

第1章 绪论

1、解释术语

(1) 软件：软件是指计算机系统上的程序及其文档。P16

(2) 软件工程：软件工程是应用计算机科学理论和技术以及工程管理原则和方法，按预算和进度实现满足用户要求的软件产品的工程，或以此为研究对象的学科。P15

(3) 软件危机：软件生产率、软件质量远远满足不了社会发展的需求，成为社会，经济发展的制约因素，人们通常把这一现象称为“软件危机”。P15

2、简答题

(1) 简述软件开发的本质。

答：软件开发的本质就是实现问题空间的概念和处理逻辑到解空间的概念和处理逻辑之间的映射。P19

(2) 简述实施软件开发的基本途径。

答：实施软件开发的基本途径是系统建模。所谓系统建模，是指运用所掌握的知识，通过抽象，给出该系统的一个结构——系统模型。P19

(3) 简述何谓模型以及软件开发中所涉及的模型。

答：模型是一个抽象。该抽象是在意图所确定的角度和抽象层次对物理系统的一个描述，描述其中的成分和成分之间所具有的特
定语义的关系，还包括对该系统边界的描述。

软件开发中所涉及的模型可分为两大类，一类称为概念模型，描述了系统是什么；另一类统称为软件模型，描述了实现概念模型
的软件解决方案。P19

(4). 简述软件开发所涉及的两大类技术。

答：软件开发所涉及的两大类技术为：一是求解软件的开发逻辑，二是求解软件的开发手段。P17

第 2 章 软件需求与软件需求规约

1、解释术语

(1) 软件需求

软件需求以一种技术形式，描述了一个产品/系统应该具有的功能、性能和其它性质。P23

(2) 功能需求

功能需求规约了系统或系统构件必须执行的功能。P24

(3) 非功能需求

非功能需求是性能、外部接口、设计约束和质量属性这 4 类需求的统称。P23

(4) 需求规约

需求规约是一个软件项/产品/系统所有需求陈述的正式文档，它表达了一个软件产品/系统的概念模型。P28

2、简述需求与需求规约的基本性质。

答：需求的基本性质：

- 1) 必要的，该需求是用户所要求的。
- 2) 无歧义的，该需求只能用一种方式解释。
- 3) 可测的，该需求是可进行测试的。
- 4) 可跟踪的，该需求可从一个开发阶段跟踪到另一个阶段。
- 5) 可测量的，该需求是可测量的。P23

需求规约的基本性质：

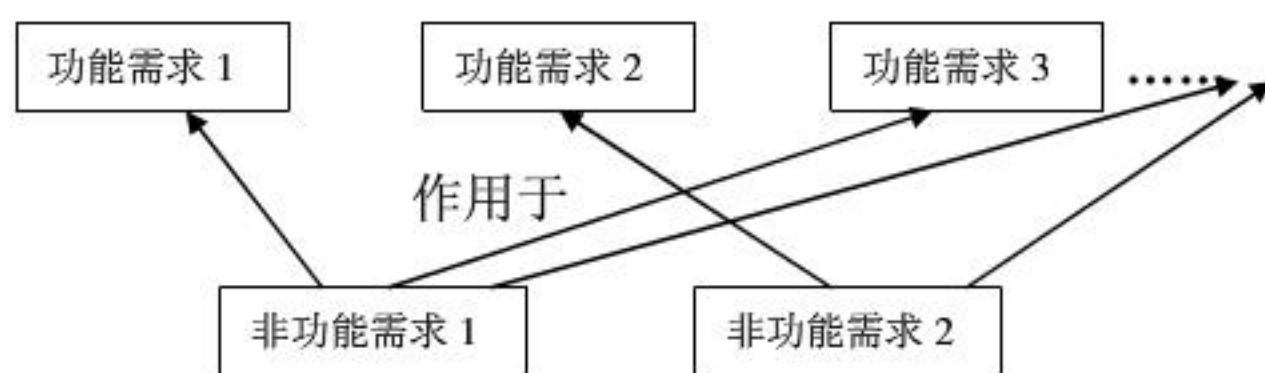
- 1) 重要性和稳定性程度：按需求的重要性和稳定性，对需求进行分级。
- 2) 可修改的：在不过多地影响其他需求的前提下，可以容易地修改一个单一需求。
- 3) 完整的：没有被遗漏的需求。
- 4) 一致的：不存在互斥的需求。P28

3、简述软件需求的分类。

答：软件需求可以分为两大类：一类是功能需求，一类是非功能需求，而非功能需求可分为性能需求，外部接口需求、设计约束和质量属性需求。P23

4、举例说明功能需求和非功能需求之间的基本关系。

答：非功能需求可作用于一个或多个功能需求，例如



其中，非功能需求 1 作用于功能需求 1 和功能需求 3 等；非功能需求 2 作用于功能需求 2 等。P24

5、有哪几种常用的初始需求发现技术？

答：有 5 种常用的需求发现技术：自悟、交谈、观察、小组会和提炼。P26

6、简述需求规约的 3 种基本形式。

(1) 非形式化的需求规约。非形式化的需求规约即以一种自然语言来表达需求规约，如同使用一种自然语言写了一篇文章。

(2) 半形式化的需求规约。半形式化的需求规约即以半形式化符号体系（包括术语表、标准化的表达格式等）来表达需求规约。

(3) 形式化的需求规约。形式化的需求规约即以一种基于良构数学概念的符号体系来编制需求规约，一般往往伴有解释性注释的支持。 P29

7、简述软件需求规约的内容和作用。

答：软件需求规约的内容有：引言、总体描述、特定需求、附录、索引。P28

需求规约的作用可概括为以下 4 点：

1) 需求规约是软件开发组织和用户之间一份事实上的技术合同书，是产品功能及其环境的体现。

- 2) 对于项目的其余大多数工作，需求规约是一个管理控制点。
 - 3) 对于产品/系统的设计，需求规约是一个正式的、受控的起始点。
 - 4) 需求规约是创建产品验收测试计划和用户指南的基础。P31
- 8、简述需求规约在项目开发中的基本作用。

答：需求规约的作用可概括为以下 4 点：

- 1) 需求规约是软件开发组织和用户之间一份事实上的技术合同书，是产品功能及其环境的体现。
 - 2) 对于项目的其余大多数工作，需求规约是一个管理控制点。
 - 3) 对于产品/系统的设计，需求规约是一个正式的、受控的起始点。
 - 4) 需求规约是创建产品验收测试计划和用户指南的基础。P31
- 9、简述需求规约和项目需求的不同。

答：需求规约和项目需求是两个不同的概念。需求规约是软件开发组织和用户之间一份事实上的技术合同书，即关注产品需求，回答“交付给客户的产品/系统是什么”；而项目需求是客户和开发者之间有关技术合同——产品/系统需求的理解，应记录在工作陈述中或其他某一项目文档中，即关注项目工作与管理，回答“开发组要做的是做什么”。P30

第三章 结构化方法

1. 基本概念

需求分析：一般来说，分析是系统地使用信息，对一个问题估算。软件需求分析是这一概念的特化，即系统化地使用“数据流”、“加工”、“数据存储”、“数据源”和“数据潭”等术语所表达的信息，对待建系统“是什么”给出一个估算——系统概念模型

软件设计：在需求分析的基础上，定义满足需求所需要的结构，即针对给定的问题，给出该问题的软件解决方案，确定“怎么做”的问题

数据流图：表达功能模型的工具，即数据流图（Dataflow Diagram） 简称 DFD 图，简单的说，DFD 图是一种描述数据变换的图形化工具，其中饮食的元素可以是数据流、数据存储、加工、数据源和数据潭等

变换型数据流图：具有较明显的输入部分和变换（主加工）部分之间的界面变换部分和输出部分之间界面的数据流图

事务型数据流图：数据到达一个加工 T，该加工 T 根据输入数据的值，在其后的基干动作序号（称为一个事务）中选出一个来执行

模块：执行一个特殊任务的一个过程以及相关的数据结构

1. 简答题

2. 何谓模块耦合？简述模块耦合的类型。

答：耦合是不同模块之间相互依赖程序的度量

内容耦合：当一个模块直接修改或操作另一个模块的数据，或一个模块不通过正常入口而转入到另一个模块时，

公共耦合：两个或两个以上的模块共同引用一个全局数据项

控制耦合一个模块通过气喘吁吁向另一个模块传递一个控制信息，接收信号的模块根据信号值进行适当的动作

标记耦合：若一个模块 A 通过接口向两个模块 B 和 C 传递一个公共参数

数据耦合：模块之间通过参数来传递数据

3. 何谓模块内聚？简述模块内聚的类型。

答：指一个模块内部各成分之间相互关联程度的度量

偶然内聚：一个模块的各成分之间基本不存在任何关系

逻辑内聚：几个逻辑上相关的功能被放在同一个模块中

时间内聚：一个模块完成的功能必须在同一时间内执行，但这些功能只是因为时间因素关联在一起

过程内聚：一个模块内部的处理成分是相关的，而且这些处理必须以特定的次序执行

通信内聚：一个模块的所有成分都操作同一数据集或生成同一数据集

顺序内聚：一个模块的各个成分和同一个功能密切相关，而且一个成分的输出作为另一个成分的输入

功能内聚：最理想的内聚，模块的所有成分对于完成单一的功能都是基本的。功能内聚的模块对完成其功能而言是充分必要的

4. 何谓模块的控制域和模块的作用域？并举例说明

控制域：模块本身以及所有直接或间接从属于它的模块的集合。

作用域：受该模块内的一个判定所影响的所有模块的影响

第四章 面向对象方法—UML

1. 基本概念

类及其属性和操作

类是一组具有相同属性、操作、关系和语义的对象的描述。类的属性是类的一个命名特征，该特征是由该类的所有对象所共享、用于表达对象状态的数据

接口是操作的一个集合，其中每个操作描述了类、构件或子系统的一个服务

关联及其链：关联是类之间的一种结构关系，是对一组具有相同结构、相同链的描述。链是对象之间具有特定语义关系的抽象

泛化：泛化是一般性类目（称为超类或父类）和它的较为特殊性类目（称为子类）之间的一种关系，有时称为” is a kind of ” 关系

聚合：聚合是关联的一种特殊形式，表达的是一种“整体/部分”关系

依赖：依赖是一种使用关系，用于描述一个类目使用另一个类目的信息和服务

2. 简答题

为了表达客观事物，UML 给出了哪些基本术语？

答：为了支持抽象分析和设计中的事物，UML 给出了 8 个基本术语，即类、接口、协作、用例、主动类、构件、制品、结点，并

给出了这些基本术语的一些变体。

为了表达客观事物之间的关系，UML 给出了哪些基本术语？这些术语之间是什么关系？

关联、泛化、细化和依赖，以及它们的一些变体

什么是对象的构成与表示？并说明。

类是一组具有相同属性、操作、关系和语义的对象的描述。对象是类的一个实例

什么是类图的构成成分？

答：类图是可视化地表达系统表态结构模型的工具，通常包含类、接口、关联、泛化和依赖

什么是顺序图的构成成分？

顺序图是一种交互图，即由一组对象以及按时序组织的对象之间的关系组成，其中还包含这些对象之间所发送的消息

如何描述对象之间的关联语义

用况之间有哪几种关系？

在什么情况下需要建立状态图？

状态图可用于创建有关系统（或系统成分）的行为生存周期模型，表达有关系统（或系统成分）的一种动态结构，给出有关系统（或系统成分）在生存期间有哪些阶段、每一阶段可从事的活动以及对外所呈现的特征等方面 的信息

对象操作和对象状态之间的关系是什么？

同时引入“操作”和“方法”的目的是什么？

答：表达模型化包之间的关系

为什么使用包？如何划分包

使用 UML 可以从那些角度来刻画一个系统的行为？为什么？

何谓顺序图中的控制操作子？试举例说明。

为了控制交互行为描述的复杂性，以便更清晰地表达顺序图中的复杂控制，给出了个个控制操作子，选择、条件、并发、迭代操

第五章 面向对象方法—RUP

1. 基本概念

RUP 的定义及主要特点

RUP 是一种软件开发过程框架，基于面向对象符号体系给出了有关软件开发过程组织及实施的指导。该框架体现了 3 个突出特征，即以用况驱动、体系结构为中心以及迭代、增量式开发

演化模型与“RUP 增量、迭代开发”之间关系

RUP 迭代、增量式开发是演化模型的一个变体，即规定了“大的”迭代数量—4 个阶段，并规定了每次迭代的目标

初使阶段：获得与特定腹部和平台无关的系统体系结构轮廓，以此建立产品功能范围；编制实例业务实例，从业务角度指出该项目的价值，减少项目主要的错误风险

精华阶段：通过捕获并描述系统的大部分需求，建立系统体系结构基线的第一个版本，主要包括用况模型和分析模型，减少次要的错误风险，到该阶段末，就能够估算成本、进步，并能详细地规划构造阶段

构造阶段：通过演化，形成最终的系统体系结构基线，开发完整的系统，确保产品可以开始向客户交付，即具有初始操作能力

移交阶段：确保有一个实在的产品发布给用户群。期间培训用户如何使用该软件

RUP 与 UML 之间关系

RUP 与 UML 是一对“姐妹”，它们构成了一种特定的软件开发方法学。其中，UML 作为一种可视化建模语言，给出了表达事物和事物之间关系的基本术语，给出了多种模型的表达工具；而 RUP 利用这些术语定义了需求获取层、系统分析层、设计层、实现层，并给出了实现各层模型之间映射的基本活动以及相关指导

什么是特征（Teature）？举例如何描述它。

从客户、用户、计划者、开发者想法和意愿中搜取特征，形成特征表。特征是一个新的项及其简要描述

“按不同科目计算平均成绩” 计算平均成绩：按所学的不同科目计算每一个学生的期末考试平均成绩，给出分数段并描述其状态（如提议、批准、合并和验证等）、实施的代价及风险、重要程度以及对其他特征的影响等

特征可作为需求，并被转换为其它制品

需求获取层及相关概念

需求获取层目标：使用 UML 中的用况、参与者以及依赖等术语来抽象客观实际问题，形成系统的需求获取模型；基本术语：用况、参与者、用于表达用况参与者之间关系的关联、用于表达用况之间的包含和扩展、用于表达参与者之间关系泛化。术语确定了系统用况模型的各种形态

需求获取模型的基本组成

使用 UML 中 的用况、参与者以及依赖等 术语来抽象客观实际问题，形成系统的需求获取模型

建造一个系统需求获取模型的活动和任务，以及各活动的输入和输出

1. 发现描述参与者和用况，输入：业务模型或领域模型，补充需求，特征表;输出：用况模型[概述]，术语表
2. 赋予用况优先级：输入：用况模型[概述]，补充需求，术语表;输出：体系结构描述[用况模型视角]
3. 精华用况：输入：用况模型[概述]，补充需求，术语表;输出：用况[精化]
4. 构造人机接口原型：输入：用况[精华]，用况模型[概述]，补充需求，术语表;输出：人机接口原理
5. 用况模型结构化：输入：用况[精华]，用况模型[概述]，补充需求，术语表;输出：用况模型[精化]

如何描述系统的参与者和用况？举例说明

参与者：发现参与者与描述参与者：1)之前已经存在业务用况模型，可依据业务模型直接发现一些候选参与者，2)没有业务用况模型，即使存在领域模型，也需要系统分析人员与客户一起来标识系统参与者

用况是系统向它的参与者提供结果（值）的功能块，表达参与者使用系统的方式，因此一个用况可用于规约系统可执行的、与参与者进行交互的一个动作序列，包括其中一些可选动作序列，并且用况还有自己 的属性

需求获取层对以后开发工作的影响？

需求分析层及相关概念

在系统用况模型的基础上，创建系统分析模型以及在该分析模型视角下的体系结构描述，系统分析模型是系统的一种概念模型，解决系统用况模型中存在的二义性和不一致性问题，并以一种系统化的形式准确地表达用户的需求

需求分析模型的基本组成

RUP 的分析如同结构化分析，其目标之一是在一个特定的抽象层上建立系统分析模型。为此，RUP 首先给出了 3 个术语：分析包、分析类和用况细化，用于表达需求中“大粒度”的概念，开发人员使用这些术语可以规约系统分析中所要使用的信息

分析类：是类的一种衍型，很少有操作和特征标记，而用责任来定义其行为，并且其属性和关系也是概念性的，包括：边界类、实体类、控制类

用况细化：是一个针对一个用况，其行为可用多个分析类之间的相互作用来细化，并记为用况细化[分析]

分析包：分析包是一种控制信息组织复杂性的机制，提供了分析制品的一种组织手段，形成了一些可管理的部分。

建造一个系统需求分析模型的活动和任务，以及各活动的输入和输出

体系结构分析：输入：用况模型、补充需求、业务模型或领域模型、体系结构描述[用况模型];输出：分析包[概述]、分析类[概述]、体系结构描述[分析]

细化用况：输入：用况模型、补充需求、业务模型或领域模型、体系结构描述[分析];输出：用况细化[分析]、分析类[概述]

对类分析：输入：用况细化[分析]、分析类[概述] 输出：分析类[完成]

对包进行分析：输入：系统体系结构描述[分析]、分析包[概述]输出：分析类[完成]

需求分析模型对以后开发工作的影响

- 1) 对设计中子系统的影响。分析包一般将影响设计子系统的结构
- 2) 对设计类的影响。分析包可以作为类设计时的规格说明。
- 3) 对用况细化[设计]的影响。用况细分[分析]对用况细化[设计]有两方面影响，一个是它们有乃至为用况创建更精确的规格说明，另一个是当对用况进行设计时，用况细化[分析]可作为其输入。

需求获取模型与需求分析模型之间比较

- 1) 语言描述不同：客户语言与开发语言
- 2) 视图：系统外与系统内
- 3) 结构：使用用况予以结构化，给出外部视角系统结构与使用衍型类结构化，给了部视角系统结构

- 4) 作用：标注“系统应该做什么，不应该做什么”与可以做出开发者理解系统如何勾画、如何设计和如何实现基础
- 5) 问题：可能存在冗余、不一致和冲突等问题与解决了上述问题
- 6) 捕获系统功能，包括体系结构方面具有意义的功能与给出细化系统功能，包括在体系结构方面具有意义的功能
- 7) 定义一些进一步需要在分析模型中予以分析用况与定义每一个用况细化

设计层及相关概念

设计目标：定义满足系统/产品分析模型所规约需求的软件结构

基本术语：设计子系统、设计类、用况细化[设计]、接口、以及用于表达子系统之间关系的依赖、用于表达设计类之间关系的关联等，这些术语确定了系统设计模型的各种形态

设计模型的基本组成

设计子系统、设计类、用况细化[设计]、接口、以及用于表达子系统之间关系的依赖、用于表达设计类之间关系的关联等，这些术语确定了系统设计模型的各种形态

建造一个系统设计模型的活动和任务，以及各活动的输入与输出

体系结构设计：输入：用况模型、补充需求、分析模型、体系结构描述[分析模型角度];输出：

子系统[概述]、接口[概述]、设计类[概述]、部署模型[概述]、体系结构描述[设计]

设计用况：输入：用况模型、补充需求、分析模型、部署模型；

输出： 用况[设计-实现]、设计类[概述]、子系统[概述]、接口[概述]

对类设计：输入：用况[设计-实现]、设计类[概述]、接口[概述] 、分析类[完成];输出：设计类[完成]

设计子系统：体系结构描述[设计]、子系统[概述]、接口[概述];输出：子系统[完成]、接口[完成]

如何处理需求中所捕获的非功能需求。

需求分析模型与设计模型之间的比较

第六章 软件生存周期过程与管理

1. 基本概念

软件测试：有规程地发现错误的过程，其中错误（ERROR）：与所期望的设计之间的偏差，该偏差可能产生不期望的系统行为或失效。失效（FAILURE）：与所规约的系统执行之间的偏差。失效是系统故障或错误的后果。故障（FAULT）：导致错误或失效的不正常条件。故障可以是偶然性的或是系统性的。

测试用例：为了发现程序中的故障而专门设计的一组数据或脚本

测试覆盖率：定量描述一个或一组测试的效率

2. 简答题

测试过程模型，并分析这一模型在软件测试技术研究以及实践中的作用

答：软件测试是一个有程序的过程，包括测试设计、测试执行、以及测试结果比较等

1. 环境模型、被测模型和错误模型在软件测试中扮演了一种很重要的角色；这些模型的质量，特别是程序模型的质量，对发现错误具有关键性的作用
2. 软件测试的错误假定是，实际结果与预期结果不符，而后在此基础上可进一步分析是什么错误

软件测试与调度的区别

答：

- 1) . 测试从一侧面证明程序员的“失败”。调度为了证明程序员是正确的
- 2). 测试以已知条件开始，使用预告定义的程序且有预知的结果，不可预见的仅是程序是否通过测试
- 3). 测试是有计划的，并要进行测试设计。高度是不受时间约束的
- 4). 测试是发现错误、改正错误、重新测试的过程。调试是一个推理过程
- 5). 的执行是有规程的。调 试的执行往往要求程序员进行必要推理
- 6). 测试经常由独立的测试组在不了解软件技术的条件下完成的。高度必须由了解详细设计的程序员完成
- 7) 大多数测试的执行和设计可由工具支持。调试时，程序员能利用的工具主要是调试器

程序流程图的作用及构成

答：着重于过程属性描述

简述语句覆盖、分支覆盖、条件组合覆盖、路径覆盖的含义及它们之间的关系

答：路径覆盖：执行所有可能穿过程序控制流程的路径

语句覆盖：至少执行程序中的所有语句一次

分支覆盖：至少将程序中的每一个分支执行一次

条件覆盖：指每个判定中的所有可能的条件取值至少执行一次

条件组合覆盖：设计足够测试，使每个判定中的所有可能的条件聚会组合至少执行一次

存在以下关系：

语句覆盖《分支覆盖《条件组合覆盖《。。。路径覆盖

简述单元测试、集成测试、有效性测试的含义及它们之间的区别

答：单元测试：主检验软件设计的最小单元——模块。该测试以详细设计文档为指导，测试模块内的重要控制路径。

集成测试：是软件组装的一个系统化技术，其目标是发现与接口有关的错误

有效性测试：发现软件实现的功能与需求规格说明书不一致的错误

简述路径测试技术、事务流测技术的主要依据

答：一个依据程序逻辑结构，一个依据软件行为描述

简述程序流程图与事务流程图之间的主要区别，并分析这些区别的主要原因

答：1)基本模型元素所表达的语义不同

2)一个事务不等于路径测试中一条路径，可能在中间某处就完成了某一用户工作，终结了一个事务

3).事务流程图中的分支和节点可能是一个复杂的过程

简述白盒测试技术的要点，并举例说明

答：白盒测试技术依据程序的逻辑结构，以控制流程图作为被测对象建模工具，其中涉及过程块、分支、节点、链以及路径，并针对测试民，给出了 4 种覆盖策略：语句覆盖、分支覆盖、条件组合覆盖和路径覆盖，它们之间具有偏序关系，并且可根据项目需求给出其他覆盖策略

事务流测试技术的要点，并举例说明

答：事务流测试技术是一种功能测试技术，目前提出了很多功能测试技术，如定义域测试技术、等价类测试技术以及基于因果图的测试技术等，统称为黑盒子测试技术。黑盒测试将被测软件看成黑盒子，只通过外部的输入和输出来发现软件中的错误，因此黑盒测试是一种基于软件规约的测试

第七章 软件生存周期过程与管理

1. 基本概念

软件生存周期过程：软件生存周期是软件产品或系统的一系列相关活动的全周期。从形成概念开始，历经开发、交付使用、在使

用中不断修订和演化，直到最后被淘汰，让位于新的软件产品

软件生存周期模型：是一个包括软件产品开发、运行和维护中有关过程、活动和任务的框架，覆盖了从该系统的需求定义到系统的使用终止

过程管理：过程规则与管理是软件项目管理的一项重要工作。没有过程规划就谈不上过程管理乃至项目管理，没有过程管理就不可能存在有效的软件工程

2. 简述软件开发中的过程类，以及它们的基本作用和它们之间的基本关系

答：分为 3 类

基本过程：那些软件生产直接相关的活动集，分 5 个过程：获取过程、供应过程、开发过程、运行过程和维护过程

支持过程：有关各方面按他们的目标所从事的一系列相关支持活动集，有助于提高系统或软件产品的质量，分为：文档过程、配置管理过程、质量保证过程、验证过程、确认过程、联合评审过程、审计过程和问题解决过程等

组织过程：与软件生产组织有关的活动集，分管理过程、基础设施过程、培训过程、改进过程

3. 在 ISO/IEC12207-2008 中如何描述一个过程？举例说明

答：1). 为获取方获取一个软件产品或服务，为供应方开发、运行、维护、提供和销毁一个软件产品，建立了一种软件生存周期框架，包含过程、活动和任务，并通过过程分类、过程描述，给出它们之间的内在关系
2). 为软件生存周期过程的定义、控制和改进提供了一个过程，即生存周期模型管理过程

供应过程：为获取方提供满足所协商需求的产品或服务

软件实现过程：软件实现过程是为了生产一个已规约系统元素，作为一个软件产品或服务而实现

软件需求分析过程：建立系统软件部分的需求

软件体系结构设计：为软件的实现及其可以按需求进行验证，提供一种设计

软件验证过程：证实一个过程或项目的每一个软件工作产品/服务是否正确地反映所规约的需求

软件确认过程：证实所期望的软件工作产品是否满足其需求？？？

4. 什么是软件开发特定的过程类？举例说明

5. 什么是验证和确认？简述它们的作用和区别

答：验证：证实一个过程或项目的每一个软件工作产品/服务是否正确地反映所规约的需求

验证和确认是有区别的。

验证是通过提供的客观证据，证实规约的需求是否得以满足

确认是通过提供的客观证据，证实有关特定期望的使用或应用的需求是否得以满足

软件确认过程：证实所期望的软件工作产品是否满足其需求

6. 简述瀑布模型以及可适应的情况

瀑布模型将软件生存周期的各项活动规定为按固定顺序而连接的若干阶段工作，形如瀑布流水，最终得到产品

适应情况：需求已被很好的理解，并且开发组织非常熟悉为实现这一模型所需求的过程

7. 简述演化模型以及可适应的情况

答：表达了一种弹性的过程模式，由一些小的开发步组成，每一步历经需求分析、设计、实现和验证，产生软件产品的一个增量，通过这些迭代，最终完成软件产品的开发

主要针对事先不能完整定义需求的软件开发

8. 简述增量模型的优缺点

答：优点 1) 第一个可交付版本所需要的成本和时间是较少的，从而可减少开发由增量表示的小系统承担的风险
2) 由于很快发布第一个版本，因此可以减少用户需求的变更
3) 允许增量投资，即在项目开始时可以仅对一个或两个增量投资

缺点：1) 如果没有对用户的变更要求进行规划，那么产生的初始增量可能会造成后来增量的不稳定
2) 如果需求不像早期思考的那样稳定和完整，那么一些增量就可能需要重新开发，重新发布
3) 由于进度和配置的复杂性，可能会增大管理成本，超出组织的能力

9 简述螺旋模型以及它与其它模型之间的主要区别

答：螺旋模型是瀑布模型与深化模型的基础上，加入两者所忽略的风险分析所建立的一种软件开发模型

螺旋模型关注解决问题的基本步骤，即标识问题，标识一些可选方案，选择一个最佳方案，遵循动作步骤并实施后续工作，突出特征，在开发的迭代中实际上只有一个迭代过程真正开发了可交付的软件

2) 与深化模型和增量模型相比，同样使用了瀑布模型作为一个嵌入的过程，即分析、设计、编码、实现和维护的过程，并且在框架和全局体系结构方面是等同的。但是，螺旋模型所关注的阶段以及它们的活动是不同的，如增加一些管理活动和支持活动。尽

管增量模型也有一些管理活动，但它基于以下假定：需求是最基本的、并且是唯一的风险源，因而在螺旋模型中增大了决策和风险的空间，螺旋模型扩大了增量模型的管理范围。

如果项目的开发风险很大或客户不能确定系统需求，在更广泛的意义上来讲，还包括一个系统或系统类型的要求，这时螺旋模型就是一个好的生存周期模型

第八章 集成化能力成熟度模型

一、术语解释

过程域：是一个业务域中一束相关实践，当它们一起得以实现时，就满足被认为对该过程域的改善具有重要作用的一组条件。

过程改善：是指人为设计的一个活动程序，其目的是改进组织的过程性能和成熟度，并改进这一程序的结果

专用目标：每一个过程域中都有一个或多个“专用目标”，用于描述该过程域必须呈现的一些独有特征

共用目标：每一个过程域中都有一个或多个“共用目标”，用于描述实现制度化的该过程必须呈现的特征

专用实践：每一个过程域中都有一个或多个“专用实践”，这些专用实践被认为对于达到该过程域的专用目标是重要活动，即期望以专用实践所描述的活动，会导致达到一个过程域的专用目标

共用实践：每一个过程域中都有一个或多个“共用实践”，这些共用实践被认为对于达到该过程域相关的共用目标是重要活动

能力等级：是指单一过程域中已达到的过程改善，能力等级是为了管理，对过程改善程序所设定的几个“台阶”

成熟度等级：是指达到预先定义的一组过程域所有目标的一种过程改善等级

2. 简答题

CMMI 提出所基于的基本思想

答：该模型基于过程途径思想，通过过程把软件质量的 3 个支撑点——受训的人员、规程和方法、工具和设备进行集成，以开发所期望的系统/产品。为此，CMMI 紧紧围绕开发、维护和运行，把经过证明的“最佳实践“放在一个结构中。该结构有乃至指导组织确定其过程的发送优先次序;有乃至指导这些改善的实施，以提高其过程能力和成熟度，并且还支持其它领域（如获取和服务）能力成熟度模型开发。

什么是过程制度化？在 CMMI 把过程制度化分几个等级？简要回答每一等级主要特征

简述 CMMI 模型支持两种过程改善路径

答：CMMI 提供了两种改善路径，一是称为能力等级是一种过程改善路径，该路径可使组织针对单一过程域不断改善该过程域
二是称为成熟度等级也是一种过程改善路径，该路径可使组织通过关注一组过程域不断改善一组相关过程域

简述 CMMI 模型的模型部件及部件间关系

组图

简述专用实践与共用实践关系

答：专用实践：每一个过程域中都有一个或多个“专用实践”，这些专用实践被认为对于达到该过程域的专用目标是重要活动，即期望以专用实践所描述的活动，会导致达到一个过程域的专用目标

共用实践：每一个过程域中都有一个或多个“共用实践”，这些共用实践被认为对于达到该过程域相关的共用目标是重要活动，例如，对共有目标“该过程予以制度化，使之成为一个已管理过程“而言，一个共用实践是”为该过程的执行、工作产品的开发以及该过程的服务，提供充足的资源“。

之所以称为“共用实践“，是因为同一实践可应用于多个过程。

简述每一成熟度等级所包含的过程域。

答：在成熟度等级，把开发、维护、运行中的过程分为 4 个组。

- 1) 包含 7 个过程域：配置管理、测量与分析、项目监控、项目规划、过程和产品质量保证、需求管理、提供方协议管理。
- 2) 包含 11 个过程域：决策分析与解决、集成项目管理、组织过程定义、缓缓过程关注、组织培训、产品集成、需求开发、风险管理、技术解决方案、验证、确定
- 3) 组织过程性能和定量项目管理
- 4) 原因分析与解决和组织创新和部署

3. 简述项目规划过程域的专用目标与专用实践

答：专用目标 1:SG1 建立估算，4 个专用实践

SP1.1 估算项目规模, SP1.2 建立工作产品和任务属性的估算 SP1.3 定义项目生存周期 SP1.4 确定工作量和成本的估算

专用目标 2:SG2 开发项目计划, 7 个专用实践:

SP2.1 建立预算和进度, SP2.2 标识项目风险 SP2.3 规划数据管理 SP2.4 规划项目资源 SP2.5 规划需要的知识和技能 SP2.6 规划利益攸关方参与 SP2.7 建立项目计划

专用目标 3:SG3 获得对该计划的承诺:

SP3.1 评审该项目的计划 SP3.2 调和工作和资源等级, 使之一致。SP3.3 获得计划承诺

4. 简述开发过程的专用目标与专用实践

答：专用目标 1:SG1 开发客户需求，2 个专用实践

SP1.1 引出要求 SP1.2 开发客户需求

专用目标 2:SG2 开发产品需求，3 个专用实践

SP2.1 建立产品和产品构件的需求 SP2.2 分配产品构件需求 SP2.3 标识接口需求

专用目标 3:SG3 分析并验证需求, 5 个专用实践

SP3.1 建立操作概念和场景 SP3.2 建立所需功能的定义 SP3.3 分析需求 SP3.4 分析需求，达到权衡 SP3.5 确认需求

5. 简述共用目标 2 及其相关的共用实践

共用目标 2:GG2 把过程制度化为一个已管理过程，10 个共用实践

GP2.1 建立组织策略 GP2.2 规划该过程 GP2.3 提供资源 GP2.4 指定责任 GP2.5 培训人员 GP2.6 管理配置 GP2.7 标识相关利益方的参与 GP2.8 监控该过程 GP2.9 客观地评估过程的符合性 GP2.10 以高层管理的视觉评审状态

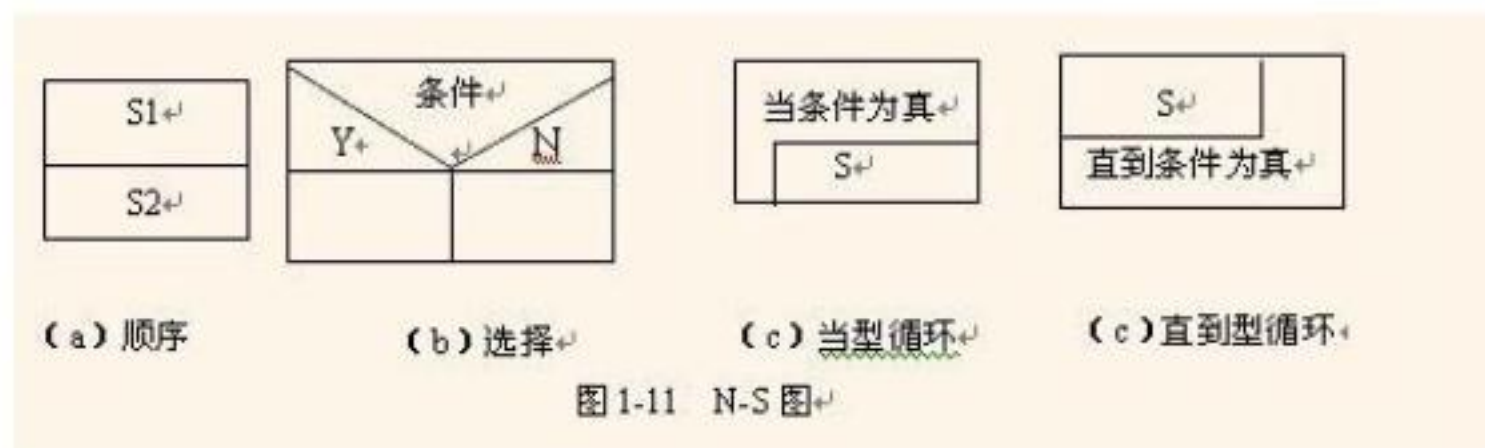
6. 简述各共用目标及其相关的共用实践

简述共用目标 2 及其相关的共用实践

答

37. 这是用 PDL 书写的一段程序。请根据你的理解，用 N-S 图表示它。P65 页

```
begin
    A;
    While(P1){
        if(P2){
            B;
        }else{
            C;
        }D;
    }
End
```



需求发现的常用技术中，无法验证所发现的需求是否满足用户的要求，而且无法验证其正确性的是__自悟__和__提炼__。

31. 简述软件需求的分类及其关系。P23-24;

答：软件需求可以分为功能需求和非功能需求 2 大类；功能需求规定了系统及构件必须执行的功能；非功能需求又可以分为性能需求、外部接口需求、设计约束和质量属性需求。功能需求是整个软件需求的主体，没有功能需求就没有性能、外部接口、设计约束和质量的需求；一个非功能需求可以作用于 1 个或多个功能需求。

32. 什么是模块?什么是模块内聚?请列出从低到高的常见内聚类型。P56、57、58、59

答：模块是执行一个特殊任务的过程以及相关的数据结构。内聚是指一个模块内部各个成分之间相互关联程度的度量。从低到高的内聚类型：偶然内聚；逻辑内聚；时间内聚；过程内聚；通信内聚；顺序内聚；功能内聚。

33. 什么是状态?什么是状态图?简述实际应用中状态图的作用。107、108、113

答：状态是类目的一个实例在其生存中的一种条件或情况；期间该实例满足这一条件，就执行某一活动或等待一个消息。状态图是现实状态机的图，强调从一个状态到另一个状态的控制流。在实际使用中状态图的作用：创建一个系统的动态图和创建一个场景的模型。

34. 简述 RUP 中需求获取的基本步骤和相关制品。P132

答：需求获取的步骤和相关制品：

第 1 步是列出候选的特征，相关制品是特征表；

第 2 步是理解系统语境，相关制品是领域模型或业务模型；

第 3 步是捕获系统功能需求，相关制品是用况模型（use case 模型）；

第 4 步是捕获非功能需求，相关制品是补充的需求或针对特殊需求的用况。

35. 简述黑盒测试技术的要点。P186

答：黑盒测试技术的要点：

(1) 支持测试工程模型的中间部分；

- （2）事务流测试技术是将路径测试技术用于功能测试的产物，是一种实用的功能测试技术，通过事务的操作逻辑发现软件中的错误；
- （3）事务流测试技术是基于软件规约的，对错误的假定是软件通过了与预想不同的事务路径；
- （4）基于事务的基本操作；事务流测试技术的最大问题和最大代价是获取事务流程图及用例设计；
- （5）事务处理流程测试要达到基本的测试覆盖。

36. 简述增量模型以及可适应的情况。P224

答：增量模型意指需求可以结构化分组，形成一个个增量，并形成一结构，之后对每一个增量进行瀑布开发。

用增量模型开发的前提是需求的结构化，模型适合“技术驱动”的软件产品开发。

31.软件开发方法：一种使用早已定义好的技术集及符号表示习惯来组织软件生产的过程.

32.数据字典：简称D D，就是用来定义数据流图中的各个成分具体含义的，它以一种准确的、无二义性的说明方式为系统的分析、设计及维护提供了有关元素的一致定义和详细的描述.

33.黑盒测试法：将程序当成是黑盒子以设计中的功能为标准在程序的接口出进行测试看它能否达到功能要求。

34.软件质量保证：向用户及社会提供满意的高质量的产品，确保软件产品从诞生到消亡为止的所有阶段的质量活动，即确定、达到和维护需要的软件质量而进行的所有有计划、有系统的管理活动。

35.模块的作用范围：一个模块的控制范围是指它本身及其所有下属模块的集合。这里下属模块包括直接下属模块及下属模块的下属模块

36.简述软件产品的特性。

答：1.软件产品是一种逻辑产品。

2.软件产品的生产主是研制，软件产品的成本主要体现在软件的开发和研制上，软件开发完成后，复制就产生了大量软件产品。

3 .软件产品不会用坏，不存在磨损、消耗问题。

4.软件产品的生产主要是脑力劳动，还未完全摆脱手工开发方式，而且碰分是"定做"的。

5.软件费用不断增加，成本相当昂贵。

37.简述在需求分析基本任务中，对问题的综合需求包括哪几个方面。

答： 一.确定对系统的综合要求

1. 功能需求

这方面的需求指定系统必须提供的服务。通过需求分析应该划分出系统必须完成的所有功能。

2. 性能需求

性能需求指定系统必须满足的定时约束或容量约束，通常包括速度(响应时间)、信息量速率、主存容量、磁盘容量、安全性等方面的需求。

3. 可靠性和可用性需求

可靠性需求定量地指定系统的可靠性。

可用性与可靠性密切相关，它量化了用户可以使用系统的程度。

4. 出错处理需求

这类需求说明系统对环境错误应该怎样响应。例如，如果它接收到从另一个系统发来的违反协议格式的消息，应该做什么?注意，上述这类错误并不是由该应用系统本身造成的。

5. 接口需求

接口需求描述应用系统与它的环境通信的格式。常见的接口需求有：用户接口需求；硬件接口需求；软件接口需求；通信接口需求。

6. 约束

设计约束或实现约束描述在设计或实现应用系统时应遵守的限制条件。在需求分析阶段提出这类需求，并不是要取代设计(或实现)过程，只是说明用户或环境强加给项目的限制条件。常见的约束有：精度；工具和语言约束；设计约束；应该使用的标准；应该使用的硬件平台。

7. 逆向需求

逆向需求说明软件系统不应该做什么。理论上有限多个逆向需求，我们应该仅选取能澄清真实需求且可消除可能发生的误解的那些逆向需求。

8. 将来可能提出的要求

应该明确地列出那些虽然不属于当前系统开发范畴，但是据分析将来很可能会提出来的要求。

二 .分析系统的数据要求

任何一个软件系统本质上都是信息处理系统，系统必须处理的信息和系统应该产生的信息在很大程度上决定了系统的面貌，对软件设计有深远影响，因此，必须分析系统的数据要求，这是软件需求分析的一个重要任务。

三. 导出系统的逻辑模型

综合上述两项分析的结果可以导出系统的详细的逻辑模型，通常用数据流图、实体-联系图、状态转换图、数据字典和主要的处理算法描述这个逻辑模型。

四.修正系统开发计划

根据在分析过程中获得的对系统的更深入更具体的了解，可以比较准确地估计系统的成本和进度，修正以前制定的开发计划。

38.简述结构化程序设计的基本要点。

答: (1)设计软件系统结构（简称软件结构）

a.采用某种设计方法，将一个复杂的系统按功能划分成模块（划分）

- b.确定模块的功能。 (功能)
- c.确定模块之间的调用关系。 (调用)
- d.确定模块之间的接口,即模块之间传递的信息。 (接口)
- e.评价模块结构的质量。 (质量)

(2)数据结构及数据库设计

- a.数据结构设计
- b.数据库设计: (概念设计、逻辑设计、物理设计)

(3)编写概要设计文档 (文档主要有:概要设计说明书、数据库设计说明书、用户手册、修订测试计划)

(4)评审

39.简述提高软件可维护性的方法。

- A、建立明确的软件质量目标。B、利用先进的软件开发技术和工具。
- C、建立明确的质量保证工作。D、选择可维护的程序设计语言。
- E、改进程序文档。

31. 软件工程过程 即软件生命周期的方法、工具和概念

32. 模块化 模块化是指解决一个复杂问题时自顶向下逐层把系统划分成若干模块的过程,有多种属性,分别反映其内部特性.每个模块完成一个特定的子功能,所有的模块按某种方法组装起来,成为一个整体,完成整个系统所要求的功能

33. 集成测试 集成测试,也叫组装测试或联合测试。在单元测试的基础上,将所有模块按照设计要求 (如根据结构图) 组装成为子系统或系统,进行集成测试。

34. 软件可靠性 (1) 在规定的条件下,在规定的时间内,软件不引起系统失效的概率;

(2) 在规定的時間周期內,在所述条件下程序执行所要求的功能的能力;

35. 模块的控制范围

四、简答题 (本大题共 4 小题,每小题 5 分,共 20 分)

36. 简述软件工程面临的问题。

答: 内容: ①软件开发技术②软件开发管理

面临的主要问题: ①软件费用②软件可靠性③软件维护④软件生产率⑤软件重用

37. 简述可行性研究报告包含的主要内容。

技术可行性：对要开发项目的功能、性能、限制条件进行分析，确定在现有的资源条件下，技术风险有多大，项目是否能实现。包括：开发的风险；资源的有效性；技术；开发人员在评估技术可行性时，一旦估计错误，将会出现灾难性后果。

经济可行性：包括成本——效果分析、公司经营长期策略、开发所需的成本和资源、潜在的市场前景。

社会可行性包括：合同、责任、侵权、用户组织的管理模式及规范，其他一些技术人员常常不了解的陷阱等。

38. 简述软件结构设计的优化准则。

39. 简述快速原型的开发步骤。

答：快速原型开发步骤可划分下列阶段：

- (1) 快速分析：迅速确定基本需求、集中力量确定需求说明。
- (2) 快速构造原型：在快速分析基础上，在强有力的软件工具支持下，快速构造所需原型。
- (3) 运行原型：在开发者指导下，用户参与原型的运行，各类人员在共同运行原型中进一步加深对系统的了解及相互间的理解，以发现各种问题。
- (4) 评价原型：在运行基础上，根据原型目标，考核原型的特性，分析原型效果是否满足用户需求，提出修改意见。
- (5) 修改原型：在评价基础上进行修改。若不满足需求说明，则根据明确的需求修改原型。若不满足用户需求，则先修改并明确用户需求，再重新构造原型。