# 第5章：系统开发基础

【考点梳理】

## 考点1、开发模型（★★★★★）

### 【考法分析】

#### 本知识点的考查形式主要有：给定情景描述或特点描述，指出对应的开发模型；给出特点的开发模型，判断描述的正误；对于统一过程，判断具体任务完成的阶段；对于敏捷开发方法，判断描述正误和一些特点的归属。

### 【要点分析】

#### 1、瀑布模型：瀑布模型是将软件生存周期中的各个活动规定为依线性顺序连接的若干阶段的模型，包括需求分析、设计、编码、运行与维护。

#### 瀑布模型的特点是容易理解，管理成本低，每个阶段都有对应的成果产物，各个阶段有明显的界限划分和顺序要求，一旦发生错误，整个项目推倒重新开始。

#### 适用于需求明确的项目，一般表述为需求明确、或二次开发，或者对于数据处理类型的项目

#### 2、V模型：强调测试贯穿项目始终，而不是集中在测试阶段。是一种测试的开发模型。

#### 3、喷泉模型：典型的面向对象的模型。特点是迭代、无间隙。会将软件开发划分为多个阶段，但各个阶段无明显界限，并且可以迭代交叉。

#### 4、原型模型：典型的原型开发方法模型。适用于需求不明确的场景，可以帮助用户明确需求。

#### 5、增量模型：融合了瀑布模型的基本成分和原型实现的迭代特征，可以有多个可用版本的发布，核心功能往往最先完成，在此基础上，每轮迭代会有新的增量发布，核心功能可以得到充分测试。强调每一个增量均发布一个可操作的产品。

#### 6、螺旋模型：典型特点是引入了风险分析。结合了瀑布模型和演化模型的优点，最主要的特点在于加入了风险分析。它是由制定计划、风险分析、实施工程、客户评估这一循环组成的，它最初从概念项目开始第一个螺旋。属于面向对象开发模型，强调风险引入。

#### 7、统一过程（在软件设计师考试中UP、RUP都指统一过程）：典型特点是用例驱动、以架构为中心、迭代和增量。统一过程把一个项目分为四个不同的阶段：

#### 构思阶段 ：包括用户沟通和计划活动两个方面，强调定义和细化用例，并将其作为主要模型。

#### 细化阶段 ：包括用户沟通和建模活动，重点是创建分析和设计模型，强调类的定义和体系结构的表示。

#### 构建阶段 ：将设计转化为实现，并进行集成和测试。

#### 移交阶段 ：将产品发布给用户进行测试评价，并收集用户的意见，之后再次进行迭代修改产品使之完善

#### 8、敏捷开发是一种以人为核心、迭代、循序渐进的开发方法，适用于小团队和小项目，具有小步快跑的思想。常见的敏捷开发方法有极限编程法、水晶法、并列争球法和自适应软件开发方法。

#### （1）极限编程是一种轻量级的开发方法，它提出了四大价值观：沟通、简单、反馈、勇气。五大原则：快速反馈、简单性假设、逐步修改、提倡更改、优质工作。十二个最佳实践：计划游戏、隐喻、小型发布、简单设计、测试先行、重构、结对编程、集体代码所有制、持续集成、每周工作40小时、现场客户和编码标准。

#### （2）水晶法强调经常交付，认为每一种不同的项目都需要一套不同的策略、约定和方法论。

#### （3）并列争球法的核心是迭代、增量交付，按照30天进行迭代开发交付可实际运行的软件。

#### （4）自适应软件开发的核心是三个非线性的，重迭的开发阶段：猜测、合作、学习。

### 【备考点拨】

#### 1、掌握常见开发模型的特点，能够加以区分；

#### 2、掌握统一过程的4个阶段的任务；

#### 3、了解敏捷开发设计到的原则。

## 考点2、软件开发方法（★）

### 【考法分析】

#### 本知识点的考查形式主要是描述某种使用情景，要求考生判断适用的开发方法，或给出开发方法，判断描述的正误。

### 【要点分析】

#### 1、结构化开发方法特点：用户至上，严格区分工作阶段，每阶段有任务和结果，

#### 强调系统开发过程的整体性和全局性，系统开发过程工程化，文档资料标准化，

#### 自顶向下，逐步分解（求精）。

#### 2、原型开发方法：适用于需求不明确的情况。

#### 3、面向对象开发方法：更好的复用性，关键在于建立一个全面、合理、统一的模型，分析、设计、实现三个阶段，界限不明确。

#### 4、面向服务开发方法：面向对象更高标准的抽象。

### 【备考点拨】

#### 1、掌握不同开发方法的特点并加以区分。

## 考点3、需求分析（★）

### 【考法分析】

#### 本知识点的考查方式主要有：给定需求描述判断其需求分类；给定概念描述判断正误；指出需求分析的任务、成果产物、工具。

### 【要点分析】

#### 1、需求分析的任务是解决做什么的问题。

#### 2、需求的分类：

#### （1）功能需求-考虑系统要做什么，在何时做，在何时以及如何修改或升级。

#### （2）非功能需求：考虑软件开发的技术性指标，例如存储容量限制、执行速度、响应时间及吞吐率等。

#### （3）设计约束：除了功能需求和非功能需求以外的需求，例如操作系统限制、开发语言限制等。

#### 3、需求分析的工具有判定表、判定树、数据流图和数据字典。

#### 4、需求分析的产物有：需求规格说明书SRS。

### 【备考点拨】

#### 1、了解需求分析的任务、工具、产物等有哪些；

#### 2、掌握需求分析的分类。

## 考点4、软件设计（★★）

### 【考法分析】

#### 本知识点的主要考查形式有：给出软件设计相关描述（概念、原则等）判断正误；或给出一些情景描述指出其内聚类型或耦合类型。

### 【要点分析】

#### 1、软件设计的任务是解决怎么做的问题。软件设计包括体系结构设计、接口设计、数据设计和过程设计。

#### 过程设计：系统结构部件转换成软件的过程描述。

#### 结构设计：定义软件系统各主要部件之间的关系。

#### 接口设计（人机界面设计）：软件内部，软件和操作系统间以及软件和人之间如何通信。

#### 数据设计：将模型转换成数据结构的定义。好的数据设计将改善程序结构和模块划分，降低过程复杂性。

#### 2、系统方案设计包括总体设计和各部分的详细设计两个方面。

#### 系统总体设计：包括系统的总体架构方案设计、软件系统的总体架构设计、数据存储的总体设计、计算机和网络系统的方案设计等。

#### 系统详细设计：包括代码设计、数据库设计、人/机界面设计、处理过程设计等。

#### 3、软件设计的原则：高内聚、低耦合

#### （内聚性）

#### 偶然聚合:模块完成的动作之间没有任何关系，或者仅仅是一种非常松散的关系。

#### 逻辑聚合:模块内部的各个组成在逻辑上具有相似的处理动作，但功能用途上彼此无关。

#### 时间聚合:模块内部的各个组成部分所包含的处理动作必须在同一时间内执行。

#### 过程聚合:模块内部各个组成部分所要完成的动作虽然没有关系，但必须按特定的次序执行。

#### 通信聚合:模块的各个组成部分所完成的动作都使用了同一个数据或产生同一输出数据。

#### 顺序聚合:模块内部的各个部分，前一部分处理动作的最后输出是后一部分处理动作的输入。

#### （耦合性）

#### 功能聚合:模块内部各个部分全部属于一个整体，并执行同一功能，且各部分对实现该功能都比不可少

#### 非直接耦合:两个模块之间没有直接关系，它们的联系完全是通过主模块的控制和调用来实现的。

#### 数据耦合:两个模块彼此间通过数据参数交换信息。

#### 标记耦合:一组模块通过参数表传递记录信息，这个记录是某一个数据结构的子结构，而不是简单变量。

#### 控制耦合:两个模块彼此间传递的信息中有控制信息。

#### 外部耦合:一组模块都访问同一全局简单变量而不是同一全局数据结构，而且不是通过参数表传递该全局变量的信息。

#### 公共耦合:两个模块之间通过一个公共的数据区域传递信息。

#### 内容耦合:一个模块需要涉及到另一个模块的内部信息。

### 【备考点拨】

#### 1、掌握软件设计的阶段、任务和原则。

#### 2、区分各种内聚类型、耦合类型。

## 考点5、软件测试（★★★★）

### 【考法分析】

#### 本知识点的考查形式主要有：给定一些描述判断正误；或对于一些具体测试方法判断分类；以及根据要求设计测试用例或判断测试用例的个数。

### 【要点分析】

#### 1、常见的软件测试方法分类：

#### 

#### 2、常见的黑盒测试方法：

#### 等价类划分：确定无效与有效等价类，设计用例尽可能多的覆盖有效类，设计用例只覆盖一个无效类。

#### 边界值分析：处理边界情况时最容易出错，选取的测试数据应该恰好等于、稍小于或稍大于边界值。

#### 3、常见的白盒测试方法：

#### 语句覆盖：语句覆盖是指选择足够多的测试用例，使得运行这些测试用例时，被测程序的每个语句至少执行一次。【最弱覆盖】

#### 判定覆盖：判定覆盖又称为分支覆盖，它的含义是，不仅每个语句至少执行一次，

#### 而且每个判定的每种可能的结果（分支）都至少执行一次。

#### 条件覆盖：条件覆盖的含义是，不仅每个语句至少执行一次，而且使得判定表达

#### 式中的每个条件都取得各种可能的结果。【条件不一定包含判定，反之亦然】

#### 判定/条件覆盖：同时满足判定覆盖和条件覆盖的逻辑覆盖。

#### 路径覆盖：它的含义是，选取足够多的测试用例，使得程序的每条可能执行到的

#### 路径都至少经过一次（如果程序中有环路，则要求每天环路路径至少经过一次）

#### 【包含判定覆盖，但不包含条件覆盖】

#### 3、各测试阶段的任务：

#### （1）验收测试：有效性测试、软件配置审查、验收测试。

#### （2）系统测试：恢复测试、安全性测试、强度测试、性能测试、可靠性测试和安装测试。

#### （3）集成测试：模块间的接口和通信。

#### （4）单元测试：模块接口、局部数据结构、边界条件、独立的路径、错误处理。

#### （5）其他测试：回归测试（修改软件后进行的测试，防止引入新的错误）。负载测试（对软件负载能力的测试）。压力测试（对软件超负荷条件下运行情况的测试）。

### 【备考点拨】

#### 1、掌握常见的测试方法分类；

#### 2、掌握常见的黑盒测试方法和白盒测试方法。

#### 3、了解软件测试相关的概念，以及一些常见的测试阶段的任务。

## 考点6、McCabe复杂度计算（★★★）

### 【考法分析】

#### 本知识点考查的主要形式是给定伪代码或程序流程图，计算其McCabe复杂度。

### 【要点分析】

#### 1、McCabe复杂度计算公式：V(G)=m-n+2，其中m是有向弧的条数，n是结点数。

#### 2、对于伪代码可以先转换为程序流程图，对程序流程图可以最终转换为结点图处理，转换时注意将交点的地方标注为新的结点，以最终的结点图带入公式结算其McCabe复杂度。

### 【备考点拨】

#### 1、掌握伪代码🡪程序流程图🡪结点图的转换，及McCabe复杂度计算过程。

## 考点7、软件维护类型（★★★★）

### 【考法分析】

#### 本知识点的考查形式为给定情景描述判断其维护类型。

### 【要点分析】

#### 1、更正性维护：针对真实存在并已经发生的错误进行的维护行为。

#### 2、预防性维护：针对真实存在但还未发生的错误进行的维护。

#### 3、适应性维护：指使应用软件适应信息技术变化和管理需求变化而进行的修改。企业的外部市场环境和管理需求的不断变化也使得各级管理人员不断提出新的信息需求。

#### 4、完善性维护：扩充功能和改善性能而进行的修改。对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能与性能特征。

### 【备考点拨】

#### 1、掌握不同维护类型的特点并加以区分。

## 考点8、软件质量保证（★）

### 【考法分析】

#### 本知识点的考查形式主要是判断某些质量属性与其依从属性的对应关系。

### 【要点分析】

#### 

### 【备考点拨】

#### 1、了解软件质量属性及其对应的依从属性。

## 考点9、软件过程改进（★★）

### 【考法分析】

#### 本知识点考查的主要形式是给定描述判断其所属的CMMI的等级或层次；给定等级或层次，判断其任务、相关描述的正误。

### 【要点分析】

#### 1、阶段式模型：

#### 

#### 2、连续式模型：

#### CL0（未完成的）：过程域未执行或未得到CL1中定义的所有目标。

#### CL1（已执行的）：其共性目标是过程将可标识的输入工作产品转换成可标识的输出工作产品，以实现支持过程域的特定目标。

#### CL2（已管理的）：其共性目标是集中于已管理的过程的制度化。根据组织级政策规定过程的运作将使用哪个过程，项目遵循已文档化的计划和过程描述，所有正在工作的人都有权使用足够的资源，所有工作任务和工作产品都被监控、控制、和评审。

#### CL3（已定义级的）：其共性目标集中于已定义的过程的制度化。过程是按照组织的裁剪指南从组织的标准过程中裁剪得到的，还必须收集过程资产和过程的度量，并用于将来对过程的改进。

#### CL4（定量管理的）：其共性目标集中于可定量管理的过程的制度化。使用测量和质量保证来控制和改进过程域，建立和使用关于质量和过程执行的质量目标作为管理准则。

#### CL5（优化的）：使用量化（统计学）手段改变和优化过程域，以满足客户的改变和持续改进计划中的过程域的功效。

### 【备考点拨】

#### 1、了解软件过程改进模型各个层级、层次的特点，能够加以区分。

【项目管理】

## 考点10、Gantt图与Pert图（★★★★）

### 【考法分析】

#### 本知识点的考查形式主要有：判断对于Gantt图和Pert图的描述是否正确；根据图示找到Pert图的关键路径，计算其项目工期，计算某些活动的松弛时间，判断某些活动的最早/最晚开始或完成时间。

### 【要点分析】

#### 1、甘特图能够消晰描述每个任务的开始/结束时间及各任务之间的并行性，也可以动态地反映项目的开发进展情况，但难以反映多个任务之间存在的逻辑关系；PERT利用项目的网络图和各活动所需时间的估计值（通过加权平均得到的）去计算项目总时间，强调任务之间的先后关系，但不能反映任务之间的并行性，以及项目的当前进展情况。

#### 2、关键路径法是图中源点至汇点的最长路径，关键路径的时间称之为项目工期，也表述为项目完成所需的最少时间。

#### 3、总时差：在不延误总工期的前提下，该活动的机动时间。一般在图中，以最晚结束时间减去最早结束时间求取，或以最晚开始时间减去最早开始时间求取。

#### 4、对于网络图我们一般采用关键路径分析法处理，关键路径法是利用进度模型时使用的一种进度网络分析技术。沿着项目进度网络路线进行正向与反向分析，从而计算出所有计划活动理论上的最早开始与完成日期、最迟开始与完成日期，不考虑任何资源限制。

#### 5、单代号网络图：结点表示活动，箭线表示活动与活动间的依赖关系。

#### 

#### 6、双代号网络图：结点表示里程碑，箭线表示活动。

#### 

### 【备考点拨】

#### 1、掌握Gantt图与Pert图相关的概念；

#### 2、掌握Pert图分析的方法，能够标注出单代号网络图和双代号网络图中的最早开始/结束数据，和最晚开始/结束时间，并以此找出图中的关键路径，掌握相关参数的计算方法。

## 考点11、风险管理（★★★）

### 【考法分析】

#### 本知识点的考查形式，主要有：对风险相关概念描述判断正误；计算风险曝光度。

### 【要点分析】

#### 1、风险的特性：具有不确定性，可能会造成损失。

#### 2、风险的类别：项目风险涉及到各种形式的预算、进度、人员、资源以及客户相关的问题，并且可能导致项目损失。技术风险涉及到技术相关的可能会导致项目损失的问题。商业风险与市场因素相关。社会风险涉及到政策、法规等因素。

#### 3、风险暴露又称风险曝光度，测量的是资产的整个安全性风险，它将表示实际损失的可能性与表示大量可能损失的资讯结合到单一数字评估中。在形式最简单的定量性风险分析中，风险曝光度可透过将风险可能性及影响相乘算出。

#### 风险曝光度（RiskExposure）=错误出现率（风险出现率）X错误造成损失（风险损失）。

### 【备考点拨】

#### 1、掌握风险相关的一些概念；

2、掌握风险曝光度的计算。