

计算机引论

山东建筑大学
计算机学院
秦松

10

Chapter 10

软件工程

从软件危机谈起

软件危机指的是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。

软件危机包含两方面问题：

- 如何开发软件，以满足不断增长，日趋复杂的需求
- 如何维护数量不断膨胀的软件产品

软件危机的原因：

- 一方面是与软件本身的特点有关，存在着复杂性
- 一方面是由软件开发和维护的方法不正确有关

从软件危机谈起

软件危机的主要表现：

1. 对软件开发成本和进度的估计常常很不准确。
2. 用户对“已完成”的软件系统不满意的现像经常发生。
3. 软件产品的质量常常靠不住。
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比例逐年上升。

软件工程的产生

1968年在NATO（北大西洋公约组织）会议上首次提出了软件工程的概念。

软件工程(Software Engineering, 简称为SE)是一门研究用工程化方法构建和维护有效的、实用的和高质量的软件的学科。它涉及到程序设计语言，数据库，软件开发工具，系统平台，标准，设计模式等方面。软件工程是一类工程。工程是将理论和知识应用于实践的科学。其中应用了计算机科学、数学和管理科学。计算机科学和数学用于构造模型与算法，工程科学用于制定规范、设计范型、评估成本及确定权衡，管理科学用于计划、资源、质量和成本的管理。

软件工程

软件工程学研究的是：如何应用一些科学理论和工程上的技术来指导大型软件系统的开发。使其发展成一门严格的工程科学；

软件工程学的最终目的是：以较低的成本研制具有较高质量的软件。

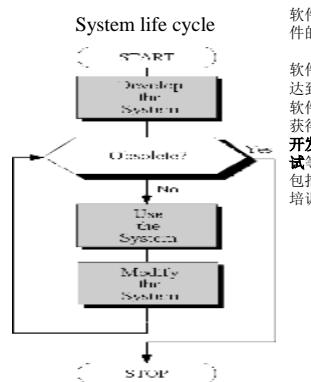
软件工程

软件工程(SoftWare Engineering)的框架可概括为：目标、过程和原则。

- (1) 软件工程目标：生产具有正确性、可用性以及开销合宜的产品。
- (2) 软件工程过程：生产一个最终能满足需求且达到工程目标的软件产品所需要的步骤。
- (3) 软件工程的原则是指围绕工程设计、工程支持以及工程管理在软件开发过程中必须遵循的原则。

软件生命周期

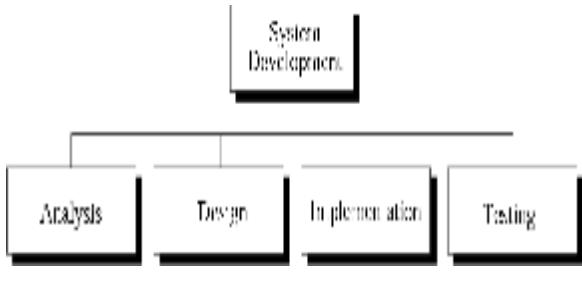
软件生命周期



软件生命周期(**SDLC**, 软件生存周期)是软件的产生直到报废的生命周期

软件工程活动是“生产一个最终满足需求且达到工程目标的软件产品所需要的步骤”。软件工程是利用合理的工程方法和原则来获得在真实机器上工作的可靠软件。**开发过程**主要包括**分析、设计、实现、测试**等活动。伴随以上活动，还有**支持活动**包括修改和完善、管理过程、支持过程、培训过程等。

软件开发周期



分析阶段

Analysis: 分析阶段包括可行性分析、问题分析和需求分析。

本阶段有四个步骤：定义用户、定义要求、定义需求、定义方法

定义用户：一般用户（通用）、特殊用户（专制）

定义要求：用户对软件的期望

定义需求：在用户需求的基础之上，定义系统的需求

定义方法：选择适当的方法满足需求

设计阶段

Design: 设计阶段

系统设计是把需求转化为软件系统的最重要的环节。系统设计的优劣从根本上决定了软件系统的质量。本阶段，主要根据需求分析的结果，对整个软件系统进行设计，如系统框架设计，数据库设计等等设计活动一般包括概要设计和详细设计。概要设计建立整个软件体系结构，包括子系统、模块以及相关层次的说明、每一模块接口定义。详细设计产生程序员可用的模块说明，包括每一模块中数据结构说明及加工描述。

完成文件、数据库设计。

1、模块化：整个程序分成若干模块，每个模块通过主程序相互连接。

2、工具：常用的是结构图，显示如何将软件分解成逻辑步骤，每个步骤就成为一个模块，同时显示各个部分之间的相互作用

实现阶段

Implementation: 实现阶段

此阶段是将软件设计的结果转换成计算机可运行的程序代码。在程序编码中必须要制定统一，符合标准的编写规范。以保证程序的可读性，易维护性，提高程序的运行效率。

- 1、工具：显示程序的逻辑流程，常用流程图、伪代码，阐明精确的算法细节，容易转换为计算机程序。
- 2、编码：选择合适的计算机语言具体实现

测试阶段

Testing: 测试阶段

编程大师说：“任何一个程序，无论它多么小，总存在着错误。”

测试活动贯穿于整个开发过程，实现完成后的测试，保证最终产品满足用户的要求。

错误是一种严重的程序缺陷。测试的目的是为了发现尽可能多的缺陷，并期望通过改错来把缺陷统统消灭，以期提高软件的质量。

1、黑盒测试：指看不见程序的内部结构，即按照规格说明来测试程序是否符合要求，“黑盒”测试方法只需测试模块的接口是否符合要求，它关心程序的外部表现而不是内部的实现细节。

2、白盒测试：必须由开发者自己执行，要对模块的所有代码进行单步跟踪测试。

经验。编程时注意积累测试用例。

软件工程师

软件工程师是对应用软件创造软件的人们的统称，软件工程师按照所处的领域不同可以分为系统分析员，软件设计师，系统架构师，程序员，测试员等等。人们也常常用程序员来泛指各种软件工程师。

质量

质量的定义

优质软件的定义：软件能够满足用户显式或隐式的需求，文档齐全，符合组织的操作标准，在其开发使用的硬件上高效运行。

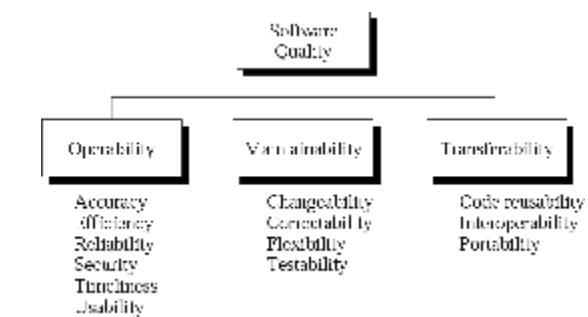
软件质量评价标准

1. 质量需求准则：着眼点是否满足用户的要求
2. 质量设计准则：开发者在设计实现时是否按软件需求保证了质量
3. 质量度量准则：为质量度量规定了一些检查项目：
 精密度量：根据质量度量准则进行详细度量
 全面度量
 简易度量

度量：如每千行的bug数，平均故障时间，还有一些其他因素

质量因素

质量因素：可操作性、可维护性、可迁移性



质量因素

可操作性涉及基本操作：它是怎样驱动的？
用户对系统的第一印象在于该系统看上去和感觉上怎样？
对在线交互式系统，使用起来是否容易直观？
是否很好的是和运行所依赖的操作系统？
全部是主观因素，不可度量。
包含其他几个因素：
准确性：没错
效率：运算速度快
可靠性：系统稳定
安全性：未授权的人不能取得数据
及时：响应用户速度快
适用性：用户使用是否方便

质量因素

维护性保证系统正常运行、升级。
包含几个因素：
可变性：随时间，需求改变
可修正性：当程序出现故障后程序恢复运行所花费的时间。
灵活性：系统可灵活多变
可测试性：
可移植性将数据或程序从一个平台移到另一个平台的能力，通用性
包含几个因素：
代码可重用性：编写出来的函数可在不同的项目中重用
互操作性：由发送数据给其它系统的能力
可移植性：将数据或程序从一个平台移到另一个平台的能力

文档

文档

软件的正确使用和有效维护离不开文档。

文档分用户文档和系统文档。

用户文档：使用手册
系统文档：分析阶段文档
设计阶段文档
实现阶段文档
通用文档
功能文档
测试文档

进行过程中的文档化：文档化是进行中的过程，记录系统改动