开发方法

目录

[开发方法 1](#_Toc524290082)

[1 软件生命周期 3](#_Toc524290083)

[2 软件开发方法 7](#_Toc524290084)

[2.1 净室方法 14](#_Toc524290085)

[2.2 结构化方法 14](#_Toc524290086)

[2.3 面向对象方法 15](#_Toc524290087)

[2.4 原型法 24](#_Toc524290088)

[2.5 逆向工程 24](#_Toc524290089)

[3 软件开发模型 26](#_Toc524290090)

[3.1 瀑布模型 29](#_Toc524290091)

[3.2 演化模型 29](#_Toc524290092)

[3.3 螺旋模型 29](#_Toc524290093)

[3.4 增量模型 30](#_Toc524290094)

[3.5 构件组装模型 30](#_Toc524290095)

[3.6 统一过程 30](#_Toc524290096)

[3.7 敏捷方法 30](#_Toc524290097)

[3.7.1 极限编程 32](#_Toc524290098)

[3.7.2 特征驱动开发 32](#_Toc524290099)

[3.7.3 Scrum 32](#_Toc524290100)

[3.7.4 水晶方法 32](#_Toc524290101)

[3.7.5 其他敏捷方法 32](#_Toc524290102)

[4 需求开发 33](#_Toc524290103)

[4.1 需求获取 33](#_Toc524290104)

[4.2 需求分析 34](#_Toc524290105)

[4.3 需求定义 36](#_Toc524290106)

[4.4 需求管理 36](#_Toc524290107)

[5 基于构件开发 36](#_Toc524290108)

[5.1 软件重用 38](#_Toc524290109)

[5.2 构件标准 39](#_Toc524290110)

[5.3 构件获取 39](#_Toc524290111)

[5.4 构件管理 39](#_Toc524290112)

[5.5 构件重用 39](#_Toc524290113)

[6 开发工具 39](#_Toc524290114)

# 软件生命周期

## 知识点

外部设计处于软件设计的开始阶段，主要是按系统需求说明来确定此系统的软件结构和对应于系统需求说明，设计出各个功能部分的功能和接口。内部设计处于软件工程中的概要设计阶段，按照外部设计中确立的系统软件结构，来细化此系统各个功能部件以及各个部件接口的设计，并且详细给出各个功能部件详细的数据输入、输出设计。

RUP软件开发生命周期是一个二维的软件开发模型，其中有9个核心工作流，分别为：业务建模、需求、分析与设计、实现、测试部署、配置管理、变更管理、项目管理以及环境。

RUP把软件开发生存周期划分为多个循环，每个循环生成产品的一个新的版本，每个循环依次由4个连续的阶段组成，每个阶段完成确定的任务。这4个阶段分别为：

初始阶段（inception)：定义最终产品视图和业务模型，并确定系统范围。

细化阶段（elaboration)：设计及确定系统的体系结构，制定工作计划及资源要求。

构造阶段（construction)：构造产品并继续演进需求、体系结构、计划直至产品提交。

移交阶段（transition)：把产品提交给用户使用。

每个阶段都有一个或多个连续的迭代组成。迭代并不是重复地做相同的事，而是针对不同用例的细化和实现。每一个迭代都是一个完整的开发过程，它需要项目经理根据当前迭代所处的阶段以及上次迭代的结果，适当地对工作流中的行为进行裁剪。在每个阶段结束前有一个里程碑评估该阶段的工作。如果未能通过该里程碑的评估，则决策者应该做出决定，是取消该项目还是继续该阶段的工作。

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。移交阶段的重点是确保软件对最终用户是可用的。

与其他软件开发过程相比，RUP具有自己的特点，即RUP是用例驱动的、以体系结构为中心的、迭代和增量的软件开发过程。

## 试题

* 系统设计是软件开发的重要阶段，(39)主要是按系统需求说明来确定此系统的软件结构，并设计出各个部分的功能和接口。

2011年(39)

A.外部设计

B.内部设计

C.程序设计

D.输入/输出设计

【答案】A 【解析】

外部设计处于软件设计的开始阶段，主要是按系统需求说明来确定此系统的软件结构和对应于系统需求说明，设计出各个功能部分的功能和接口。内部设计处于软件工程中的概要设计阶段，按照外部设计中确立的系统软件结构，来细化此系统各个功能部件以及各个部件接口的设计，并且详细给出各个功能部件详细的数据输入、输出设计。内部设计细化外部设计中的各种功能。

* RUP强调采用（34）的方式来开发软件，这样做的好处是（35）。

2015年(34)

A.原型和螺旋

B.螺旋和增量

C.迭代和增量

D.快速和迭代

2015年(35)

A.在软件开发的早期就可以对关键的、影响大的风险进行处理

B.可以避免需求的变更

C.能够非常快速地实现系统的所有需求

D.能够更好地控制软件的质量

【答案】C A 【解析】

RUP将项目管理、业务建模、分析与设计等统一起来，贯穿整个开发过程。RUP中的软件过程在时间上被分解为4个顺序的阶段，分别是初始阶段、细化阶段、构建阶段和移交阶段。每个阶段结束时都要安排一次技术评审，以确定这个阶段的目标是否已经满足。如果评审结果令人满意，就可以允许项目进入下一个阶段。可以看出，基于RUP的软件过程是一个迭代和增量的过程。通过初始、细化、构建和移交4个阶段就是一个开发周期，每次经过这4个阶段就会产生一代软件。除非产品退役，否则通过重复同样的4个阶段，产品将演化为下一代产品，但每一次的侧重点都将放在不同的阶段上。这样做的好处是在软件开发的早期就可以对关键的、影响大的风险进行处理。

* 基于RUP的软件过程是一个迭代过程。一个开发周期包括初始、细化、构建和移交四个阶段，每次通过这四个阶段就会产生一代软件，其中建立完善的架构是（33)阶段的任务。采用迭代式开发，(34)。

2013年(33)

A.初始

B.细化

C.构建

D.移交

2013年(34)

A.在每一轮迭代中都要进行测试与集成

B.每一轮迭代的重点是对特定的用例进行部分实现

C.在后续迭代中强调用户的主动参与

D.通常以功能分解为基础

【答案】B A 【解析】 RUP中的软件过程在时间上被分解为4个顺序的阶段，分别是初始阶段、细化阶段、构建阶段和移交阶段。

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。移交阶段的重点是确保软件对最终用户是可用的。

基于RUP的软件过程是一个迭代过程，通过初始、细化、构建和移交4个阶段就是一个开发周期，每次经过这4个阶段就会产生一代产品，在每一轮迭代中都要进行测试与集成。

* RUP是一个二维的软件开发模型，其核心特点之一是（27)。RUP将软件开发生存周期划分为多个循环（cycle),每个循环由4个连续的阶段组成，每个阶段完成确定的任务。设计及确定系统的体系结构，制定工作计划及资源要求是在（28)阶段完成的。

2011年(27)

A.数据驱动

B.模型驱动

C.用例驱动

D.状态驱动

2011年(28)

A.初始（inception)

B.细化（elaboration)

C.构造（construction)

D.移交（transition)

【答案】C B 【解析】

RUP软件开发生命周期是一个二维的软件开发模型，其中有9个核心工作流，分别为：业务建模、需求、分析与设计、实现、测试部署、配置与变更管理、项目管理以及环境。

RUP把软件开发生存周期划分为多个循环，每个循环生成产品的一个新的版本，每个循环依次由4个连续的阶段组成，每个阶段完成确定的任务。这4个阶段分别为：

初始阶段：定义最终产品视图和业务模型，并确定系统范围。

细化阶段：设计及确定系统的体系结构，制定工作计划及资源要求。

构造阶段：构造产品并继续演进需求、体系结构、计划直至产品提交。

移交阶段：把产品提交给用户使用。

每个阶段都有一个或多个连续的迭代组成。迭代并不是重复地做相同的事，而是针对不同用例的细化和实现。每一个迭代都是一个完整的开发过程，它需要项目经理根据当前迭代所处的阶段以及上次迭代的结果，适当地对工作流中的行为进行裁剪。在每个阶段结束前有一个里程碑评估该阶段的工作。如果未能通过该里程碑的评估，则决策者应该做出决定，是取消该项目还是继续该阶段的工作。

与其他软件开发过程相比，RUP具有自己的特点，即RUP是用例驱动的、以体系结构为中心的、迭代和增量的软件开发过程。

# 软件开发方法

## 净室方法

### 知识点

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，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

### 试题

* 下列关于各种软件开发方法的叙述中，错误的是（32)。

2011年(32)

A.结构化开发方法的缺点是开发周期较长，难以适应需求变化

B.可以把结构化方法和面向对象方法结合起来进行系统开发，使用面向对象方法进行自顶向下的划分，自底向上地使用结构化方法开发系统

C.与传统方法相比，敏捷开发方法比较适合需求变化较大或者开发前期需求不是很清晰的项目，以它的灵活性来适应需求的变化

D.面向服务的方法以粗粒度、松散耦合和基于标准的服务为基础，增强了系统的灵活性、可复用性和可演化性

【答案】B 【解析】 结构化方法也称为生命周期法，是一种传统的信息系统开发方法，由结构化分析、结构化设计和结构化程序设计三部分组成，其精髓是自顶向下、逐步求精和模块化设计。

结构化方法的主要特点是：开发目标清晰化、开发工作阶段化、开发文档规范化和设计方法结构化。结构化方法特别适合于数据处理领域的问题，但是不适应于规模较大、比较复杂的系统开发。结构化方法的缺点是开发周期长、难以适应需求的变化、很少考虑数据结构。

面向对象方法是目前比较主流的开发方法。面向对象方法是系统的描述及信息模型的表示与客观实体相对应，符合人们的思维习惯，有利于系统开发过程中用户与开发人员的交流和沟通，缩短开发周期，提高系统开发的正确性和效率。可以把结构化方法和面向对象方法结合起来进行系统开发。首先使用结构化方法进行自顶向下的整体划分；然后再自底向上地采用面向对象方法开发系统。

敏捷方法是从20世纪90年代开始逐渐引起广泛关注的一种新型软件开发方法，以应对快速变化的需求。敏捷方法是一种以人为核心、迭代、循序渐进的开发方法。敏捷方法强调，让客户满意和软件尽早增量发布：小而高度自主的项目团队；非正式g方法：最小化软件工程工作产品以及整体精简开发。与传统方法相比，敏捷开发方法比较适合需求变化较大或者开发前期需求不是很清晰的项目，以它的灵活性来适应需求的变化。

面向服务的方法以粗粒度、松散耦合和基于标准的服务为基础，增强了系统的灵活性、可复用性和可演化性。

* 以下关于软件开发方法的叙述，错误的是（31)。

2012年(31)

A.对于较为复杂的应用问题，适合釆用形式化方法进行需求分析

B.形式化方法的优势在于能够精确地表述和研究应用问题及其软件实现

C.净室软件工程将正确性验证作为发现和排除错误的主要机制

D.净室软件工程强调统计质量控制技术，包括对客户软件使用预期的测试

【答案】A 【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。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，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。

## 结构化方法

### 知识点

结构化方法的主要特点是：开发目标清晰化、开发工作阶段化、开发文档规范化和设计方法结构化。结构化方法特别适合于数据处理领域的问题，但是不适应于规模较大、比较复杂的系统开发。结构化方法的缺点是开发周期长、难以适应需求的变化、很少考虑数据结构。

由结构化分析、结构化设计、结构化程序设计组成，是一种面向数据流的开发方法。结构化分析是根据分解与抽象的原则，按照系统中数据处理的流程，用数据流图来建立系统的功能类型，从而完成需求分析工作。结构化设计是根据模块独立性准则、软件结构优化准则将数据流图转换为软件的结构体系，用软件结构图来建立系统的物理模型，实现系统的概要设计。结构化程序设计使用3种基本控制结构构造程序，任何程序都可以由顺序、选择和重复3种基本控制结构构造。

自顶向下方法是一种决策策略。软件开发涉及作什么决策、如何决策和决策顺序等决策问题。

自顶向下方法在任何时刻所作的决定都是当时对整个设计影响最大的那些决定。如果把所有决定分组或者分级，那么决策顺序是首先作最高级的决定，然后依次地作较低级的决定。同级的决定则按照随机的顺序或者按别的方法。一个决策的级别是看它距离要达到的最终目的（因此是软件的实际实现）的远近程度。从问题本身来看，或是由外(用户所见的）向内（系统的实现）看，以距离实现近的决定为低级决定，远的为高级决定。

在这个自顶向下的过程中，一个复杂的问题（任务）被分解成若干个较小较简单的问题（子任务），并且一直继续下去，直到每个小问题（子任务）都简单到能够解决（实现）为止。

自顶向下方法的优点是：

•可为企业或机构的重要决策和任务实现提供信息。

•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。

•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：

•对系统分析和设计人员的要求较高。

•开发周期长，系统复杂，一般属于一种高成本、大投资的工程。

•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，

•从经济角度来看，很难说自顶向下的做法在经济上市合算的。

结构化方法分析模型的核心是数据字典，围绕这个核心，有三个层次的模型，分别是数据模型、功能模型和行为模型（也称为状态模型）。在实际工作中，一般使用E-R图表示数据模型，用DFD表示功能模型，用状态转换图表示行为模型。这三个模型有着密切的关系，它们的建立不具有严格的时序性，而是一个迭代的过程。

### 试题

自顶向下方法是一种决策策略。软件开发涉及作什么决策、如何决策和决策顺序等决策问题。

* 以下关于自顶向下开发方法的叙述中，正确的是（37)

2013年(37)

A.自顶向下过程因为单元测试而比较耗费时间

B.自顶向下过程可以更快地发现系统性能方面的问题

C.相对于自底向上方法，自顶向下方法可以更快地得到系统的演示原型

D.在自顶向下的设计中，如发现了一个错误，通常是因为底层模块没有满足其规格说明（因为高层模块己经被测试过了）

【答案】C 【解析】 自顶向下方法是一种决策策略。软件开发涉及作什么决策、如何决策和决策顺序等决策问题。

自顶向下方法在任何时刻所作的决定都是当时对整个设计影响最大的那些决定。如果把所有决定分组或者分级，那么决策顺序是首先作最高级的决定，然后依次地作较低级的决定。同级的决定则按照随机的顺序或者按别的方法。一个决策的级别是看它距离要达到的最终目的（因此是软件的实际实现）的远近程度。从问题本身来看，或是由外(用户所见的）向内（系统的实现）看，以距离实现近的决定为低级决定，远的为高级决定。

在这个自顶向下的过程中，一个复杂的问题（任务）被分解成若干个较小较简单的问题（子任务），并且一直继续下去，直到每个小问题（子任务）都简单到能够i：接解决（实现）为止。

自顶向下方法的优点是：

•可为企业或机构的重要决策和任务实现提供信息。

•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。

•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：

•对系统分析和设计人员的要求较高。

•开发周期长，系统复杂，一般属于一种高成本、大投资的工程。

•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， •从经济角度来看，很难说自顶向下的做法在经济上市合算的。

* 在结构化分析方法中，用（27）表示功能模型，用（28）表示行为模型。

2014年(27)

A.ER图

B.用例图

C.DFD

D.对象图

2014年(28)

A.通信图

B.顺序图

C.活动图

D.状态转换图

【答案】C D 【解析】

结构化分析方法的基本思想是自顶向下，逐层分解，把一个大问题分解成若干个小问题，每个小问题再分解成若干个更小的问题。经过逐层分解，每个最低层的问题都是足够简单、容易解决的。结构化方法分析模型的核心是数据字典，围绕这个核心，有三个层次的模型，分别是数据模型、功能模型和行为模型（也称为状态模型）。在实际工作中，一般使用E-R图表示数据模型，用DFD表示功能模型，用状态转换图表示行为模型。这三个模型有着密切的关系，它们的建立不具有严格的时序性，而是一个迭代的过程。

## 面向对象方法

知识点

1. 面向对象方法是目前比较主流的开发方法。面向对象方法是系统的描述及信息模型的表示与客观实体相对应，符合人们的思维习惯，有利于系统开发过程中用户与开发人员的交流和沟通，缩短开发周期，提高系统开发的正确性和效率。可以把结构化方法和面向对象方法结合起来进行系统开发。首先使用结构化方法进行自顶向下的整体划分；然后再自底向上地采用面向对象方法开发系统。
2. 面向对象的分析模型主要由顶层架构图、用例与用例图、领域概念模型构成
3. 面向对象的设计模型则包含以包图表示的软件体系结构图、以交互图表示的用例实现图、完整精确的类图、针对复杂对象的状态图和用以描述流程化处理过程的活动图等。
4. 在面向对象设计中，类可以分为三种类型：实体类、边界类和控制类。①实体类映射需求中的每个实体，实体类保存需要存储在永久存储体中的信息。实体类是对用户来说最有意义的类，通常采用业务领域术语命名，一般来说是一个名词，在用例模型向领域模型转化中，一个参与者一般对应于实体类。②控制类是用于控制用例工作的类，一般是由动宾结构的短语（“动词+名词”或“名词+动词”）转化来的名词。控制类用于对一个或几个用例所特有的控制行为进行建模，控制对象通常控制其他对象，因此它们的行为具有协调性。③边界类用于封装在用例内、外流动的信息或数据流。边界类是一种用于对系统外部环境与其内部运作之间的交互进行建模的类。边界对象将系统与其外部环境的变更隔离开，使这些变更不会对系统其他部分造成影响。
5. 用例建模用来描述待开发系统的功能需求，主要元素是用例和参与者
6. 参与者是指系统以外的，需要使用系统或与系统交互的事物，包括：人或组织、设备、外部系统等。
7. 比较隐晦的参与者包括：时间、温度
8. 用例是对系统行为的动态描述，用例获取是需求分析阶段的主要任务之一
9. 用例之间的关系包括：包含、扩展、泛化
10. 两个用例之间的关系主要有两种情况：一种是用于重用的包含关系，用构造型include表示；另一种是用于分离出不同行为的扩展，用构造型extend表示。
11. ①包含关系：当可以从两个或两个以上的原始用例中提取公共行为，或者发现能够使用一个构件来实现某一个用例的部分功能是很重要的事时，应该使用包含关系来表示它们。
12. ②扩展关系：如果一个用例明显地混合了两种或两种以上的不同场景，即根据情况可能发生多种事情，可以断定将这个用例分为一个主用例和一个或多个辅用例描述可能更加清晰。
13. 类图主要用来描述系统的静态结构，是组件图和配置图的基础
14. 类之间的关系包括6种：关联、聚合、组合、依赖、泛化、实现。
15. 依赖关系：一个事物发生变化影响另一个事物。
16. 泛化关系：特殊/一般关系。
17. 关联关系：描述了一组链，链是对象之间的连接。
18. 聚合关系：整体与部分生命周期不同。
19. 组合关系：整体与部分生命周期相同。
20. 实现关系：接口与类之间的关系
21. 状态图和活动图
22. 状态图主要用于描述一个对象在其生存期间的动态行为，表现一个对象所经历的状态序列，引起状态转移的事件（event），以及因状态转移而伴随的动作（action）。
23. 活动图可以用于描述系统的工作流程和并发行为。活动图其实可看作状态图的特殊形式，活动图中一个活动结束后将立即进入下一个活动（在状态图中状态的转移可能需要事件的触发）。
24. 两者最大的区别是：状态图侧重于描述行为的结果，而活动图侧重描述行为的动作。其次活动图可描述并发行为，而状态图不能

试题

* 试题二【说明】某公司拟研制一款高空监视无人直升机，该无人机采用遥控一自主复合型控制实现垂直升降。该直升机飞行控制系统由机上部分和地面部分组成，机上部分主要包括无线电传输设备、飞控计算机、导航设备等，地面部分包括遥控操纵设备、无线电传输设备以及地面综合控制计算机等。其主要工作原理是地面综合控制计算机负责发送相应指令，飞控计算机按照预定程序实现相应功能。经过需求分析，对该无人直升机控制系统纵向控制基本功能整理如下：(a)飞控计算机加电后，应完成系统初始化，飞机进入准备起飞状态；(b)在准备起飞状态中等待地面综合控制计算机发送起飞指令，飞控计算机接收到起飞指令后，进入垂直起飞状态；(c)垂直起飞过程中如果飞控计算机发现飞机飞行异常，飞行控制系统应转入无线电遥控飞行状态，地面综合控制计算机发送遥控指令；(d)垂直起飞达到预定起飞高度后，飞机应进入高度保持状态；(e)飞控计算机在收到地面综合控制计算机发送的目标高度后，飞机应进入垂直升降状态，接近目标高度；垂直升降过程中出现飞机飞行异常，控制系统应转入无线电遥控飞行；(f)飞机到达目标高度后，应进入高度保持状态，完成相应的任务；(g)飞机在接到地面综合控制计算机发送的任务执行结束指令后，进入飞机降落状态；(h)飞机降落过程中如果出现飞机飞行异常，控制系统应转入无线电遥控飞行；(i)飞机降落到指定着陆高度后，进入飞机着陆状态，应按照预定着陆算法，进行着陆；(j)无线电遥控飞行中，地面综合控制计算机发送着陆指令，飞机进入着陆状态，应按照预定着陆算法，进行着陆。

2015【问题1】

状态图和活动图是软件系统设计建模中常用的两种手段，请用200字以内文字简要说明状态图和活动图的含义及其区别。

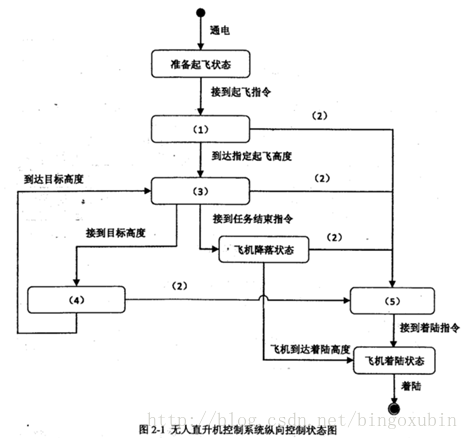
状态图主要用于描述一个对象在其生存期间的动态行为，表现一个对象所经历的状态序列，引起状态转移的事件（event），以及因状态转移而伴随的动作（action）。

活动图可以用于描述系统的工作流程和并发行为。活动图其实可看作状态图的特殊形式，活动图中一个活动结束后将立即进入下一个活动（在状态图中状态的转移可能需要事件的触发）。

两者最大的区别是：状态图侧重于描述行为的结果，而活动图侧重描述行为的动作。其次活动图可描述并发行为，而状态图不能。

2015【问题2】

根据题干中描述的基本功能需求，架构师王工通过对需求的分析和总结给出了无人直升机控制系统纵向控制状态图(图2-1)。请根据题干描述，提炼出相应状态及条件，并完善图2-1所示状态图中的(1)～(5)，将答案填写在答题纸中。



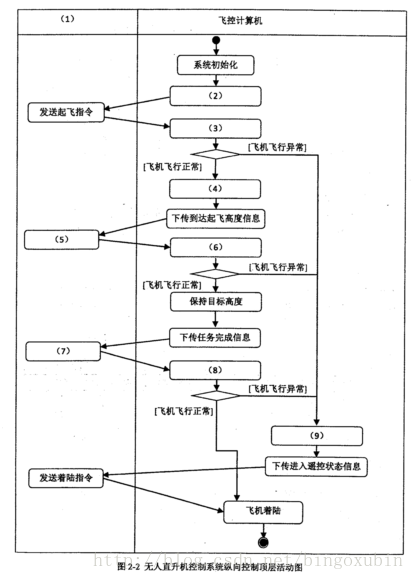
(1) 垂直起飞状态 (2) 飞机飞行异常 (3) 高度保持状态 (4) 垂直升降状态 (5) 无线电遥控飞行状态。

本问题考查系统建模中状态图的设计与应用。考生应该在熟记基本概念的基础上结合实际问题灵活掌握并应用这些概念。

在解答本题时，首先需要对题目中描述的基本功能需求(a)〜(j)进行分析与梳理，确定系统控制中的所有状态以及状态间的转换条件，再结合问题2中己经给出的状态，完成其余状态及条件的设计。

2015【问题3】

根据题目中描述的基本功能需求，架构师王工给出了无人直升机控制系统纵向控制的顶层活动图（图2-2）。请根据题干描述，完善图2-2活动图的（1）-（9），将答案填写在答题纸中。



(1) 地面综合控制计算机 (2) 下传起飞就绪信息 (3) 垂直起飞 (4) 高度保持 (5) 发送目标高度 (6) 垂直升降 2015年(7)

发送任务结束指令 (8) 飞机降落 (9) 无线电遥控飞行

本问题考查系统建模中活动图的设计与应用。考生应该掌握泳道活动图的概念并且学会应用。泳道活动图，是将一个活动图中的活动状态进行分组，每一组表示一个特定的类或者对象，它们负责完成组内的活动。每个活动都明确属于一个泳道，不可以跨越泳道，而转移则可以跨越泳道。

在解答本题时，首先需要对题目中描述的基本功能需求进行分析与梳理，确定题目中存在哪些硬件设备与飞控计算机进行交互，以及设备间的交互关系，再结合问题3中己经给出的活动，完成其余活动及设备的设计。

* 阅读以下关于软件系统建模的叙述，在答题纸上回答问题1至问题3。【说明】某软件公司计划开发一套教学管理系统，用于为高校提供教学管理服务。该教学管理系统基本的需求包括：(1)系统用户必须成功登录到系统后才能使用系统的各项功能服务；(2)管理员（Registrar）使用该系统管理学校（University）、系（Department）、教师（Lecturer）、学生（Student）和课程(Course)等教学基础信息；(3)学生使用系统选择并注册课程，必须通过所选课程的考试才能获得学分；如果考试不及格，必须参加补考，通过后才能获得课程学分；(4)教师使用该系统选择所要教的课程，并从系统获得选择该课程的学生名单；(5)管理员使用系统生成课程课表，维护系统所需的有关课程、学生和教师的信息；(6)每个月到了月底系统会通过打印机打印学生的考勤信息。项目组经过分析和讨论，决定采用面向对象开发技术对系统各项需求建模。

2016年【问题1】（7分） 用例建模用来描述待开发系统的功能需求，主要元素是用例和参与者。请根据题目所述需求，说明教学服务系统中有哪些参与者。

【参考答案】 学生、教师、管理员、时间、打印机。

【试题解析】

参与者是指系统以外的，需要使用系统或与系统交互的事物，包括：人或组织、设备、外部系统等。在本题中，较为容易识别的参与者包括：学生、教师、管理员，比较隐晦的参与者包括：时间、打印机。

2016年【问题2】（7分）

用例是对系统行为的动态描述，用例获取是需求分析阶段的主要任务之一。请指出在面向对象系统建模中，用例之间的关系有哪几种类型？对题目所述教学服务系统的需求建模时，“登录系统”用例与“注册课程”用例之间、“参加考试”用例与“参加补考”用例之间的关系分别属于哪种类型？

【参考答案】用例之间的关系包括：包含、扩展、泛化。 “登录系统”用例与“注册课程”用例之间的关系为：包含关系。

“参加考试”用例与“参加补考”用例之间的关系为：扩展关系。

2016年【问题3】（11分）

类图主要用来描述系统的静态结构，是组件图和配置图的基础。请指出在面向对象系统建模中，类之间的关系有哪几种类型？对题目所述教学服务系统的需求建模时，类University与类Student之间、类University和类Department之间、类Student和类Course之间的关系分别属于哪种类型？

【参考答案】 类之间的关系包括：关联、聚合、组合、依赖、泛化、实现（可写可不写，因为实现是接口与类之间的关系，而接口是一种特殊的类）。

类University与类Student之间的关系是：聚合关系。 类University与类Department之间的关系是：组合关系。

类Student与类Course之间的关系是：关联关系。

【试题解析】 依赖关系：一个事物发生变化影响另一个事物。 泛化关系：特殊/一般关系。 关联关系：描述了一组链，链是对象之间的连接。

聚合关系：整体与部分生命周期不同。 组合关系：整体与部分生命周期相同。 实现关系：接口与类之间的关系。

* 最少知识原则（也称为迪米特法则）是面向对象设计原则之一，指一个软件实体应当尽可能少地与其他实体发生相互作用。这样，当一个实体被修改时，就会尽可能少地影响其他的实体。下列叙述中，“(31)”不符合最少知识原则。

2011年(31)

A.在类的划分上，应当尽量创建松耦合的类

B.在类的设计上，只要有可能，一个类型应当设计成不变类

C.在类的结构设计上，每个类都应当尽可能提高对其属性和方法的访问权限

D.在对其他类的引用上，一个对象对其他对象的引用应当降到最低

【答案】C 【解析】

常用的面向对象设计原则包括开闭原则、里氏替换原则、依赖倒置原则、组合/聚合复用原则、接口隔离原则和最少知识原则等。这些设计原则首先都是面向复用的原则，遵循这些设计原则可以有效地提高系统的复用性，同时提高系统的可维护性。

最少知识原则（也称为迪米特法则）是面向对象设计原则之一，指一个软件实体应当尽可能少地与其他实体发生相互作用。这样，当一个实体被修改时，就会尽可能少地影响其他的实体。

最少知识原则主要用于控制信息的过载。在将最少知识原则运用到系统设计中时， 要注意以下几点：

①在类的划分上，应当尽量创建松耦合的类，类之间的耦合度越低，就越有利于复用。一个处在松稱合中的类一旦被修改，不会对关联的类造成太大波动。

②在类的结构设计上，每个类都应当尽量降低其属性和方法的访问权限。 ③在类的设计上，只要有可能，一个类型应当设计成不变类。

④在对其他类的引用上，一个对象对其他对象的引用应当降到最低。

* 对于违反里氏替换原则的两个类A和B，可以采用的候选解决方案中，正确的是(35).

2012年(35)

A.尽量将一些需要扩展的类或者存在变化的类设计为抽象类或者接口，并将其作为基类，在程序中尽量使用基类对象进行编程

B.创建一个新的抽象类C，作为两个具体类的超类，将A和B共同的行为移动到C中，从而解决A和B行为不完全一致的问题

C.将B到A的继承关系改成组合关系

D.区分是“Is-a”还是“Has-a”。如果是Is-a，可以使用继承关系，如果是Has-a,应该改成组合或聚合关系

【答案】A 【解析】 里氏替换原则是面向对象设计原则之一，由Barbara liskov提出，其基本思想是，一个软件实体如果使用的是一个基类对象，那么一定适用于其子类对象，而且觉察不出基类对象和子类对象的区别，即把基类都替换成它的子类，程序的行为没有变化。反过来则不一定成立，如果一个软件实体使用的是一个子类对象，那么它不一定适用于基类对象。

在运用里氏替换原则时，尽量将一些需要扩展的类或者存在变化的类设计为抽象类或者接口，并将其作为基类，在程序中尽量使用基类对象进行编程。由于子类继承基类并实现其中的方法，程序运行时，子类对象可以替换基类对象，如果需要对类的行为进行修改，可以扩展基类，增加新的子类，而无需修改调用该基类对象的代码。

* 在面向对象设计的原则中、（36）原则是指抽象不应该依赖予细节，细节应该依赖于抽象，即应针对接口编程，而不是针对实现编程。

2015年(36)

A.开闭

B.里氏替换

C.最少知识

D.依赖倒置

【答案】D 【解析】

依赖倒置原则是指抽象不应该依赖于细节，细节应当依赖于抽象。换言之，要针对接口编程，而不是针对实现编程。在程序代码中传递参数时或在组合(或聚合)关系中，尽量引用层次高的抽象层类，即使用接口和抽象类进行变量类型声明、参数类型声明和方法返回类型声明，以及数据类型的转换等，而不要用具体类来做这些事情。为了确保该原则的应用，一个具体类应当只实现接口和抽象类中声明过的方法，而不要给出多余的方法，否则，将无法调用到在子类中增加的新方法。

实现开闭原则的关键是抽象化，并且从抽象化导出具体化实现，如果说开闭原则是OOD的目标的话，那么依赖倒置原则就是OOD的主要机制。有了抽象层，可以使得系统具有很好的灵活性，在程序中尽量使用抽象层进行编程，而将具体类写在配置文件中，这样，如果系统行为发生变化，则只需要扩展抽象层，并修改配置文件，而无须修改原有系统的源代码，在不修改的情况下来扩展系统功能，满足开闭原则的要求。依赖倒置原则是COM、CORBA、EJB、Spring等技术和框架背后的基本原则之一。

* 用例（usecase)用来描述系统对事件做出响应时所采取的行动。用例之间是具有相关性的。在一个“订单输入子系统”中，创建新订单和更新订单都需要核查用户账号是否正确。用例“创建新订单“、“更新订单”与用例“核查户账号”之间是（32）关系。

2009年(32)

A.包含（include)

B.扩展（extend)

C.分类（classification)

D.聚集（aggregation)

【答案】A 【解析】

用例是在系统中执行的一系列动作，这些动作将生成特定参与者可见的价值结果。它确定了一个和系统参与者进行交互，并可由系统执行的动作序列。用例模型描述的是外部执行者（Actor)所理解的系统功能。用例模型用于需求分析阶段，它的建立是系统开发者和用户反复讨论的结果，表明了开发者和用户对需求规格达成的共识。两个用例之间的关系主要有两种情况：一种是用于重用的包含关系，用构造型include表示；另一种是用于分离出不同行为的扩展，用构造型extend表示。①包含关系：当可以从两个或两个以上的原始用例中提取公共行为，或者发现能够使用一个构件来实现某一个用例的部分功能是很重要的事时，应该使用包含关系来表示它们。

②扩展关系：如果一个用例明显地混合了两种或两种以上的不同场景，即根据情况可能发生多种事情，可以断定将这个用例分为一个主用例和一个或多个辅用例描述可能更加清晰。

* 面向对象的设计模型包含以（33）表示的软件体系结构图，以(34)表示的用例实现图，完整精确的类图，针对复杂对象的状态图和用以描述流程化处理的活动图等。

2009年(33)

A.部署图

B.包图

C.协同图

D.交互图

2009年(34)

A.部署图

B.包图

C.协同图

D.交互图

【答案】B D 【解析】

面向对象的设计模型包含以包图表示的软件体系结构图，以交互图表示的用例实现图，完整精确的类图，针对复杂对象的状态图和用以描述流程化处理的活动图等。

* 在UML提供的系统视图中，（35）是逻辑视图的一次执行实例，描述了并发与同步结构；（36）是最基本的需求分析模型。

2014年(35)

A.进程视图

B.实现视图

C.部署视图

D.用例视图

2014年(36)

A.进程视图

B.实现视图

C.部署视图

D.用例视图

【答案】A D 【解析】

UML对系统架构的定义是系统的组织结构，包括系统分解的组成部分，以及它们的关联性、交互机制和指导原则等提供系统设计的信息。具体来说，就是指以下5个系统视图：

①逻辑视图。逻辑视图也称为设计视图，它表示了设计模型中在架构方面具有重要意义的部分，即类、子系统、包和用例实现的子集。

②进程视图。进程视图是可执行线程和进程作为活动类的建模，它是逻辑视图的一次执行实例，描述了并发与同步结构。

③实现视图。实现视图对组成基于系统的物理代码的文件和构件进行建模。

④部署视图。部署视图把构件部署到一组物理节点上，表示软件到硬件的映射和分布结构。 ⑤用例视图。用例视图是最基本的需求分析模型。

* 用例（use case）用来描述系统对事件做出响应时所采取的行动。用例之间是具有相关性的。在一个会员管理系统中，会员注册时可以采用电话和邮件两种方式。用例“会员注册”和“电话注册”、“邮件注册”之间是（31）关系。

2015年(31)

A.包含（include）

B.扩展(extend)

C.泛化（generalize）

D.依赖（depends on）

【答案】C 【解析】 用例之间的关系主要有包含、扩展和泛化，利用这些关系，把一些公共的信息抽取出来，以便于复用，使得用例模型更易于维护。

①包含关系。当可以从两个或两个以上的用例中提取公共行为时，应该使用包含关系来表示它们。其中这个提取出来的公共用例称为抽象用例，而把原始用例称为基本用例或基础用例。

②扩展关系。如果一个用例明显地混合了两种或两种以上的不同场景，即根据情况可能发生多种分支，则可以将这个用例分为一个基本用例和一个或多个扩展用例，这样使描述可能更加清晰。

③泛化关系。当多个用例共同拥有一种类似的结构和行为的时候，可以将它们的共性抽象成为父用例，其他的用例作为泛化关系中的子用例。在用例的泛化关系中，子用例是父用例的一种特殊形式，子用例继承了父用例所有的结构、行为和关系。

* 面向对象的分析模型主要由（31）、用例与用例图、领域概念模型构成；设计模型则包含以包图表示的软件体系结构图、以交互图表示的（32）、完整精确的类图、针对复杂对象的状态图和描述流程化处理过程的（33）等。

2016年(31)

A.业务活动图

B.顶层架构图

C.数据流模型

D.实体联系图

2016年(32)

A.功能分解图

B.时序关系图

C.用例实现图

D.软件部署图

2016年(33)

A.序列图

B.协作图

C.流程图

D.活动图

【答案】B C D

【解析】本题考查的是教程“4.4.2 面向对象的分析设计”的内容。面向对象的分析模型主要由顶层架构图、用例与用例图、领域概念模型构成。设计模型则包含以包图表示的软件体系结构图、以交互图表示的用例实现图、完整精确的类图、针对复杂对象的状态图和用以描述流程化处理过程的活动图等。

* 在面向对象设计中，(31)可以实现界面控制、外部接口和环境隔离。(32)作为完成用例业务的责任承担者，协调、控制其他类共同完成用例规定的功能或行为。

2013年(31)

A.实体类

B.控制类

C.边界类

D.交互类

2013年(32)

A.实体类

B.控制类

C.边界类

D.交互类

【答案】C B 【解析】 类封装了信息和行为，是面向对象的重要组成部分。在面向对象设计中，类可以分为三种类型：实体类、边界类和控制类。

①实体类映射需求中的每个实体，实体类保存需要存储在永久存储体中的信息。实体类是对用户来说最有意义的类，通常采用业务领域术语命名，一般来说是一个名词，在用例模型向领域模型转化中，一个参与者一般对应于实体类。

②控制类是用于控制用例工作的类，一般是由动宾结构的短语（“动词+名词”或“名词+动词”）转化来的名词。控制类用于对一个或几个用例所特有的控制行为进行建模，控制对象通常控制其他对象，因此它们的行为具有协调性。

③边界类用于封装在用例内、外流动的信息或数据流。边界类是一种用于对系统外部环境与其内部运作之间的交互进行建模的类。边界对象将系统与其外部环境的变更隔离开，使这些变更不会对系统其他部分造成影响。

* 在面向对象设计中，用于描述目标软件与外部环境之间交互的类被称为（29)， 它可以（30)。

2011年(29)

A.实体类

B.边界类

C.模型类

D.控制类

2011年(30)

A.表示目标软件系统中具有持久意义的信息项及其操作

B.协调、控制其他类完成用例规定的功能或行为

C.实现目标软件系统与外部系统或外部设备之间的信息交流和互操作

D.分解任务并把子任务分派给适当的辅助类

【答案】B C 【解析】本题考查面向对象开发方法的基础知识。

类封装了信息和行为，是面向对象的重要组成部分。设计类是面向对象设计过程中最重要的组成部分，也是最复杂和最耗时的部分。在面向对象设计过程中，类可以分为三种类型：实体类、边界类和控制类。

实体类映射需求中的每个实体。实体类保存需要存储在永久存储体中的信息。实体类是对用户来说最有意义的类，通常采用业务领域术语命名，一般来说是一个名词，在用例模型向领域模型的转化中，参与者一般对应于实体类。

控制类是用于控制用例工作的类，一般是由动宾结构的短语（“动词+名词”或“名词+动词”）转化而来的名词。控制类用于对一个或几个用例所特有的控制行为进行建模，控制对象（控制类的实例）通常控制其他对象。因此它们的行为具有协调性。

边界类用于封装在用例内、外流动的信息或数据流。边界类是一种用于对系统外部环境与其内部运作之间的交互进行建模的类，用于实现目标软件系统与外部系统或外部设备之间的信息交流和互操作。

## 原型法

### 知识点

原型是软件系统的初始版本，用来演示概念并尝试设计选择，通常用来发现更多的问题和可能的解决方案。快速迭代式的原型开发能够有效控制成本，根据原型与最终产品之间的关系，原型开发分为三类：抛弃式原型开发利用原型验证和澄清系统的需求描述，重新构造系统：演化式原型开发逐步改进和细化原型，将原型进化直至产生出目标系统；增量式原型开发在建立软件总体设计的基础上，采用增量开发方法，使原型成为最终系统。

### 试题

* 快速迭代式的原型开发能够有效控制成本，(40)是指在开发过程中逐步改进和细化原型直至产生出目标系统。

2011年(40)

A.可视化原型开发

B.抛弃式原型开发

C.演化式原型开发

D.量式原型开发

【答案】C 【解析】本题考查应用系统开发方法的基础知识。

原型是软件系统的初始版本，用来演示概念并尝试设计选择，通常用来发现更多的问题和可能的解决方案。快速迭代式的原型开发能够有效控制成本，根据原型与最终产品之间的关系，原型开发分为三类：抛弃式原型开发利用原型验证和澄清系统的需求描述，重新构造系统：演化式原型开发逐步改进和细化原型，将原型进化直至产生出目标系统；增量式原型开发在建立软件总体设计的基础上，采用增量开发方法，使原型成为最终系统。

## 逆向工程

### 知识点

### 试题

* 逆向工程导出的信息可以分为实现级、结构级、功能级和领域级四个抽象层次。程序的抽象语法树属于（29)；反映程序分量之间相互依赖关系的信息属于（30)。

2013年(29)

A.实现级

B.结构级

C.功能级

D.领域级

2013年(30)

A.实现级

B.结构级

C.功能级

D.领域级

【答案】A B 【解析】

逆向工程与重构工程是目前预防性维护采用的主要技术。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。逆向工程导出的信息可以分为如下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。

③功能级：包括反映程序段功能及程序段之间关系的信息。

④领域级：包括反映程序分量或程序诸实体与应用领域概念之间对应关系的信息。显然，上述信息的抽象级别越高，它与代码的距离就越远，通过逆向工程恢复的难度亦越大，而自动工具支持的可能性相对变小，要求人参与判断和推理的工作增多。

* 逆向工程导出的信息可以分为4个抽象层次，其中（28）可以抽象出程序的抽象语法树、符号表等信息；(29)可以抽象出反应程序段功能及程序段之间关系的信息。

2009年(28)

A.实现级

B.结构级

C.功能级

D.领域级

2009年(29)

A.实现级

B.结构级

C.功能级

D.领域级

【答案】A C 【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。 ③功能级：包括反映程序段功能及程序段之间关系的信息。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。

# 软件开发模型

软件产品从形成概念开始，经过开发、使用和维护，直到最后退役的全过程成为软件生存周期。一个完整的软件生存周期是以需求为出发点，从提出软件开发计划的那一刻开始，直到软件在实际应用中完全报废为止。软件生存周期的提出了是为了更好地管理、维护和升级软件，其中更大的意义在于管理软件开发的步骤和方法。

软件生存周期模型又称软件开发模型（software develop model)或软件过程模型 (software process model),它是从某个特定角度提出的软件过程的简化描述。软件生存周期模型主要有瀑布模型、演化模型、原型模型、螺旋模型喷泉模型和基于可重用构件的模型等。

瀑布模型是最早使用的软件生存周期模型之一。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，每一个阶段都是建立在前一个阶段的正确结果之上，前一个阶段的错误和疏漏会隐蔽地带入后一个阶段。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。

演化模型主要针对事先不能完整定义需求的软件开发，是在快速开发一个原型的基础上，根据用户在调用原型的过程中提出的反馈意见和建议，对原型进行改进，获得原型的新版本，重复这一过程，直到演化成最终的软件产品。演化模型的主要优点是，任何功能一经开发就能进入测试，以便验证是否符合产品需求，可以帮助引导出高质量的产品要求。其主要缺点是，如果不控制地让用户接触开发中尚未稳定的功能，可能对开发人员产生负面的影响。

* (28)把整个软件开发流程分成多个阶段，每一个阶段都由目标设定、风险分析、开发和有效性验证以及评审构成。

2010年(28)

A.原型模型

B.瀑布模型

C.螺旋模型

D.V模型

【答案】C 【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。

* 以下关于软件生存周期模型的叙述，正确的是(25)。

2012年(25)

A.在瀑布模型中/前一个阶段的错误和疏漏会隐蔽地带到后一个阶段

B.在任何情况下使用演化模型，都能在一定周期内由原型演化到最终产品

C.软件生存周期模型的主要目标是为了加快软件开发的速度

D.当一个软件系统的生存周期结束之后，它就进入到一个新的生存周期模型

【答案】A 【解析】

软件产品从形成概念开始，经过开发、使用和维护，直到最后退役的全过程成为软件生存周期。一个完整的软件生存周期是以需求为出发点，从提出软件开发计划的那一刻开始，直到软件在实际应用中完全报废为止。软件生存周期的提出了是为了更好地管理、维护和升级软件，其中更大的意义在于管理软件开发的步骤和方法。

软件生存周期模型又称软件开发模型（software develop model)或软件过程模型 (software process model),它是从某个特定角度提出的软件过程的简化描述。软件生存周期模型主要有瀑布模型、演化模型、原型模型、螺旋模型喷泉模型和基于可重用构件的模型等。

瀑布模型是最早使用的软件生存周期模型之一。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，每一个阶段都是建立在前一个阶段的正确结果之上，前一个阶段的错误和疏漏会隐蔽地带入后一个阶段。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。

演化模型主要针对事先不能完整定义需求的软件开发，是在快速开发一个原型的基础上，根据用户在调用原型的过程中提出的反馈意见和建议，对原型进行改进，获得原型的新版本，重复这一过程，直到演化成最终的软件产品。演化模型的主要优点是，任何功能一经开发就能进入测试，以便验证是否符合产品需求，可以帮助引导出高质量的产品要求。其主要缺点是，如果不控制地让用户接触开发中尚未稳定的功能，可能对开发人员及永固都会产生负面的影响。

* 快速应用开发（Rapid Application Development，RAD)通过使用基于（29)的开发方法获得快速开发。当(30)时，最适合于采用RAD方法。

2012年(29)

A.用例

B.数据结构

C.剧情

D.构件

2012年(30)

A.一个新系统要采用很多新技术

B.新系统与现有系统有较高的互操作性

C.系统模块化程度较高

D.用户不能很好地参与到需求分析中

【答案】D C 【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。

③RAD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。

## 瀑布模型

## 演化模型

### 知识点

把对遗留系统的评价结果分列在坐标的4个象限内。对处在不同象限的遗留系统采取不同的演化策略。

①淘汰策略。第四象限为低水平、低价值区，即遗留系统的技术含量较低，且具有较低的业务价值。对这种遗留系统的演化策略为淘汰，即全面重新开发新的系统以代替遗留系统。完全淘汰是一种极端性策略，一般是企业的业务产生了根本变化，遗留系统已经基本上不再适应企业运作的需要；或者是遗留系统的维护人员、维护文档资料都丢失了。经过评价，发现将遗留系统完全淘汰，开发全新的系统比改造旧系统从成本上考虑更合算。

②继承策略。第二象限为低水平、高价值区，即遗留系统的技术含量较低，已经满足企业运作的功能或性能要求，但具有较高的商业价值，目前企业的业务尚紧密依赖该系统。称这种遗留系统的演化策略为继承。在开发新系统时，需要完全兼容遗留系统的功能模型和数据模型。为了保证业务的连续性，新老系统必须并行运行一段时间，再逐渐切换到新系统上运行。

③改造策略。第一象限为高水平、高价值区，即遗留系统的技术含量较高，本身还有强大的生命力。系统具有较高的业务价值.基本上能够满足企业业务运作和决策支持的需要。这种系统可能建成的时间还很短，称这种遗留系统的演化策略为改造。改造包括系统功能的增强和数据模型的改造两个方面。系统功能的增强是指在原有系统的基础上增加新的应用要求，对遗留系统本身不做改变；数据模型的改造是指将遗留系统的旧的数据模型向新的数据模型的转化。

④集成策略。第三象限为高水平、低价值区，即遗留系统的技术含量较高，但其业务价值较低，可能只完成某个部门(或子公司)的业务管理。这种系统在各自的局部领域里工作良好，但对于整个企业来说，存在多个这样的系统，不同的系统基于不同的平台、不同的数据模型，形成了一个个信息孤岛，对这种遗留系统的演化策略为集成。

### 试题

* 对于遗留系统的评价框架如下图所示，那么处于“高水平、低价值”区的遗留系统适合于采用的演化策略为（37）。

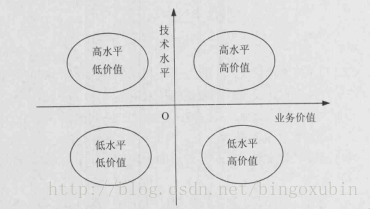
2015年(37)

A.淘汰

B.继承

C.改造

D.集成



【答案】D 【解析】 把对遗留系统的评价结果分列在坐标的4个象限内。对处在不同象限的遗留系统采取不同的演化策略。

①淘汰策略。第四象限为低水平、低价值区，即遗留系统的技术含量较低，且具有较低的业务价值。对这种遗留系统的演化策略为淘汰，即全面重新开发新的系统以代替遗留系统。完全淘汰是一种极端性策略，一般是企业的业务产生了根本变化，遗留系统已经基本上不再适应企业运作的需要；或者是遗留系统的维护人员、维护文档资料都丢失了。经过评价，发现将遗留系统完全淘汰，开发全新的系统比改造旧系统从成本上考虑更合算。

②继承策略。第二象限为低水平、高价值区，即遗留系统的技术含量较低，已经满足企业运作的功能或性能要求，但具有较高的商业价值，目前企业的业务尚紧密依赖该系统。称这种遗留系统的演化策略为继承。在开发新系统时，需要完全兼容遗留系统的功能模型和数据模型。为了保证业务的连续性，新老系统必须并行运行一段时间，再逐渐切换到新系统上运行。

③改造策略。第一象限为高水平、高价值区，即遗留系统的技术含量较高，本身还有强大的生命力。系统具有较高的业务价值.基本上能够满足企业业务运作和决策支持的需要。这种系统可能建成的时间还很短，称这种遗留系统的演化策略为改造。改造包括系统功能的增强和数据模型的改造两个方面。系统功能的增强是指在原有系统的基础上增加新的应用要求，对遗留系统本身不做改变；数据模型的改造是指将遗留系统的旧的数据模型向新的数据模型的转化。

④集成策略。第三象限为高水平、低价值区，即遗留系统的技术含量较高，但其业务价值较低，可能只完成某个部门(或子公司)的业务管理。这种系统在各自的局部领域里工作良好，但对于整个企业来说，存在多个这样的系统，不同的系统基于不同的平台、不同的数据模型，形成了一个个信息孤岛，对这种遗留系统的演化策略为集成。

* 遗留系统的演化可以采用淘汰、继承、改造和集成四种策略。若企业中的遗留系统技术含量较高，业务价值较低，在局部领域中工作良好，形成了一个个信息孤岛时，适合于采用（28)演化策略。

2013年(28)

A.淘汰

B.继承

C.改造

D.集成

【答案】D 【解析】 遗留系统的演化可以采用淘汰、继承、改造和集成四种策略。

淘汰策略适用于技术含量较低，且具有较低的业务价值的遗留系统，即通过全面重新开发新的系统以代替遗留系统。

若遗留系统的技术含量较低，能满足企业运作的功能或性能要求，但具布较高的商业机制，目前企业的业务上紧密依赖该系统，这种遗留系统的演化策略为继承。在开发新系统时，需要完全兼容遗留系统的功能模型和数据模型。为了保证业务的连续性，新老系统必须并行运行一段时间。

对于技术含量较高，本身还有极大的生命力，又具有较高的业务价值，基本上能够满足企业业务运作和决策支持需要的遗留系统，采用改造策略进行演化。改造包括系统功能的增强和数据模型的改造两个方面。

遗留系统的技术含量较高，但其业务价值较低，可能只完成某个部门（或子公司）的业务管理。这种系统在各自局部领域里工作良好，但对于整个企业来说，存在多个这样的系统，不同的系统基于不同的平台、不同的数据模型，形成了一个个信息孤岛。对于这种遗留系统的演化策略为集成。

## 螺旋模型

* 螺旋模型将整个软件开发过程分为多个阶段，每个阶段都由目标设定、（26)、开发和有效性验证以及评审4个部分组成。

2012年(26)

A.需求分析

B.风险分析

C.系统设计

D.架构设计

【答案】B 【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。

* 螺旋模型在（26）的基础上扩展而成。

2016年(26)

A.瀑布模型

B.原型模型

C.快速模型

D.面向对象模型

【答案】B

【解析】

## 增量模型

## 构件组装模型

## 统一过程

## 敏捷方法

### 知识点

敏捷方法是从20世纪90年代开始逐渐引起广泛关注的一种新型软件开发方法，以应对快速变化的需求。敏捷方法是一种以人为核心、迭代、循序渐进的开发方法。敏捷方法强调，让客户满意和软件尽早增量发布：小而高度自主的项目团队；非正式g方法：最小化软件工程工作产品以及整体精简开发。与传统方法相比，敏捷开发方法比较适合需求变化较大或者开发前期需求不是很清晰的项目，以它的灵活性来适应需求的变化。

### 试题

* 下列关于不同软件开发方法所使用的模型的描述中，正确的是(32)。

2010年(32)

A.在进行结构化分析时，必须使用数据流图和软件结构图这两种模型

B.采用面向对象开发方法时，可以使用状态图和活动图对系统的动态行为进行建模

C.实体联系图(E-R图)是在数据库逻辑结构设计时才开始创建的模型

D.UML的活动图与程序流程图的表达能力等价

【答案】B 【解析】

结构化分析方法是一种面向数据流的需求分析方法，其基本思想是自顶向下逐层分解。数据流图是进行结构化分析时所使用的模型，其基本成分包括数据流、加工、数据

存储和外部实体。在进行结构化设计时，通过对数据流图进行变换分析和事务分析可以导出程序结构图。

数据库设计可以分为4个主要阶段：①用户需求分析。数据库设计人员采用一定的辅助工具对应用对象的功能、性能、限制等要求所进行的科学分析。②概念设计。概念结构设计是对信息分析和定义，如视图模型化、视图分析和汇总。对应用对象精确地抽象、概括而形成的独立于计算机系统的企业信息模型。描述概念模型的较理想的工具是E-R图。③逻辑设计。将抽象的概念模型转化为与选用的DBMS产品所支持的数据模型相符合的逻辑模型，它是物理设计的基础。包括模式初始设计、子模式设计、应用程序设计、模式评价以及模式求精。④物理设计。逻辑模型在计算机中的具体实现方案。

UML是面向对象软件的标准化建模语言，其中状态图、活动图、顺序图和通信图可以用来对系统的动态行为进行建模。活动图展现了在系统内从一个活动到另一个活动的流程。活动图强调对象之间的控制流程。在活动图上可以表示分支和汇合。活动图与传统的程序流程图是不等价的。

* (25)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。.

2009年(25)

A.信息工程

B.结构化

C.面向对象

D.敏捷

【答案】D 【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。

* 下列关于敏捷方法的叙述中，错误的是（29）。

2014年(29)

A.与传统方法相比，敏捷方法比较适合需求变化大或者开发前期对需求不是很清晰的项目

B.敏捷方法尤其适合于开发团队比较庞大的项目

C.敏捷方法的思想是适应性，而不是预设性

D.敏捷方法以原型开发思想为基础，采用迭代式增量开发

【答案】B 【解析】本题考査敏捷方法的相关概念。

敏捷方法是从20世纪90年代开始逐渐引起广泛关注的一些新型软件开发方法，以应对快速变化的需求。敏捷方法的核心思想主要有以下三点。

①敏捷方法是“适应性”而非“预设性”的。传统方法试图对一个软件开发项目在很长的时间跨度内做出详细的计划，然后依计划进行开发。这类方法在计划制定完成后拒绝变化。而敏捷方法则欢迎变化，其实它的目的就是成为适应变化的过程，甚至能允许改变自身来适应变化。

②敏捷方法是以人为本，而不是以过程为本。传统方法以过程为本，强调充分发挥人的特性，不去限制它，并且软件开发在无过程控制和过于严格繁琐的过程控制中取得一种平衡，以保证软件的质量。

③迭代增量式的幵发过程。敏捷方法以原型开发思想为基础，采用迭代增最式开发，发行版本小型化。

与RUP相比，敏捷方法的周期可能更短。敏捷方法在几周或者几个月的时间内完成相对较小的功能，强调的是能尽早将尽量小的可用的功能交付使用，并在整个项目周期中持续改善和增强，并且更加强调团队中的高度写作。相对而言，敏捷方法主要适合于以下场合：

①项目团队的人数不能太多，适合于规模较小的项目。

②项目经常发生变更。敏捷方法适用于需求萌动并且快速改变的情况，如果系统有比较高的关键性、可靠性、安全性方面的要求，则可能不完全适合。

③高风险项目的实施。 ④从组织结构的角度看，组织结构的文化、人员、沟通性决定了敏捷方法是否使用。

极限编程

XP (Extreme Programming，极限编程)在所有的敏捷型方法中，XP是最引人瞩目的。它源于Smalltalk圈子，特别是Kent Beck和Ward Cunningham在20世纪80年代末的密切合作。XP在一些对费用控制严格的公司中的使用，已经被证明是非常有效的。

Scrum

Scrum是橄榄球比赛中“争球”的意思，你可以脑补争球时的敏捷、激情和你争我夺。现在Scrum是广泛应用于IT界的一套项目管理工具

Scrum的核心，是把整个项目分成若干个冲刺，每次2-4周，冲完一次再来一次。

Scrum的本质，是把一次漫长的长跑，分割成一段段全力以赴的冲刺，通过一套标准流程方法，提高效率。

Scrum是由三个角色（产品负责人，Scrum Master，团队成员）、四个仪式（冲刺计划会，每日站会，冲刺评审会，冲刺回顾会）和三个物件（产品积压，冲刺积压，燃尽图）组成的一套项目管理方法。

产品负责人，召开冲刺计划会，定下三件事：冲刺目标。定下目标后，把它们从“产品积压”，移入“冲刺积压”。

冲刺方法。选定冲刺方法。

分配任务。团队成员若干人，在6个步骤中，各自主动领取任务。

燃尽，是烧完了的意思。随着时间推移，剩余工作量越来越少。把计划进度，画成一根从左上到右下的直线。然后，把实际进度用红线标在旁边，看着工作量像蜡烛一样，不断燃尽。

冲刺评审会，由主编负责主持，一起审阅最终交付的文章。冲刺回顾会，讨论开始做什么，停止做什么，继续做什么。

水晶方法

Cockburn的水晶系列方法，水晶系列方法是由AlistairCockburn提出的。它与XP方法一样，都有以人为中心的理念，但在实践上有所不同。Alistair考虑到人们一般很难严格遵循一个纪律约束很强的过程，因此，与XP的高度纪律性不同，Alistair探索了用最少纪律约束而仍能成功的方法，从而在产出效率与易于运作上达到一种平衡。也就是说，虽然水晶系列不如XP那样的产出效率，但会有更多的人能够接受并遵循它。

开放源码界的开发方法

开放式源码，这里提到的开放式源码指的是开放源码界所用的一种运作方式。开放式源码项目有一个特别之处，就是程序开发人员在地域上分布很广，这使得它和其他敏捷方法不同，因为一般的敏捷方法都强调项目组成员在同一地点工作。开放源码的一个突出特点就是查错排障(debug)的高度并行性，任何人发现了错误都可将改正源码的“补丁”文件发给维护者。然后由维护者将这些“补丁”或是新增的代码并入源码库。

FDD

Coad的功用驱动开发方法(FDD-Feature Driven Development)，FDD是由Jeff DeLuca和大师Peter Coad提出来的。像其他方法一样，它致力于短时的迭代阶段和可见可用的功能。在FDD中，一个迭代周期一般是两周。在FDD中，编程开发人员分成两类：首席程序员和“类”程序员(classowner)。首席程序员是最富有经验的开发人员，他们是项目的协调者、设计者和指导者，而“类”程序员则主要做源码编写。

其他敏捷方法

* （27）适用于程序开发人员在地域上分布很广的开发团队。（28）中，编程开发人员分成首席程序员和“类”程序员。

2016年(27)

A.水晶系列（Crystal）开发方法

B.开放式源码（Open source）开发方法

C.SCRUM开发方法

D.功用驱动开发方法（FDD）

2016年(28)

A.自适应软件开发（ASD）

B.极限编程（XP）开发方法

C.开放统—过程开发方法（OpenUP）

D.功用驱动开发方法（FDD）

【答案】A D

【解析】 (1) XP (Extreme Programming，极限编程)在所有的敏捷型方法中，XP是最引人瞩目的。它源于Smalltalk圈子，特别是Kent Beck和Ward

Cunningham在20世纪80年代末的密切合作。XP在一些对费用控制严格的公司中的使用，已经被证明是非常有效的。 (2)Cockburn的水晶系列方法，水晶系列方法是由AlistairCockburn提出的。它与XP方法一样，都有以人为中心的理念，但在实践上有所不同。Alistair考虑到人们一般很难严格遵循一个纪律约束很强的过程，因此，与XP的高度纪律性不同，Alistair探索了用最少纪律约束而仍能成功的方法，从而在产出效率与易于运作上达到一种平衡。也就是说，虽然水晶系列不如XP那样的产出效率，但会有更多的人能够接受并遵循它。(3)开放式源码，这里提到的开放式源码指的是开放源码界所用的一种运作方式。开放式源码项目有一个特别之处，就是程序开发人员在地域上分布很广，这使得它和其他敏捷方法不同，因为一般的敏捷方法都强调项目组成员在同一地点工作。开放源码的一个突出特点就是查错排障(debug)的高度并行性，任何人发现了错误都可将改正源码的“补丁”文件发给维护者。然后由维护者将这些“补丁”或是新增的代码并入源码库。(4)SCRUM。SCRUM己经出现很久了，像前面所论及的方法一样，该方法强调这样一个事实，即明确定义了的可重复的方法过程只限于在明确定义了的可重复的环境中，为明确定义了的可重复的人员所用，去解决明确定义了的可重复的问题。(5) Coad的功用驱动开发方法(FDD-Feature Driven Development)，FDD是由Jeff DeLuca和大师Peter Coad提出来的。像其他方法一样，它致力于短时的迭代阶段和可见可用的功能。在FDD中，一个迭代周期一般是两周。在FDD中，编程开发人员分成两类：首席程序员和“类”程序员(classowner)。首席程序员是最富有经验的开发人员，他们是项目的协调者、设计者和指导者，而“类”程序员则主要做源码编写。 (6)ASD方法，ASD (Adaptive Software Development)方法由JimHighsmith提出，其核心是三个非线性的、重叠的开发阶段：猜测、合作与学习。

# 需求开发

## 需求获取

* 下列关于联合需求计划(Joint Requirement Planning, JRP)的叙述中，不正确的是（26）。

2014年(26)

A.在JRP实施之前，应制定详细的议程，并严格遵照议程进行

B.在讨论期间尽量避免使用专业术语

C.JRP是一种相对来说成本较高但十分有效的需求获取方法

D.JRP的主要目的是对需求进行分析和验证

【答案】D 【解析】本题考查联合需求计划(JRP)的相关概念。

JRP是一个通过高度组织的群体会议来分析企业内的问题并获取需求的过程，它是联合应用开发（JAD)的-部分。

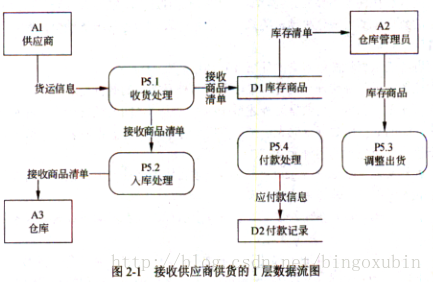
JRP的主要意图是收集需求，而不是对需求进行分析和验证。实施JRP时应把握以下主要原则：在JRP实施之前，应制定详细的议程，并严格遵照议程进行；按照既定的时间安排进行；尽量完整地记录会议期间的内容；在讨论期间尽量避免使用专业术语；充分运用解决冲突的技能；会议期间应设置充分的间歇时间；鼓励团队取得-致意见；保证参加JRP的所有人员能够遵守实现约定的规则。

## 需求分析

知识点

试题

* 试题二【说明】某公司正在研发一套新的库存管理系统。系统中一个关键事件是接收供应商供货。项目组系统分析员小王花了大量时间在仓库观察了整个事件的处理过程，并开发出该过程所执行活动的列表：供应商发送货物和商品清单，公司收到商品后执行收货处理，包括卸载商品、确定收到了订单上的商品、处理与供应商的分歧等。对于已有商品，调整其库存信息，对于新采购的商品，在库存中添加新的商品记录。收货完成后，系统执行入库处理，将商品放到仓库对应的货架上。在付款处理活动中，自动生成应付账款信息，如果查询到该供应商有待付款记录，则进行合并付款，付款完成后消除应付账款记录。最后，仓库管理员根据最新的库存商品，调整出货信息。小王根据自己观察的过程创建了该事件的1层数据流图，如图 2-1 所示。



【2014问题1】 请用300以内文字说明数据流图(Data Flow Diagram)的基本元素及其作用。

四种元素：

(1) External Agent(实体/外部代理)：定义位于项目范围之外，但与正在被研发的系统有交互关系的人、部门、外部系统或组织。

(2) Process(加工/处理)：在输入数据流或条件上执行，或者对输入数据流或条件做出响应的工作。

(3) Data Store(数据存储)：静止的数据，表示系统中需要保存的数据。

(4) Data Flow(数据流)：运动中的数据，表示到一个过程的数据输入，或者来自一个过程的数据输出。

本题考查系统过程建模的相关知识。 数据流图（Data Flow Diagram)从数据传递和加工角度，以图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程，是结构化系统分析方法的主要表达工具及用于表示软件模型的一种图示方法。为了表达数据处理过程的数据加工情况，用一个数据流图往往是不够的。层次结构的数据流图按照系统的层次结构进行逐步分解，并以分层的数据流图反映这种结构关系，能清楚地表达和容易理解整个系统。

本问题考查数据流图中包含的元素及其作用。

数据流图通过外部代理（实体）描述系统与外界之间的数据交互关系，内部的活动通过处理（加工）表示，用数据流描述系统中不同活动之间的数据传输内容和方向，需要持久化存储的数据用数据存储表示，一般用文件系统或者数据库表存储数据-数据流图中所包含的四种元素：

(1) 外部实体（External Agent)定义位于项目范围之外，但与正在被研发的系统有交互关系的人、部门、外部系统或组织； (2)加工（Process)在输入数据流或条件上执行，或者对输入数据流或条件做出响应的工作； (3) 数据存储（Data Store)描述静止的数据，表示系统中需要保存的数据； (4) 数据流（Data Flow)描述运动中的数据，表示到一个过程的数据输入，或者来自一个过程的数据输出。

【2014问题2】

数据流图在绘制过程中可能出现多种语法错误，请分析上图所示数据流图中哪些地方有错误，并分别说明错误的类型。

四种错误： 2014年(1)

D1到A2：缺少移动数据流的加工。 (2) P5.3：没有输出数据流，输入输出不平衡。 (3) P5.4：没有输入数据流，输入输出不平衡。

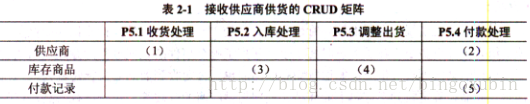
(4) D2：数据存储没有输出的数据流。

本问题考查数据流图绘制过程中常见的错误。

数据流图中的错误包括两类：第一类是逻辑错误，加工节点输入输出不平衡，包括黑洞、灰洞和无输入三种类型；第二类是语法错误，比如数据存储不完整、在数据存储与外部代理之间或者各自之间没有经过加工之间发生数据流等。根据图2-1所示，P5.3和P5.4属于逻辑错误，数据流图不平衡，D2没有输出数据流，D1到A2缺少加工等属于第二类错误。

【2014问题3】 系统建模过程中为了保证数据模型和过程模型的一致性，需要通过数据-过程-CRUD矩阵来实现数据模型和过程模型的同步，请在表 2-1

所示CRUD矩阵（1）～（5）中填入相关操作。



(1) R (2) R (3) CRU (4) RU (5) CRUD

CRUD(Create\Read\Update\Delete)矩阵用于检查系统建模过程中数据模型和过程模型的一致性，分别表示了加工对于数据的新增、读取、修改和删除四种操作。根据需求陈述和表2-1所示内容，P5.1收货处理和P5.4付款处理两个加工分别需要获得供应商的货运信息和付款记录，（1)和（2)处为读取操作（R)；P5.2入库处理中需要添加新的商品记录或者查询并修改现有商品的库存信息，（3)处为创建、读取和更新操作；P5.3凋整出货会读取并修改库存商品信息，（4)处为读取和修改操作（RU)；P5.4付款处理中除了生成付款记录、读取或修改付款记录外，对于己经付款的信息要消除应付款信息，所以（5)处为新増、读取、修改和删除四种操作（CRUD)。

* 下列叙述中，不满足好的需求陈述要求的是（27）。

2015年(27)

A.每一项需求都必须完整、准确地描述即将要开发的功能

B.需求必须能够在系统及其运行环境的能力和约束条件内实现

C.每一项需求记录的功能都必须是用户的真正的需要

D.所有需求都应被视为同等重要

【答案】D 【解析】 理想情况下，每一项用户、业务需求和功能需求都应具备下列性质。

完整性。每一项需求都必须完整地描述即将交付使用的功能。它必须包含开发人员设计和实现这项功能需要的所有信息。

正确性。每一项需求都必须准确地描述将要开发的功能。判断正确性的参考是需求来源，如实际用户和高级的系统需求。如果一项软件需求与其相对应的系统需求发生冲突，这是不正确的。

可行性。需求必须能够在系统及其运行环境的已知能力和约束条件内实现。

必要性。每一项需求记录的功能都必须是用户的真正需要，或者是为符合外部系统需求或标准而必须具备的功能。每项需求都必须来源于有权定义需求的一方。对每项需求都必须追溯至特定的客户需求的来源，例如用例、业务规则或者其他来源。

有优先次序。为每一项功能需求、特性或用例指定一个实现优先级，以表明它在产品的某一版本中的重要程度。如果所有需求都被视为同等重要，项目经理就很难采取措施应对预算削减、进度拖后、人员流失或开发过程中需求增加等情况。

无歧义。一项需求声明对所有读者应该只有一种一致的解释，编写需求时应该使用用户所在领域的、简洁明了的语言。应该在词汇表中列出所有专用的和可能让用户感到迷惑的术语。

可验证性。如果某项需求不可验证，那么判定其实现的正确与否就成了主观臆断，而不是客观分析。不完备、不一致、不可行或有歧义的需求也是不可验证的。

* 基于UML的需求分析过程的基本步骤为：利用（27)表示需求;利用（28)表示目标软件系统的总体架构。

2012年(27)

A.用例及用例图

B.包图及类图

C.剧情及序列图

D.组件图及部署图

2012年(28)

A.用例及用例图

B.包图及类图

C.剧情及序列图

D.组件图及部署图

【答案】A B 【解析】 基于UML的需求分析过程大致可分为以下步骤：

①利用用例及用例图表示需求。从业务需求描述出发获取执行者和场景；对场景进行汇总、分类、抽象，形成用例；确定执行者与用例、用例与用例图之间的关系，生成用例图。

②利用包图和类图表示目标软件系统的总体框架结构。根据领域知识、业务需求描述和既往经验设计目标软件系统的顶层架构；从业务需求描述中提取“关键概念”，形成领域概念模型；从概念模型和用例出发，研究系统中主要的类之间的关系，生成类图。

## 需求定义

## 需求管理

* （25）是关于需求管理正确的说法。

2016年(25)

A.为达到过程能力成熟度模型第二级，组织机构必须具有3个关键过程域

B.需求的稳定性不属于需求属性

C.需求变更的管理过程遵循变更分析和成本计算、问题分析和变更描述、变更实现的顺序

D.变更控制委员会对项目中任何基线工作产品的变更都可以做出决定

【答案】D

【解析】 过程能力成熟度模型（CMM）在软件开发机构中被广泛用来指导软件过程改进。该模型描述了软件处理能力的5个成熟级别。为了达到过程能力成熟度模型的第二级，组织机构必须具有 6 个关键过程域 KPA（Key ProcessAreas）。故A选项错误。除了文本，每一个功能需求应该有一些相关的信息与它联系，我们把这些信息称为需求属性。对于一个大型的复杂项目来说，丰富的属性类别显得尤为重要。例如，在文档中考虑和明确如下属性：创建需求的时间、需求的版本号、创建需求的作者、负责认可该软件需求的人员、需求状态、需求的原因和根据、需求涉及的子系统、需求涉及的产品版本号、使用的验证方法或者接受的测试标准、产品的优先级或者重要程度、需求的稳定性。故B选项错误。需求的变更遵循以下流程：(1)问题分析和变更描述。这是识别和分析需求问题或者一份明确的变更提议，以检查它的有效性，从而产生一个更明确的需求变更提议。

(2)变更分析和成本计算。使用可追溯性信息和系统需求的一般知识，对需求变更提议进行影响分析和评估。变更成本计算应该包括对需求文档的修改、系统修改的设计和实现的成本。一旦分析完成并且被确认，应该进行是否执行这一变更的决策。(3)变更实现。这要求需求文档和系统设计以及实现都要同时修改。如果先对系统的程序做变更，然后再修改需求文档，这几乎不可避免地会出现需求文档和程序的不一致。故C选项错误。

# 基于构件开发

* 基于构件的开发模型包括软件的需求分析定义、(35)、(36)、(37)，以及测试和发布5个顺序执行的阶段。

2009年(35)

A.构件接口设计

B.体系结构设计

C.元数据设计

D.集成环境设计

2009年(36)

A.数据库建模

B.业务过程建模

C.对象建模

D.构件库建立

2009年(37)

A.应用软件构建

B.构件配置管理

C.构件单元测试

D.构件编码实现

【答案】B D A 【解析】本题考查基于构件的软件开发模型的基础知识。

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，通过组合手段髙效率、髙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。

## 软件重用

* 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。软件元素包括（34）、测试用例和领域知识等。

2016年(34)

A.项目范围定义、需求分析文档、设计文档

B.需求分析文档、设计文档、程序代码

C.设计文档、程序代码、界面原型

D.程序代码、界面原型、数据表结构

【答案】B

【解析】本题考查的是教程“4.5软件的重用”的内容。

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。软件元素包括需求分析文档、设计过程、设计文档、程序代码、测试用例、领域知识等。对于新的软件开发项目而言，它们或者是构成整个目标软件系统的部件，或者在软件开发过程中发挥某种作用。通常将这些软件元素称为软部件。

* 软件的横向重用是指重用不同应用领域中的软件元素。(31)是一种典型的、原始的横向重用机制。

2010年(31)

A.对象

B.构件

C.标准函数库

D.设计模式

【答案】C 【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，软件重用可以区别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部件重用。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部件的可重用性。

## 构件标准

* 以下关于软件构件及其接口的叙述，错误的是（38）.

2009年(38)

A.构件是软件系统中相对独立且具有一定意义的构成成分

B.构件在容器中进行管理并获取其属性或者服务

C.构件不允许外部对所支持的接口进行动态发现或调用

D.构件可以基于对象实现，也可以不基于对象实现试

【答案】C 【解析】本题考查软件构件的基本概念。

软件构件是软件系统中具有一定意义的、相对独立的可重用单元。与对象相比，构件可以基于对象实现，也可以不作为对象实现。构件需要在容器中管理并获取容器提供的服务；客户程序可以在运行状态下利用接口动态确定构件所支持的功能并调用。

## 构件获取

## 构件管理

## 构件重用

# 开发工具

* 处理流程设计是系统设计的重要内容。以下关于处理流程设计工具的叙述中，不正确的是（30）。

2015年(30)

A.程序流程图（PFD）用于描速系统中每个模块的输入，输出和数据加工

B.N-S图容易表示嵌套关系和层次关系，并具有强烈的结构化特征

C.IPO图的主体是处理过程说明，可以采用流程图、判定树/表等来进行描述

D.问题分析图（PAD）包含5种基本控制结构，并允许递归使用

【答案】A 【解析】

在处理流程设计过程中，为了更清晰地表达过程规则说明，陆续出现了一些用于表示处理流程的工具，这些工具包括三类：图形工具、表格工具和语言工具。其中常见的图形工具包括程序流程图、IPO图、盒图、问题分析图、判定树，表格工具包括判定表，语言工具包括过程设计语言等。

程序流程图(Program HowDiagram，PFD)用一些图框表示各种操作，它独立于任何一种程序设计语言，比较直观、清晰，易于学习掌握。流程图中只能包括5种基本控制结构：顺序型、选择型、WHILE循环型(当型循环)、UNTIL循环型(直到型循环)和多分支选择型。

IPO图是由IBM公司发起并逐步完善的一种流程描述工具，其主体是处理过程说明，可以采用流程图、判定树、判定表、盒图、问题分析图或过程设计语言来进行描述。IPO图中的输入、输出与功能模块、文件及系统外部项都需要通过数据字典来描述，同时需要为其中的某些元素添加注释。

N-S图与PFD类似，也包括5种控制结构，分别是顺序型、选择型、WHILE循环型(当型循环)、UNTIL循环型(直到型循环)和多分支选择型，任何一个N-S图都是这5种基本控制结构相互组合与嵌套的结果。在N-S图中，过程的作用域明确；它没有箭头，不能随意转移控制；而且容易表示嵌套关系和层次关系；并具有强烈的结构化特征。但是当问题很复杂时，N-S图可能很大。

问题分析图(Problem AnalysisDiagram，PAD)是继PFD和N-S图之后，又一种描述详细设计的工具。PAD也包含5种基本控制结构，并允许递归使用。

过程设计语言(Process DesignLanguage，PDL)也称为结构化语言或伪代码(pseudocode)，它是一种混合语言，采用自然语言的词汇和结构化程序设计语言的语法，用于描述处理过程怎么做，类似于编程语言。过程设计语言用于描述模块中算法和加工逻辑的具体细节，以便在开发人员之间比较精确地进行交流。

对于具有多个互相联系的条件和可能产生多种结果的问题，用结构化语言描述则显得不够直观和紧凑，这时可以用以清楚、简明为特征的判定表(DecisionTable)来描述。判定表采用表格形式来表达逻辑判断问题，表格分成4个部分，左上部分为条件说明，左下部分为行动说明，右上部分为各种条件的组合说明，右下部分为各条件组合下相应的行动。

判定树(DecisionTree)也是用来表示逻辑判断问题的一种常用的图形工具，它用树来表达不同条件下的不同处理流程，比语言、表格的方式更为直观。判定树的左侧(称为树根)为加工名，中间是各种条件，所有的行动都列于最右侧。