

软件工程复习笔记

徐大鹏

2017年6月18日

软件工程大水课！一定要考好！求保佑！

1 软件工程概述

1.1 复习提纲里的考点

1. SE的定义、目的、方法及作用(P2 / P16)

定义是什么？方法呢？作用呢？不知道。

- 章前简介：我们的最终目标是，生产出高质量软件，进而找到解决方案，并考虑那些对质量有影响的特性。
- 1.2节：要写出健壮的、易于理解和维护的并且能以最高效的方式完成工作的代码，必须具备专业软件工程师的技巧和洞察力。因此软件工程的目标就是设计和开发高质量软件。
- 1.1.2节：软件工程师的角色：软件工程师的精力集中于将计算机作为求解问题的工具，而不是研究硬件设计或者算法的理论证明。

2. 说明**错误、缺陷、失败**的含义与联系。（请举例说明）（6页）（44页习题3）

当人们在进行软件开发活动的过程中出错（错误）时，就会出现故障。失效是指系统违背了它应有的行为。故障是系统的内部视图，是从开发人员的角度看待系统；失效是系统的外部视图，是从用户的角度。见图1。

3. 软件质量应从哪几个方面来衡量？(P9 – P12)

产品的质量 用户角度：易于学习、易于使用；故障的数目少，故障类型都是次要的（次要的、主要的、灾难性的）。设计和编写代码的人



图1-4 人为错误是如何引起失效的

Figure 1: 错误和失效

员、维护该程序的人员：考虑产品的内部特性，把故障的数目和类型看做产品质量的证据。

过程的质量 只要有活动出了差错，产品的质量就会受到影响。提出问题：What? When? Where? How?

商业环境背景下的质量 提供的产品和服务。

4. 现代软件工程大致包含的几个阶段及各个阶段文档(P23-P24)

1.6.2: 构建系统

需求分析和定义 与客户会面以确定需求，这些需求是对系统的描述。

系统设计 系统设计告诉客户，从客户的角度看，系统会是什么样的。然后客户要对设计进行评审。当设计得到批准之后，整个系统设计将被用来生成其中单个程序的设计。

程序设计

编写程序

单元测试 链接之前作为单独的代码段进行测试。

集成测试 将模块组合到一起，确保他们能够正确运行。

系统测试 对整个系统的测试，用于确保起初指定的功能和交互得以实现。

系统交付

维护 出现任何问题，或者需求发生变化时。

5. 什么是抽象？(P30)

1.8.2节。

书上有一句话定义：抽象是在某种概括层次上对问题的描述，使得我们能够集中于问题的关键方面而不会陷入细节。

6. 什么是软件过程？软件过程的重要性是什么？包含几个阶段？(P32, P45)

1.8.2节。图2

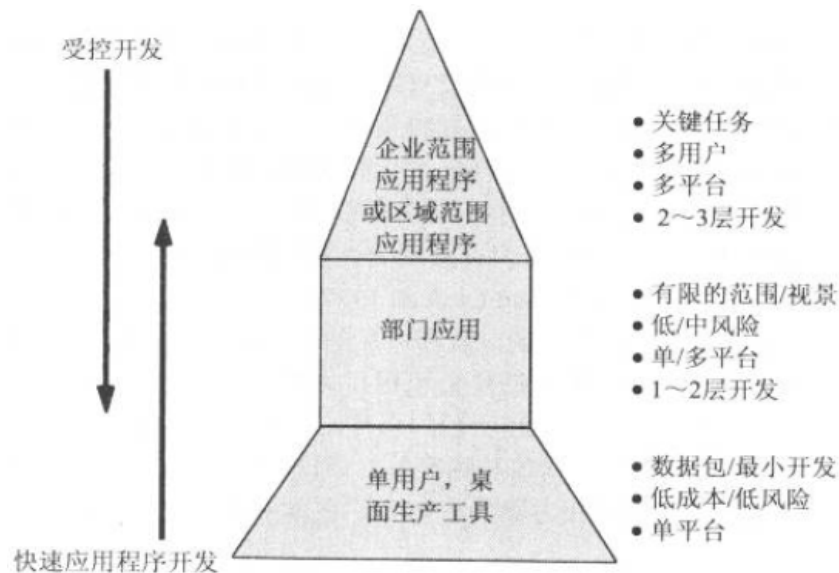


图1-14 不同开发中的差别 (Wasserman 1996)

Figure 2: 软件过程的差别

7. 什么是重用等软件工程主要概念？(P34)

1.8.2节。图3

在软件开发和维护中，通常通过复用以前开发项目中的项来利用应用程序之间的共性。例如，在不同的开发项目中，我们使用同样的操作系统和数据库管理系统，而不是每次都构建一个新的。类似地，当我们构建一个与以前做过的项目类似但有所不同的系统时，可以复用需求集、部分设计以及测试脚本或数据。Barnes和Bollinger指出，复用并不是一个新的思想，他们还给出了很多有趣的例子，说明复用的不仅仅是代码 (Barnes and Bollinger 1991)。

Prieto-Diaz介绍了这样一种理念：可复用构件是一种商业资产 (Prieto-Diaz 1991)。公司和组织机构对那些可复用的项进行投资，而当这些项再次用于后面的项目中的时候，就可以获得巨大的收益。但是，制定一个长期、有效的可复用计划可能是很困难的，因为存在如下这些障碍。

Figure 3: 复用

2 模型化过程和生命周期

2.1 提到的考点

1. 什么是软件过程？软件过程的重要性是什么？(P45-46)
2. 瀑布模型及各阶段文档，优缺点？(P49)
3. 原型的概念(P51)
4. 论述分阶段开发模型的含义，其基本分类及特点是什么？(56 页)
5. 螺旋模型四个象限的任务及四重循环的含义？(P58) P80-81 页习题2，3。
6. 什么是UP，RUP？

3 计划和管理项目

3.1 复习提纲里的考点

1. 什么是项目进度？活动？里程碑？(83页)

项目进度 通过列举项目的各个阶段，把每个阶段分解成离散的任务或活动，来描述特定项目的软件开发周期。

活动 是项目的一部分，在一段时间内发生。

里程碑 是活动完成的时刻。

2. 如何计算软件项目活动图的关键路径？冗余时间？最早和最迟开始时间(习题2, 3) (课堂习题讲解)

关键路径就是最长路径，即每一个节点的时差都为零的路径。冗余时间也就是时差，满足

$$\text{时差} = \text{可用时间} - \text{真实时间} = \text{最晚开始时间} - \text{最早开始时间}$$

计算上，要先求出最长路径，然后沿最长路径回溯，找到每一个活动的最早开始时间和最晚开始时间，然后求出每一个活动的时差。

3. 软件团队人员应该具备的能力是什么？(96 页)
3.2.1节。不明白。

4. 软件项目组织的基本结构？(101 页)

主程序员负责制和忘我方法。根据实际情况可以结合这两种极端情况。
主程序员负责制的组织结构如何？忘我方法适于那些情况？他们的对比？（对比见图4）

表3-5 组织结构的比较	
高度结构化	松散的结构
高度确定性	不确定性
重复	新技术或工艺
大型项目	小型项目

Figure 4: 组织结构的对比

5. 试述COCOMO模型的三个阶段基本工作原理或含义。(111 页)

3.3.2节。见图5

在阶段1，项目通常构建原型以解决包含用户界面、软件和系统交互、性能和技术成熟性等方面在内的高风险问题。这时，人们对正在创建的最终产品的可能规模知之甚少，因此COCOMO II用应用点（其创建者对它的命名）来估算规模。正如我们将看到的，这种技术根据高层的工作量生成器（如屏幕数量和报告数量、第3代语言构件数）来获取项目的规模。

在阶段2（即早期设计阶段），已经决定将项目开发向前推进，但是设计人员必须研究几种可选的体系结构和操作的概念。同样，仍然没有足够的信息支持准确的工作量和工期估算，但是远比第1阶段知道的信息要多。在阶段2，COCOMO II使用功能点对规模进行测量。功能点是在参考文献IFPUG（1994a and b）中详细讨论的一种技术，估算在需求中获取的功能。因此，与应用点相比，它们提供了更为丰富的系统描述。

在阶段3（后体系结构阶段），开发已经开始，而且已经知道了相当多的信息。在这个阶段，可以根据功能点或代码行来进行规模估算，而且可以较为轻松地估算很多成本因素。

Figure 5: COCOMO模型的三个阶段

6. 什么是软件风险？主要风险管理活动？有几种降低风险的策略？(P119, P122)

易。

7. 找出图3.23和图3.24(P139)的关键路径。

章末必做练习题。

4 获取需求

4.1 提到的考点

1. 需求的含义是什么？(143 页)

需求就是对期望的行为的表达。需求指定客户想要什么行为，而不是如何实现这些行为。

2. 需求作为一个工程，其确定需求的过程是什么？(144 页图4.1)

图6

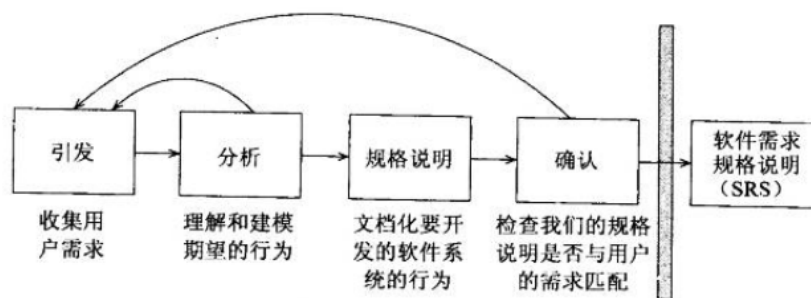


图4-1 获取需求的过程

Figure 6: 获取需求的过程

3. 举例说明获取需求时，若有冲突发生时，如何考虑根据优先级进行需求分类。(152 页)
4.3.1节。
4. 需求文档分为哪两类？(153 页)
4.3.2节。
5. 什么是功能性需求和非功能性需求/质量需求？设计约束？过程约束？(149 页)
4.3节开头。
6. 知道DFD图的构成及画法(172 页)
4.5.8节
7. 在需求原型化方面，什么是抛弃型原型？什么是演化型原型？(192-193 页)
4.7节末尾。

5 设计体系结构

1. 什么是软件体系结构？设计模式？设计公约？设计？概念设计？技术设

计? (223-224 页)

5.1节开头，5.1.1节。概念设计和技术设计没有找到。

2. 软件设计过程模型的几个阶段?

跟第四章第二个提到的考点差不多。图7

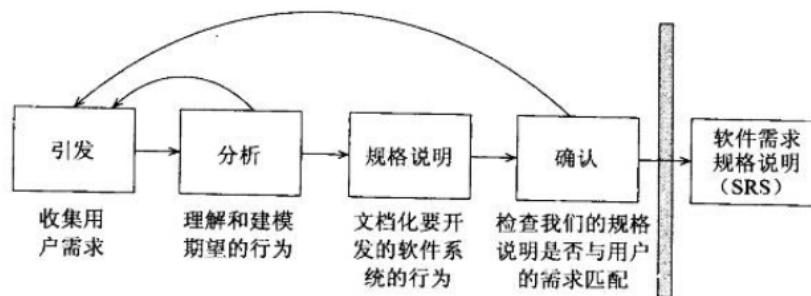


图4-1 获取需求的过程

Figure 7: 获取需求的过程

3. 什么是模块化? 什么是抽象? (238 页)

4. 论述设计用户界面应考虑的问题。(242 页)

5. 5.5 节—模块独立性—耦合与内聚的概念及各个层次划分? (248—xxx 页)

6. 举例说明耦合与内聚的基本分类。以及各个分类的含义与特征(284 页习题4, 5)

6 设计模块

1. 什么是设计模式?

见设计体系结构。

2. OO设计的基本原则?

OO 开发有何优势? (291 页) OO 开发过程有几个步骤? (292 页) 熟悉用例图的组成和画法, 用例的几个要素的含义, 掌握用例图的实例解析方法(294 页) 用例图、类图等对面向对象的项目开发的意义是什么? 熟悉类图中各个类之间的基本关系分类(303-305) 熟悉类图等的组成和画法(300-308 页) 知道UML 其他图示结构的基本用途。

//为什么说编码工作是纷繁复杂甚至令人气馁? (337 页) //一般性的编程原则应该从哪三个方面考虑? (340-344 页) //论述编码阶段实现某种算法时所涉及的问题。(342 页) 在编写程序内部文档时, 除了HCB 外, 还应添加什么注释信息? (352-354 页) 什么是极限编程(XP)? 以及派对编程? (357 页)

// 产生软件缺陷的原因? (365 页) // 将软件缺陷进行分类的理由? (367 页) 几种主要的缺陷类型? (367-368 页) 什么是正交缺陷分类? (369 页) 测试的各个阶段及其任务? (372 页图8.3) // 测试的态度问题? (为什么要独立设置测试团队?)(373 页) 掌握测试的方法——黑盒、白盒的概念? (374) 什么是单元测试? 什么是走查和检查? (376 页) 黑盒白盒方法各自的分类? 测试用例的设计和给出方法(结合补充材料) 黑盒白盒方法的分类, 各种覆盖方法等。(课件和补充课件) 如何面对一个命题, 设计和给出测试用例的问题。(课件) ——课堂练习的测试题目和讲解内容集成测试及其主要方法的分类? (390-392)(驱动, 桩的概念) // 传统测试和OO 测试有何不同? OO 测试有何困难? (398-399 页) // 测试计划涉及的几个步骤? (400 页) (了解)

系统测试的主要步骤及各自含义? (420 页, 图9.2) 什么是系统配置? 软件配置管理? // 基线? (423 页)(或见课件) // 什么是回归测试? (425 页) 功能测试的含义及其作用? (430 页) 功能测试的基本指导原则? (431) 性能测试的含义与作用? (436 页) 性能测试的主要分类? (436 页) // 什么是可靠性、可用性和可维护性? (438 页) 确认测试, 确认测试分类? (基准测试和引导测试)(447-448 页) 什么是alpha 测试? β 测试? (448 页) // 什么是安装测试? (450 页)