基于体系结构的开发模型 ABSDM 相关知识。

ABSDM 模型把整个基于体系结构的软件过程划分为:体系结构需求、设计、文档化、复审、实现和演化等 6 个过程。因此第一空选择 A 选项体系结构复审,第二空选择 A 选项体系结构实现。

体系结构需求:需求过程主要是获取用户需求,标识系统中所要用到的构件。

体系结构设计:体系结构设计是一个迭代过程,如果要开发的系统能够从已有的系统中导出大部分,则可以使用已有系统的设计过程。

体系结构文档化:绝大多数的体系结构都是抽象的,由一些概念上的构件组成,因此要去实现体系结构,还必须得把体系结构文档化。体系结构文档化过程的主要输出结果是体系结构规格说明和测试体系结构需求的质量设计说明书这 2 个文档。本题第三空选择 C 选项。

体系结构复审:体系结构设计、文档化和复审是一个迭代过程。复审的目的是表示潜在的风险,及早发现体系结构设计中的缺陷和错误,包括体系结构能否满足需求、质量需求是否在设计中得到体现、层次是否清晰、构件划分是否合理、文档表达是否明确、构件设计是否满足功能与性能的要求等。

体系结构实现:所谓“实现”就是要用实体显示出一个软件体系结构,即要符合体系结构描述的结构化设计决策,分割成规定的构件,按规定的方式互相交互。整个实现过程是以复审后的文档化的体系结构说明书为基础的,每个构件必须满足软件体系结构中说明的对其他构件的责任。最后一步是测试,包括单个构件的功能性测试和被组装应用的整体功能和性能测试。

体系结构演化:在构件开发过程中,用户的需求可能还有变动。在软件开发完毕,正常运行后,由一个单位移植到另一个单位,需求也会发生变化。在这两种情况下,就必须相应地修改软件体系结构,以适应新的变化了的软件需求。体系结构演化是使用系统演化步骤去修改应用,以满足新的需求。

6.7 形式化方法

1. 【2012 年题 22 解析】

软件开发方法是指软件开发过程所遵循的办法和步骤,从不同的角度可以对软件开发方法进行不同的分类。

形式化方法是一种具有坚实数学基础的方法,从而允许对系统和开发过程做严格处理和论证,适用于那些系统安全级别要求极高的软件的开发。形式化方法的主要优越性在于它能够**数学地表述和研究应用问题及软件实现(B 选项)**。但是它要求开发人员具备良好的数学基础。用形式化语言书写的大型应用问题的软件规格说明往往过于细节化,并且难于为用户和软件设计人员所理解。由于这些缺陷,形式化方法在目前的软件开发实践中并未得到普遍应用。

净室软件工程(Cleanroom Software Engineering, CSE)是软件开发的一种形式化方法,可以开发较高质量的软件。它使用盒结构规约进行分析和建模,并且将**正确性验证作为发现和排除错误的主要机制(C 选项)**,使用统计测试来获取认证软件可靠性所需要的信息。CSE 强

调在规约和设计上的严格性,还**强调统计质量控制技术**,包括基于客户对软件的预期使用测试(D 选项)。

【答案】A。

6.8 其他

1. 【2009 年题 23 解析】

逆向工程导出的信息可分为如下 4 个抽象层次。

实现级: 包括程序的抽象语法树、符号表等信息。

结构级: 包括反映程序分量之间相互依赖关系的信息, 例如调用图、结构图等。

功能级: 包括反映程序段功能及程序段之间关系的信息。

领域级: 包括反映程序分量或程序与应用领域概念之间对应关系的信息。

2. 【2009 年题 39 解析】

软件架构文档是对软件架构的一种描述,帮助程序员使用特定的程序设计语言实现软件架构。软件架构文档的写作应该遵循一定的原则,这些原则包括:文档要从使用者的角度进行编写;必须分发给所有与系统有关的开发人员;应该保持架构文档的即时更新,但更新不要过于频繁;架构文档中描述应该尽量避免不必要的重复:每次架构文档修改都应该记录进行修改的原则。

【答案】A。

3. 【2010 年题 28 解析】

【答案】B。ACD 选项说法绝对。

4. 【2013 年题 25 解析】

自顶向下方法的优点是:

1、可为企业或机构的重要决策和任务实现提供信息。

2、支持企业信息系统的整体性规划,并对系统的各子系统的协调和通信提供保证。

3、方法的实践有利于提高企业人员整体观察问题的能力,从而有利于寻找到改进企业组织的途径。

自顶向下方法的缺点是:

1、对系统分析和设计人员的要求较高。

2、开发周期长,系统复杂,一般属于一种高成本、大投资的工程。

3、对于大系统而言自上而下的规划对于下层系统的实施往往缺乏约束力。

4、从经济角度来看,很难说自顶向下的做法在经济上是合算的。

【答案】C。

5. 【2016 年题 19 解析】

A、D, 区别太复杂,选择性放弃。

6. 【2017 年题 28 解析】

暂无。

【答案】D、A。

7. 【2017 年题 33 解析】

体系结构文档化过程的主要输出结果是体系结构规格说明和测试体系结构需求的质量设计说明书这两个文档。软件体系结构的文档要求与软件开发项目中的其他文档是类似的。文档的完整性和质量是软件体系结构成功的关键因素。文档要从使用者的角度进行编写,必须分发给所有与系统有关的开发人员,且必须保证开发者手上的文档是最新的。

【答案】: A。

8. 【2019 年题 24 解析】

逆向工程导出的信息可分为如下 4 个抽象层次。

- ① 实现级：包括程序的抽象语法树、符号表等信息。
- ② 结构级：包括反映程序分量之间相互依赖关系的信息，例如调用图、结构图等。本题第一空描述的是结构级，选择 B 选项。
- ③ 功能级：包括反映程序段功能及程序段之间关系的信息。本题第二空描述的是功能级，选择 C 选项。
- ④ 领域级：包括反映程序分量或程序与应用领域概念之间对应关系的信息。

9. 【2019 年题 25 解析】

所谓软件的逆向工程就是分析已有的程序，寻求比源代码更高级的抽象表现形式。一般认为，凡是在软件生命周期内将软件某种形式的描述转换成更为抽象形式的活动都可称为逆向工程。

与之相关的概念是：

重构 (restructuring)，指在同一抽象级别上转换系统描述形式；本题描述的是重构工程。本题描述的是重构，选择 D 选项。

设计恢复 (design recovery)，指借助工具从已有程序中抽象出有关数据设计、总体结构设计和过程设计的信息（不一定是原设计）。

再工程 (re-engineering)，也称修复和改造工程，它是在逆向工程所获信息的基础上修改或重构已有的系统，产生系统的一个新版本。

10. 【2019 年题 41 解析】

本题是对软件开发方法相关内容的考查。

软件开发方法是指软件开发过程所遵循的办法和步骤，从不同的角度可以对软件开发方法进行不同的分类。

从开发风范上开，可分为自顶向下的开发方法和自底向上的开发方法。（在实际软件开发中，大都是两种方法结合，只不过是应用于开发的不同阶段以何者为主而已）

“先对最高层次中的问题进行定义、设计、编程和测试，而将其中未解决的问题作为一个子任务放到下一层次中去解决”描述的是自顶向下的开发，第一空选择 D 选项。

“根据系统功能要求，从具体的器件、逻辑部件或者相似系统开始，通过对其进行相互连接、修改和扩大，构成所要求的系统”描述的是自底向上的开发，第二空选择 A 选项。

从性质上看，可分为形式化方法和非形式化方法。

形式化方法是一种具有坚实数学基础的方法，从而允许对系统和开发过程做严格处理和论证，适用于那些系统安全级别要求极高的软件的开发。第三空选择 B 选项。

非形式化方法则不把严格性作为其主要着眼点，通常以各种开发模型的形式得以体现。

从适应范围来看，可分为整体性方法与局部性方法。适用于软件开发全过程的方法称为整体性方法；适用于开发过程某个具体阶段的软件方法称为局部性方法。