

第 1 章 软件工程概述

1.1 软件危机

1. 软件 = 程序+数据+文档
2. 软件是能够完成**预定功能和性能的可执行的计算机程序**和使程序正常执行所需的数据, 加上描述程序的操作和使用的文档。
3. **软件危机**是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。
软件危机**包含**下述两方面的问题:
 - 1) 如何**开发**软件, 以满足对软件日益增长的需求
 - 2) 如何**维护**数量不断膨胀的已有软件
4. 软件危机的**典型表现**:
 - 1) 对软件开发成本和进度的估计常常很不正确
 - 2) 用户对“已完成的”软件系统不满意的现象经常发生
 - 3) 软件产品的质量往往靠不住
 - 4) 软件常常是不可维护的
 - 5) 软件通常没有适当的文档资料
 - 6) 软件成本在计算机系统总成本中所占的比例逐年上升
 - 7) 软件开发生产率提高的速度远远跟不上计算机应用迅速普及深入的趋势
5. 软件危机产生的**原因**:
 - 1) 来自软件自身的特点: 是逻辑部件, 缺乏可见性; 规模庞大、复杂, 修改、维护困难。
 - 2) 软件开发与维护的方法不当: 忽视需求分析; 认为软件开发等于程序编写; 轻视软件维护。
 - 3) 供求矛盾将是一个永恒的主题: 面对日益增长的软件需求, 人们显得力不从心。
6. **消除**软件危机的**途径**:
 - 1) **消除**错误的概念和做法
 - 2) 推广使用**实践**中总结出来的成功技术和方法
 - 3) 开发和使用更好的**软件工具**
 - 4) 有必要的**组织管理措施**

1.2 软件工程

1. **什么是软件工程?**
1993 年 IEEE 的定义: 软件工程是: ① 把系统的、规范的、可度量的途径应用于软件开发、运行和维护过程, 也就是把**工程应用于软件**; ② **研究**①中提到的**途径**。
2. 软件工程的基本原理:
 - 1) 用分阶段的生命周期计划严格管理
 - 2) 坚持进行阶段评审
 - 3) 实行严格的产品控制
 - 4) 采用现代程序设计技术
 - 5) 结果应能清楚地审查
 - 6) 开发小组的人员应该少而精
 - 7) 承认不断改进软件工程实践的必要性
3. 在软件生命周期全过程中使用的一整套**技术方法的集合**称为**方法学**(methodology), 也

称为范型(paradigm)。

4. 软件工程方法学包括 3 个**要素**：方法、工具和工程。
5. 软件工程方法学包括：**传统方法学、面向对象方法学**

1.3 软件生命周期

1. 软件的**生命周期**：

- 1) 软件定义：问题定义、可行性研究、需求分析
- 2) 软件开发：总体设计、详细设计、编码和单元测试、综合测试
- 3) 运行维护：软件维护

1.4 软件过程

1. 软件过程：软件过程是为了**获得高质量软件所需要完成的一系列任务的框架**，它规定了完成各项任务的工作步骤。
2. 软件过程模型（软件生命周期模型）：软件过程模型是软件开发全部**过程、活动和任务**的结构框架。
3. 常见模型：瀑布模型、快速原型模型、增量模型、螺旋模型、喷泉模型、RUP、敏捷过程与极限编程
4. 比较**瀑布模型、快速原型模型、增量模型和螺旋模型**的优缺点，说明每种模型的使用范围：
 - 1) 瀑布模型
 - a) 优点：它**提供了一个模板**，这个模板使得分析、设计、编码、测试和支持的方法可以在该模板下有一个**共同的指导**。虽然有不少缺陷，但比在软件开发中随意的状态要好得多。
 - b) 缺点：(1) 实际的项目**大部分情况难以按照该模型给出的顺序进行**，而且这种模型的迭代是间接的，这很容易由微小的变化而造成大的混乱。(2) 经常情况下客户难以表达真正的需求，而这种模型却要求如此，这种模型是**不欢迎具有二义性问题存在的**。(3) 客户要等到**开发周期的晚期才能看到程序运行的测试版本**，而在这时发现大的错误时，可能引起客户的惊慌，而后果也可能是灾难性的。
 - 2) 快速原型模型
 - a) 优点：使用户能够感受到实际的系统，使开发者能够**快速地构造出系统的框架**。
 - b) 缺点：产品的先天性不足，因为开发者常常需要做**实现上的折中**，可能采用**不合适的操作系统或程序设计语言**，以使原型能够尽快工作。
 - 3) 增量模型
 - a) 优点：(1) **人员分配灵活**，刚开始不用投入大量人力资源，当核心产品很受欢迎时，可增加人力实现下一个增量。(2) 当配备的人员不能在设定的期限内完成产品时，它提供了一种先推出核心产品的途径，这样就可以**先发布部分功能给客户**，对客户起到镇静剂的作用。
 - b) 缺点：(1) 至始至终**开发者和客户纠缠**在一起，直到完全版本出来。(2) 适合于**软件需求不明确、设计方案有一定风险**的软件项目。该模型具有一定的市场。
 - 4) 螺旋模型
 - a) 优点：对于**大型系统及软件的开发**，这种模型是一个很好的方法。开发者和客户能够较好地**对待和理解每一个演化级别上的风险**。
 - b) 缺点：(1) 需要**相当的风险分析评估的专门技术**，且成功依赖于这种技术。(2)

很明显一个大的没有被发现的风险问题，将会导致问题的发生，可能导致演化的方法失去控制。(3) 这种模型相对比较新，应用不广泛，其功效需要进一步的验证。**该模型适合于大型软件的开发**

5. 为什么说喷泉模型较好的体现了**面向对象软件开发过程无缝和迭代**的特性？

使用面向对象方法学开发软件时，各个阶段都**使用统一的概念和表示符号**，因此，整个开发过程都是吻合一致的，或者说是无缝连接的，这自然就很容易实现**各个开发步骤的反复多次迭代，达到认识的逐步深化**，而**喷泉模型则很好的体现了面向对象软件开发过程迭代和无缝的特性**。

6. Rational 统一过程主要适用于：**大型的需求不断变化的复杂软件系统项目**。
7. 敏捷过程的适用范围：**商业竞争环境下对小型项目提出的有限资源和有限开发时间的约束**。
8. 微软过程的适用范围：**商业环境下具有有限资源和有限开发时间约束的项目的软件过程模式**。

1.5 补充

- 1) 我们拥有一套讲述如何开发软件的书籍，书中充满了标准与示例，可以帮助我们解决软件开发中遇到的任何问题。

答：好的参考书无疑能指导我们的工作，充分利用书籍中的方法、技术和技巧，可以有效地解决软件开发中大量常见的问题。但**实践者并不能依赖于书籍**，因为在现实工作中，由于条件千差万别，即使是**相当成熟的软件工程规范，常常也无法套用**。另外，软件技术日新月异，**没有哪一种软件标准能长盛不衰**。

- 2) 如果我们已经落后于计划，可以增加更多的程序员来赶上进度。

答：软件开发不同于传统的机械制造，**人多不见得力量大**。如果给落后于计划的项目增添新人，可能会**更加延误项目**。因为新人会产生很多新的错误，使项目混乱，并且原有的开发人员向新人解释工作和交流思想都要**花费时间**，使实际的开发时间更少，所以制定恰如其分的项目计划是很重要的。

- 3) 项目需求总是在不断变化，但这些变化能够很容易地满足，因为软件是灵活的。

答：软件需求确实是经常变化的，但这些**变化产生的影响会随着其引入时间的不同而不同**。对需求把握得越准确，软件的修修补补就越少。有些需求在一开始时很难确定，在开发过程中要不断地加以改正。**软件修改越早代价越少，修改越晚代价越大**，就跟治病一样道理。

- 4) 有了对目标的一般描述就足以开始写程序了，我们以后可以再补充细节。

答：**不完善的系统定义是软件项目失败的主要原因**。关于待开发软件的应用领域、功能、性能、接口、设计约束和标准等需要详细的描述，而这些只有通过用户和开发人员之间的通信交流才能确定。越早开始写程序，就要花越长时间才能完成它。

- 5) 一旦我们写出了程序并使其正常运行，我们的工作就结束了。人们有时认为，只有差的软件产品才需要维护。

答：软件投入的 50%~70%是花费在交付给用户之后。品质差的产品被丢弃，只有**好的产品才需要维护和改进**。

6) 一个成功的项目唯一应该提交的就是运行程序。

答：软件包括程序、数据和文档，其中文档是成功开发的基础，为软件维护提供了指导。

第 2 章 可行性研究

2.1 可行性研究的任务

1. 可行性研究的目的：用最小的代价，在尽可能短的时间内确定问题是否能够解决。

2.3 系统流程图

1. 定义：系统流程图实际上是物理数据流图，它描述组成系统的主要物理元素以及信息在这些元素间流动和处理的情况。

2.4 数据流图

1. 定义：描绘信息流和数据从输入移动到输出的过程中所经受的变换，反映了数据在软件中流动和被处理的逻辑过程，是描绘系统逻辑模型的极好工具。

2.5 数据字典

1. 定义：是关于数据的信息集合，是对数据流图中包含的所有元素定义的集合。