

# 计算机考研复试面试常问问题 软件工程篇

---

## 1、什么是软件工程?目前有哪几种主要的软件工程方法?

概括地说,软件工程是指导计算机软件开发和维护的一门工程学科.采用工程的概念、原理、技术、和方法来开发与维护软件,把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来,以经济地开发出高质量的软件并有效地维护它,这就是软件工程。它涉及程序设计语言、数据库、软件开发工具、系统平台、标准、设计模式等方面。

主要有传统方法也称为经典软件工程方法或生命周期方法或结构化范型和面向对象方法。

## 2、软件的生命周期

软件生命周期又称为软件生存周期或系统开发生命周期,是软件的产生直到报废的生命周期。软件生存周期包括:

- (1) 问题定义:弄清"用户需要计算机解决什么样的问题",提出"系统目标和范围的说明",提交用户审查和确认。
- (2) 可行性分析:把待开发系统的目标以明确的语言描述出来,并从经济、技术、法律等多个方面进行可行性分析。
- (3) 需求分析:弄清用户对软件系统的全部需求,编写需求规格说明书和初步的用户手册,提交评审。
- (4) 开发阶段:设计、实现(完成源程序的编码)、测试
- (5) 维护:改正性维护(由于开发测试的不彻底、不完全),适应性维护(适应环境变化),完善性维护(使用过程中提出的一些建设性意见),预防性维护(改善软件系统的可维护性和可靠性)。

## 3、软件过程

软件生命周期所涉及的一系列相关过程，是指一套关于项目的阶段、状态、方法、技术和开发、维护软件的人员以及相关Artifacts（计划、文档、模型、编码、测试、手册等）组成。包含基本过程类、支持过程类、组织过程类。

- (1) 基本过程类包括获取过程、供应过程、开发过程、运作过程、维护过程和管理过程。
- (2) 支持过程类包括文档过程、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、审计过程以及问题解决过程。
- (3) 组织过程类包括基础设施过程、改进过程、培训过程。

#### 4、软件需求分析的任务是什么？有哪些主要步骤？

需求分析的基本任务是深入描述软件的功能和性能、确定软件设计的约束和软件同其它系统元素的接口细节、定义软件的其它有效性需求，总之，需求分析的任务就是借助于当前系统的逻辑模型导出目标系统的逻辑模型，解决目标系统的“做什么”的问题。

**主要步骤：**

##### 1.问题识别

- (1)功能需求：明确所开发的软件必须具备什么样的功能。
- (2)性能需求：明确待开发的软件的技术性能指标。
- (3)环境需求：明确软件运行时所需要的软、硬件的要求。
- (4)用户界面需求：明确人机交互方式、输入输出数据格式。

##### 2.分析与综合，导出软件的逻辑模型

分析人员对获取的需求，进行一致性的分析检查，在分析、综合中逐步细化软件功能，划分成各个子功能。用图文结合的形式，建立起新系统的逻辑模型。

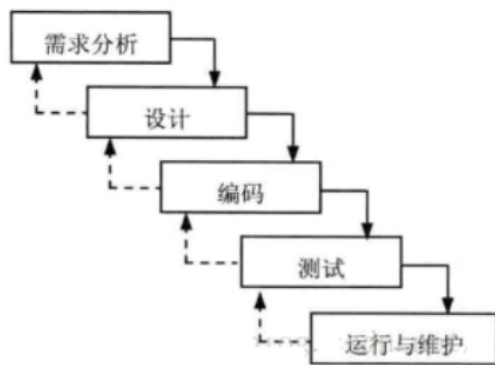
##### 3.编写文档

- (1)编写“需求规格说明书”，把双方共同的理解与分析结果用规范的方式描述出来，作为今后各项工作的基础。
- (2)编写初步用户使用手册，着重反映被开发软件的用户功能界面和用户使用的具体要求，用户手册能强制分析人员从用户使用的观点考虑软件。
- (3)编写确认测试计划，作为今后确认和验收的依据。
- (4)修改完善软件开发计划。在需求分析阶段对待开发的系统有了更进一步的了解，所以能更准确地估计开发成本、进度及资源要求，因此对原计划要进行适当修正。

#### 5、软件工程模型

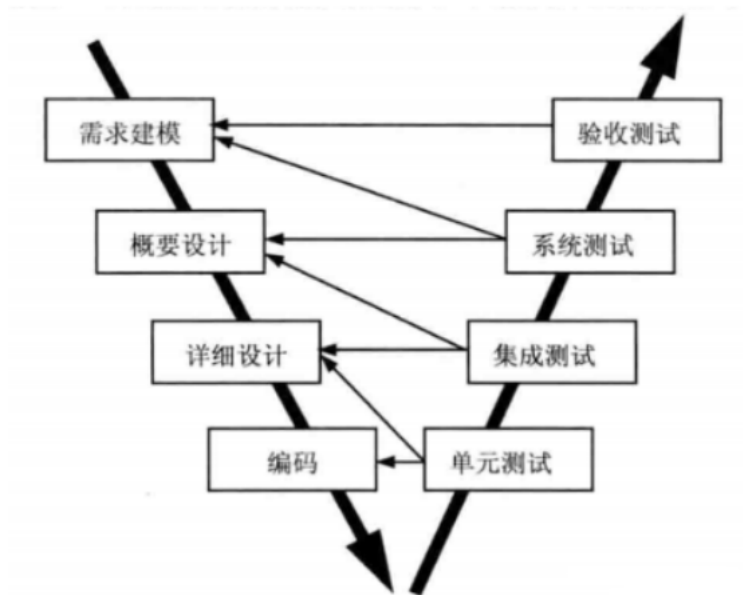
常见软件工程模型如下：

- 1) **瀑布模型**：将软件生命周期中的各个活动规定为线性连接的模型，包括需求分析、设计、编码、测试、运行与维护，由前至后、相互衔接的固定顺序，如同瀑布流水逐级下落。



瀑布模型是以文档作为驱动、适合于软件需求很明确的软件项目的模型。

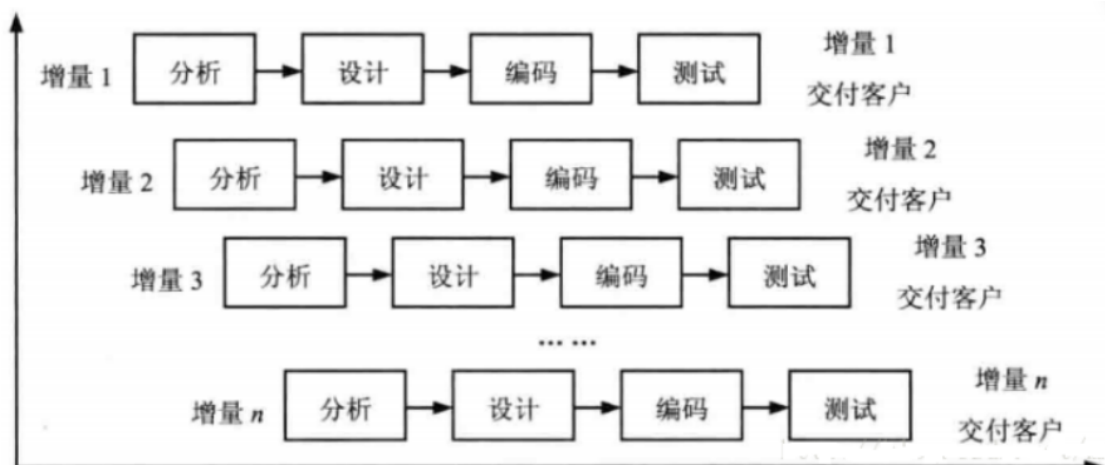
2) **V模型**：瀑布模型的一个变体，提供了一种验证确认活动应用于早期软件工程工作中的方法。



瀑布模型的优点：1) 容易理解，管理成本低；2) 强调开发的阶段性早期计划及需求调查和产品测试。

瀑布模型的缺点：1) 客户必须能够完整、正确和清晰地表达他们的需要；2) 开始2个或3个阶段，很难评估真正的进度；3) 项目结束时，出现大量的集成和测试工作；4) 需求或设计中的错误往往只有到了项目后期才能够被发现，对于项目风险的控制能力较弱，从而导致项目常常延期完成，开发费用超出预算。

3) **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，它假设可以将需求分段为一系列增量产品，每一增量可以分别开发。



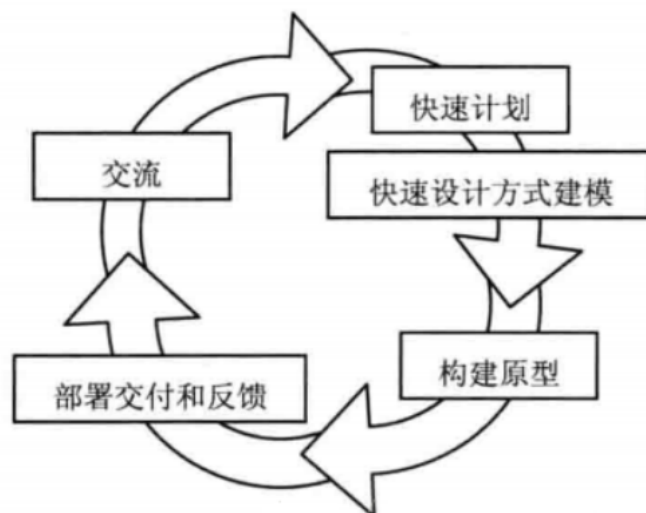
使用增量模型，第1个增量往往是核心的产品。客户对每个增量的使用和评估都作为下一个增量发布的新特征和功能，这个过程在每一个增量发布后不断重复，直到产生了最终的完善产品。增量模型强调每一个增量均发布一个可操作的产品。

增量模型的优点：1) 容易理解，管理成本低；2) 强调开发的阶段性早期计划及需求调查和产品测试；3) 第一个可交付版本所需要的成本和时间很少；4) 开发由增量表示的小系统所承担的风险不大；5) 减少用户需求的变更；6) 运行增量投资，即在项目开始时，可以仅对一个或两个增量投资。

增量模型的缺点：1) 如果没有对用户的变更需求进行规划，那么产生的初始增量可能会造成后来增量的不稳定；2) 如果需求不想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) 管理发生的成本、进度和配置的复杂性可能会超出组织的能力。

4) **演化模型**：是迭代的过程，软件开发人员能逐步开发出更完整的软件版本，适用于软件需求缺乏准确认识的情况，典型的演化模型有原型模型和螺旋模型。

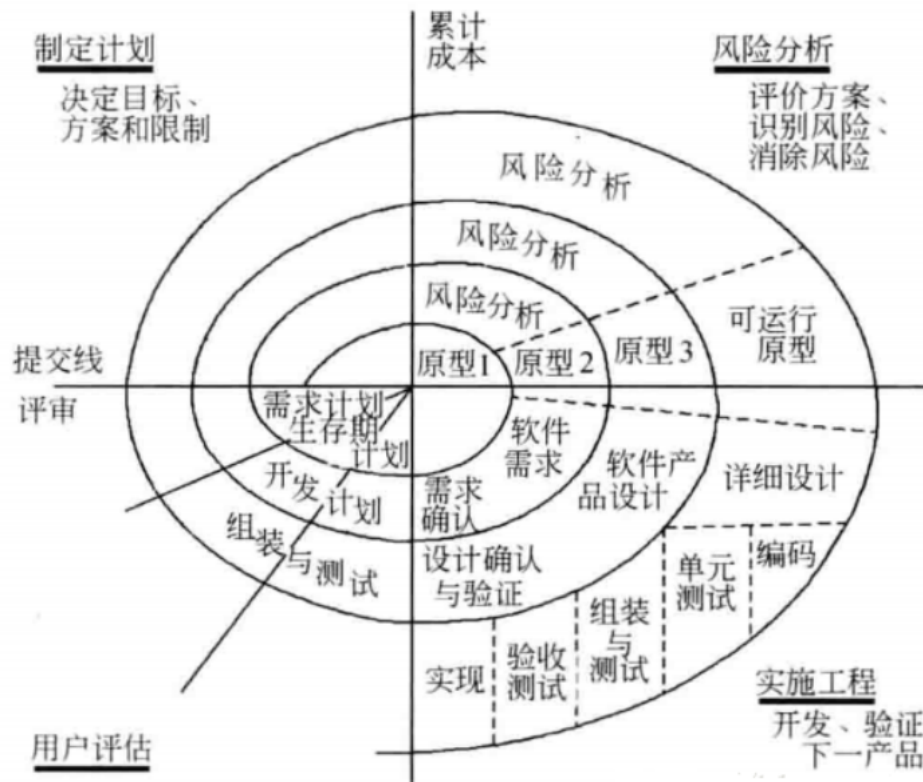
4. 1) **演化模型之原型模型**：是预期系统的一个可执行版本，反映了系统性的一个选定的子集，一个原型不必满足目标软件的所有约束，目的是能快速、低成本地构建原型。



原型模型开始于沟通，其目的是定义软件的总体目标，标识需求，然后快速制定原型开发的计划，确定原型的目标和范围，采用快速射击的方式对其进行建模，并构建原型。

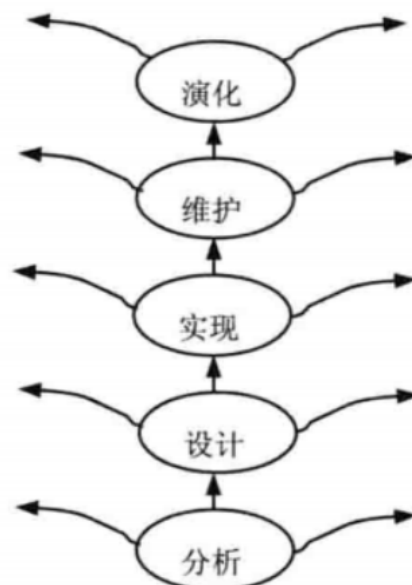
根据原型的目的，可分为三种：1) 探索型原型：目的是弄清目标的要求，确定所希望的特性，并探讨多种方案的可行性；2) 实验型原型：目的是验证方案或算法的合理性，是在大规模开发和实现前，用于考查方案是否合适、规格说明是否可靠等；3) 演化型原型：目的是将原型作为目标系统的一部分，通过对原型的多次改进，逐步将原型演化成最终的目标系统。

4. 2) **演化型模型之螺旋模型**：将瀑布模型与演化模型结合起来，加入了两种模型均忽略的风险分析，弥补了这两种模型的不足。螺旋模型将开发过程分为几个螺旋周期，每个螺旋周期大致和瀑布模型相符合：



螺旋模型强调风险分析，使得开发人员和用户对每个演化层出现的风险有所了解，从而做出应有的反应。因此，该模型特别适用于庞大、复杂并且具有高风险的系统。

5) **喷泉模型**: 一种以用户需求为动力, 以对象作为驱动的模型, 适合于面向对象的开发方法。它克服了瀑布模型不支持软件重用和多项开发活动集成的局限性, 喷泉模型使开发过程具有迭代性和无间隙性。



优点：提高软件项目的开发效率，节省开发时间。

缺点：开发阶段是重叠的，开发过程中需要大量的开发人员，不利于项目的管理。需要严格的管理文档，使得审核的难度加大。

### 6、简述结构化分析、设计的要点:

结构化分析方法适合于数据处理类型软件的需求分析。其要点是"自顶向下"地开发系统,由整体到各组成部分,由表及里,由抽象到具体,逐步求精。

(1) 模块化 (2)由顶向下,逐步求精.(3)上层模块分解为下层模块,有三种不同的结构形式,即顺序结构,选择结构和循环结构。

## 7、软件测试的目标是什么,有哪几种主要有测试方法? 软件测试的目标:

(1)测试是为了发现程序中的错误而执行程序的过程;

(2)好的测试方案是极可能发现迄今为止尚未发现的错误的测试方案;

(3)成功的测试是发现了至今为止尚未发现的错误的测试。

软件测试的方法有动态测试、静态测试和正确性证明。

a.黑盒测试: 黑盒测试是将被测软件看出一个黑盒子,只考虑系统的输入和输出,完全不考虑程序内部的逻辑结构和处理过程。黑盒测试的依据是开发各阶段的需求规格说明。

b.白盒测试: 白盒测试是将黑盒子打开,研究源代码和程序内部的逻辑结构;

## 8、白盒测试主要有哪些覆盖?

语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖、点覆盖、边覆盖、路径覆盖

## 9、选择一种程序设计语言的主要有哪些依据?

为了使程序容易测试和维护以减少生命周期的总成本,选用的高级语言应该有理想的模块化机制,以及可读性好的控制结构和数据结构;

为了便于调试和提高软件可靠性,语言特点应该使编译程序能够尽可能多地发现程序中的错误;

为了降低软件开发和维护的成本,选用的语言应该有良好的独立编译机制。

上述这些要求是选择语言的理想标准,但是在实际选用语言时不能仅仅考虑理论上的标准,还必须同时考虑实用方面的各种限制。(1)系统用户的要求 (2)可以使用的编译程序 (3)可以得到的软件工具 (4)系统规模 (5)程序员的知识 (6)软件可移植性要求 (7)软件的应用领域

## 10、软件的维护的目标是什么,有哪几种维护类型?

纠正在使用过程中暴露出来的错误而进行的改进性维护,适应外部环境的变化而进行的适应性维护,改进原有的软件而进行的完善性维护,以及改进将来的可维护性和可靠性而进行的预防性维护。软件维护的分类: 软件维护主要划分为纠错性维护、适应性维护和完善性维护。

(1)纠错性维护。由于前期的测试不可能揭露软件系统中所有潜在的错误,用户在使用软件时仍将会遇到错误,诊断和改正这些错误的过程称为纠错性维护。

(2)适应性维护。由于新的硬件设备不断推出,操作系统和编译系统也不断地升级,为了使软件能适应新的环境而引起的程序修改和扩充活动称为适应性维护。

(3)完善性维护。在软件的正常使用过程中,用户还会不断地提出新的需求。为了满足用户新的需求而增加软件功能的活动称为完善性维护。

## 11、简述提高软件质量的主要措施。

复审:是在软件生命周期每个阶段结束之前,都采用一定的标准对该段产生的软件配置成分进行严格的正式或非正式的检测。

复查:是检查已有的材料,以断定在软件生命周期某个阶段的工作是否能够开始或继续。

管理复审:是向开发组织或使用部门的管理人员提供有关项目的总体状况、成本和进度等方面的情况,以便他们从管理角度对开发工作进行审查。

测试:包括测试计划、测试过程和测试结果3个阶段。



## 12、面向对象如何实现模块独立性，其耦合和内聚的含义是什么？

因为对象是由数据及可以对这些数据施加的操作所组成的统一体，而且对象是以数据为中心的，操作围绕对其数据所需做的处理来设置，没有无关的操作。因此，对象内部各种元素彼此结合得很紧密。内聚性相当强，由于完成对象所需要的元素（数据和方法）基本上都被封装在对象内部，它与外界的联系自然就比较少。因此，对象之间的耦合通常比较松。总之，面向对象使用对象、类、继承和消息的方法，既使用类和继承等机制，而且对象之间仅能通过传递消息实现彼此通信来实现模块的独立性。

## 13、面向对象和面向过程软件工程有哪些区别？

(1) 面向过程就是分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现，使用的时候一个一个依次调用就可以了；面向对象是把构成问题事务分解成各个对象，建立对象的目的不是为了完成一个步骤，而是为了描述某个事物在整个解决问题的步骤中的行为。

(2) 面向过程是把一件事一项工程分解成为一个个小的功能,用一个个函数来实现；面向对象是把事情看成是一个个小的对象组成的,或者说一个个小部分组成的,这些对象之间的相互关系,构成了整个项目.在面向对象的思想中，万物皆对象。而“类”，就是对象的抽象或者说是概括。

## 14、简述对象、类、消息、方法的基本概念。

(1)对象是人们要进行研究的任何事物，从最简单的整数到复杂的飞机等均可看作对象，它不仅能表示具体的事物，还能表示抽象的规则、计划或事件。

(2) 类是具有相同或相似性质的对象的抽象。对象的抽象是类，类的具体化就是对象，也可以说类的实例是对象。类具有属性，它是对象的状态的抽象，用数据结构来描述类的属性。类具有操作，它是对象的行为的抽象，用操作名和实现该操作的方法来描述。

(3)对象之间进行通信的结构叫做消息。在对象的操作中，当一个消息发送给某个对象时，消息包含接收对象去执行某种操作的信息。发送一条消息至少要包括说明接受消息的对象名、发送给该对象的消息名（即对象名、方法名）。一般还要对参数加以说明，参数可以是认识该消息的对象所知道的变量名，或者是所有对象都知道的全局变量名。

(4) 类中操作的实现过程叫做方法，一个方法有方法名、参数、方法体。

## 15、简述面向对象分析设计的三个模型。

答：三个模型：对象模型、动态模型、功能模型

(1) 对象模型描述系统的静态结构，包括类和对象，它们的属性和操作，以及它们之间的关系。构造对象模型的目的在于找出与应用程序密切相关的概念。对象模型用包含对象及对象的关系图表示。

(2) 动态模型着重于系统的控制逻辑，考察在任何时候对象及其关系的改变，描述这些涉及时序和改变的状态。动态模型包括状态图和事件跟踪图。状态图是一个状态和事件的网络，侧重于描述每一类对象的动态行为。事件跟踪图则侧重于说明系统执行过程中的一个特点“场景”，也叫做脚本(scenarios)，是完成系统某个功能的一个事件序列。脚本通常起始于一个系统外部的输入事件，结束于一个系统外部的输出事件。

(3) 功能模型着重于系统内部数据的传送和处理。功能模型表明，通过计算，从输出数据能得到什么样的输出数据，但不考虑参加计算的数据按什么时序执行。功能模型由多个数据流图组成，它们指明从外部输出，通过操作和内部存储，直到外部输出的整个数据流情况。功能模型还包括了对象模型内部数据间的限制。功能模型中的数据流图往往形成一个层次结构，一个数据流图的过程可以由下一层的数据流图作进一步的说明。

## 16、UML 里面有哪些图？

UML 图包括九种：使用案例图、类图、对象图、构件图、部署图、活动图、协作图、状态图、序列图。在这些图中使用案例图、类图、序列图是最有用的。

#### 1、需求

采用用例图描述需求。

#### 2、分析

采用类图描述静态结构

采用顺序图、合作图、活动图、状态图描述动态行为

#### 3、设计

采用类图、包，对类的接口进行设计

#### 4、实现

将类用某现象对象语言实现

#### 5、继承与交付

构件图、包、部署图

#### 6、测试

单元测试——类图和类的说明书

### 17、软件系统的三个测试阶段

开发测试：在开发过程中对系统进行测试以发现bug和缺陷。

发布测试：在系统发布给用户之前，单独的测试团队测试系统的完整版本。

用户测试：其中系统的用户或潜在用户在自己的环境中测试系统。