

第一章 绪论

考点一：软件工程概念

- 1、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。
- 2、**软件危机**：软件产率、软件质量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。
- 3、软件工程概念的提出是倡导以工程的**原理**、**原则**和**方法**进行软件开发，以期解决出现的软件危机。
- 4、软件工程的发展可分为两个时期：20 世纪 60 年代末到 80 年代初、20 世纪 80 年代以来。前期主要研究**系统实现技术**；后者主要表现为**软件质量**和**软件工程管理**。

考点二：软件开发的本质

- 1、软件开发的本质概括为：不同抽象层术语之间的“映射”，以及不同抽象层处理逻辑之间的“映射”。它涉及到两方面的问题：
 - (1) 一是如何实现这样的映射，这是技术层面上的问题；
 - (2) 二是如何管理这样的映射，以保障映射的有效性和正确性，这是管理层面上的问题。
- 2、计算机软件是计算机系统**中的程序及其文档**。
- 3、**软件开发的本质**可概括为：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**。实现这一映射的基本途径是系统建模。
- 4、在软件开发中，软件系统模型大体上分为两类：**概念模型**和**软件模型**。概念模型描述了系统是什么；软件模型描述了实现概念模型的软件解决方案。软件模型又分为设计模型、实现模型和部署模型等。

第二章 软件需求与软件需求规约

考点一：需求与需求获取

1、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：该需求只能以一种方式解析。
- (3) 可测的：该需求可进行测试的。
- (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。

2、验证需求是不是歧义的，一般可采用**需求复审**。验证需求是不是可测的，可在标识任何所需要的数据和设施的基础上，开发一个测试概念。验证需求是不是可测量的，可通过检验一个特征是否存在但需要考虑设计、实现和测试阶段所发生的各种情况。

3、常用的初始需求发现技术包括：

- (1) **自悟**：需求人员把自己作为系统的最终用户，审视该系统并提出问题。
- (2) **交谈**：为了确定系统应该提供的功能，需求人员通过提出问题/用户回答的方式，直接询问用户需要的是一个什么样的系统，**可能会引起客户抵触**。
- (3) 观察：通过观察用户执行其现行的任务和过程，了解系统运行环境。
- (4) 小组会：举行客户和开发人员的联系会议，与客户组织的一些代表共同发需求。
- (5) 提炼：复审技术文档，提取相关的信息。

4、问题空间理解：软件系统产品的需求分析工作中，面临的“三大挑战”的是问题空间理解、人与人之间的通信和需求的变化性。

5、设计约束是一种需求，它限制了软件系统或软件系统构件的设计方案的范围。例如：系统必须用 C++ 或其他面向对象语言编写，并且系统用户接口需要菜单；任取 1s，一个特定应用所消耗的可用计算能力平均不超过 50%。

考点二：需求规约

1、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。

2、需求规约般需要满足以下 4 个基本性质：

- (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。

3、需求规约的表达：

- (1) 非形式化的需求规约：以**一种自然语言**来表达需求规约。
- (2) 半形式化的需求规约：以**半形式化符号体系**来表达需求规约。
- (3) 形式化的需求规约：以一种**基于良构数学概念**的符号体系来编制需求规约。

4、需求规约的**作用**，其包含以下四点：

- (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) 对于产品/系统的设计，需求规约是一个正式的、受控的起始点。
- (3) 对于项目的其余大多数工作，需求规约是一个管理控制点。
- (4) 需求规约是创建产品验收计划和用户指南的基础。



第三章 结构化方法

考点一：结构化需求分析

学生成绩

1、数据流是数据的流动，表示形式例如： →

2、加工是数据的变换单元，即它接受输入的数据，对其进行处理并产生输出，用  表示。

3、数据存储是数据的静态结构，用于表达在分析中使用的、表达“结构化客体”的信息，用  表示。

4、数据源是数据流的起点；数据潭是数据流的归宿地，用  表示。

5、结构化分析方法给出了一种能表达功能模型的图形化工具是数据流图(DFD图)。

6、结构化分析建模的基本步骤：

- (1) 建立系统环境图，确定系统语境；
- (2) 自顶向下，逐步求精，建立系统的层次数据流图；
- (3) 定义数据字典；
- (4) 描述加工。

7、定义数据结构的符号：

符 号	描 述	符 号	描 述
=	定义为	{ }	重复
+	顺序	m..n	边界
[]	选择		

8、描述加工的工具：结构化自然语言、判定表、判定树。

考点二：结构化设计——总体设计

1、在结构化方法中，HIPO图应用在总体设计阶段，由IPO图和H图两部分组成的。

2、事务设计的基本步骤：

- (1) 设计准备，复审并精化系统模型；
- (2) 确定事务处理中心；
- (3) 设计系统模块结构图的顶层和第一层；
- (4) 自顶向下，逐步求精。

3、耦合是指不同模块之间相互依赖程度的度量。耦合程度从强到弱依次为：内容耦合、公共耦合、控制耦合、标记耦合、数据耦合。

(1) 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，这样的耦合称为内容耦合。

(2) 公共耦合：两个或连个以上的模块共同引用一个全局数据项，这种耦合被称为公共耦合。

(3) 控制耦合：一个模块通过接口向另一个模块传递一个控制信号，接收信号的模块根据信号值进行适当的动作，这种耦合称为控制耦合。

(4) 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，那么成模块B和C之间存在一个标记耦合。

(5) 数据耦合：模块之间通过参数来传递数据。

4、**内聚**是指**内部各成分之间相互关联程度**的度量。高内聚是指一个模块中各部分之间存在着很强的依赖；低内聚是指一个模块中各部分之间存在着较少的依赖。

5、常见的内聚类型从低到高分别为：**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**。

- (1) 偶然内聚：一个模块的各成分之间基本不存在任何关系。
- (2) 逻辑内聚：几个逻辑上相关的功能被放在同一个模块中。
- (3) 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。
- (4) 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。
- (5) 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。
- (6) 顺序内聚：一个模块的各个成分和同一个功能密切相关。
- (7) 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本能的。

6、在一个软件结果中，**深度**表示其控制的层数；**宽度**是指同一个层次上模块总数的最大值。**模块扇出**是指一个模块直接控制（调用）的下级模块数目；**模块扇入**表示有多少个上级模块直接调用它。

7、模块的**作用域**是指受该模块内一个判定所影响的所有模块的集合。

考点三：结构化设计——详细设计

1、详细设计的**任务**是具体描述模块结构图中的每一模块，即给出实现模块功能的实施机制，包括一组例程和数据结构，从而精确地定义了满足需求所规约的结构。

2、详细设计的**目标**是将总体设计阶段所产生的系统高层结构映射为以这些术语所表达的低层结构，也是系统的最终结构。

3、结构化程序设计的基本控制结构包括**顺序、选择和循环**。

4、典型的详细设计工具有：**程序流程图、盒图（N-S图）、PAD图**。

获取完整资料 请下载 APP 或关注公众号



扫码下载 app



扫码关注公众号