

软件工程 Software Engineering



中南大学计算机学院
主讲:黄芳 教授
2021年9月

教学安排与特色

1. 教材
[1] 沈备军主编. 软件工程原理. 北京:高等教育出版社. 2013年;
2. 教学特色: 软件逆向工程项目实践与课程教学同步进行
源码阅读>软件再造>原型与报告(分组进行)
采用分阶段迭代式完成项目开发, 提交实践报告。
工具: VISIO、UML、Trustie平台
3. 共48学时
4. 考核方式
项目实践报告、课外阅读、作业与笔试相结合。

教材: 软件工程原理

高等教育出版社
ISBN: 978-7-04-036906-9



计算机软件已经成为一种驱动力

- ▶ 进行商业活动的引擎;
- ▶ 现代科学研究和工程问题解决的基础;
- ▶ 区分现代产品和服务的关键因素;
- ▶ 现代社会中不可缺少的。

应用于:

交通、医药、通讯、军事、娱乐、
办公、社交、学习.....

举例: Dell, Amazon, 淘宝, Ctrip等成功案例

“软件定义一切”

软件工程是一门交叉学科

软件工程的主要研究内容

软件开发技术: 软件开发方法学
软件开发过程
软件工具和软件工程环境

软件工程管理: 软件管理学
软件经济学
软件心理学

软件工程所包含的内容不是一成不变的,
随着人们对软件系统的研制开发和生产的理解。
应该用发展的眼光看待它。

软件工程知识体系SWEBOK

- ▶ 软件工程知识体系(Software Engineering Body of Knowledge) 简称SWEBOK
- ▶ 由IEEE国际组织推出
- ▶ 版本: 当前版本 V3(2014)
- ▶ 网址: <http://www.swebok.org>
- ▶ 国际软件工程师证书:
 - CSDA, 针对大学应届生
 - CSDP, 针对有经验的工程师

SWEBOK V3的15个知识域

- ▶ 软件工程实践类知识域
 - 软件需求、软件设计、软件构造、软件测试、软件维护、软件配置管理、软件工程管理、软件过程管理、软件工程方法、软件质量、软件工程职业实践
- ▶ 软件工程教育要求类知识域
 - 工程经济基础、计算基础、数学基础、工程基础

教学 内容 范围

- 软件危机与软件工程
- 软件过程
- 软件建模方法
- 需求工程
 - 构建分析模型
 - 结构化分析
 - 面向对象的分析
- 设计工程
 - 结构化设计
 - 面向对象的设计
- 人机界面设计
- 软件测试
- 软件调试和维护
- 软件工程管理
 - 软件质量管理
 - 软件项目计划
- UML与VISIO初步
- 项目实践



中国软件产业现状

- ▶ 主权大国必须建立基于自主技术的、完整的软件产业体系；
- ▶ 世界软件市场的份额分配大致为：美国60%，西欧20%，日本12%，其它8%；
- ▶ 软件本国提供率：中国11%左右，美国97%；
- ▶ “印度模式”还是“中国模式”；
- ▶ 软件人才结构不合理，缺乏中高级软件人才；软件人员缺乏软件工程化的概念。

基于开源和群智的软件开发方法突出重围！

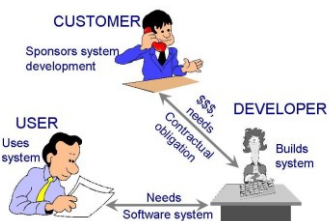
“软件工程”课程教学与实践的目标

- ▶ 转变对软件开发的认识：

程序 $\xrightarrow{\text{上升}}$ 系统
- ▶ 转变思维定式：

程序员 $\xrightarrow{\text{上升}}$ 系统工程师
(系统分析员)
- ▶ 工程化训练

- ▶ Participants (stakeholders) in a software development project



系统分析员的地位



“软件工程”课程 与其它软件专业课的区别

- (1) 立足于系统的整体。
- (2) 讲授系统分析、系统设计、测试及维护的理论和方法。
- (3) 构筑一个软件系统，实践软件开发全过程。

第一章 软件危机与软件工程 Software Crisis and Software Engineering

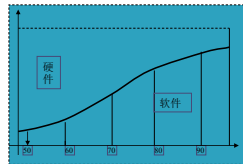
主要内容

- ▶ 1、计算机的发展与软件危机
The development of Computer and Software Crisis
- ▶ 2、软件工程
Software Engineering
- ▶ 3、技术审查和管理复审
Inspections and Reviews

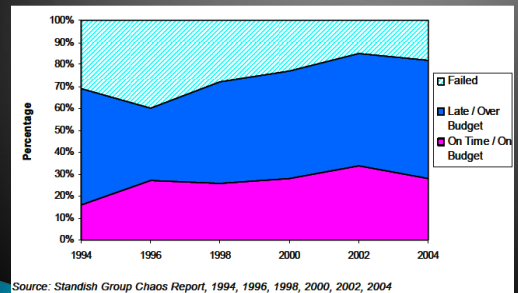
1、计算机的发展与软件危机

The Development of Computer and Software Crisis

- ▶ 计算机的发展
 - 初期硬件
 - 出现操作系统
 - 大量的应用软件
 - 供需矛盾
 - 软件危机



软件项目成功率调查



软件危机 Software Crisis

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。

概括来说，软件危机包含两方面问题：

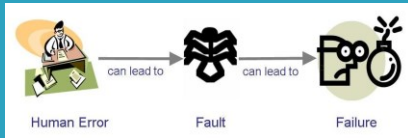
- 一、如何开发软件，以满足不断增长，日趋复杂的需求；
- 二、如何维护数量不断膨胀的软件产品。

软件危机的表现 Representations of Software Crisis

(1) 对软件开发成本和进度的估计常常不准确，软件生产进度无法控制。开发成本超出预算，实际进度比预定计划一再拖延的现象并不罕见。

(2) 软件需求定义不准确，易偏离用户需求。用户对“已完成”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。Bug一大堆，Patch一个接一个。



(4) 软件的可维护程度非常之低。很多程序的错误是很难改正的，不能根据用户的需求变化在原有程序中增加新的功能。

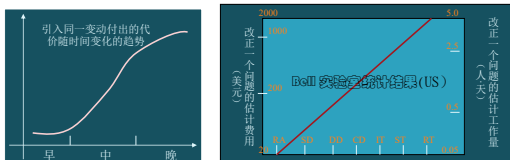
(5) 软件通常没有适当的文档资料。这些文档作为管理和评价软件开发工程的进展情况，作为信息交换工具；对于软件维护人员文档更是必不可少。

(6) 软件的成本不断提高。随着技术的进步硬件成本不断降低，而软件成本随着人力资源成本的增高、软件规模的增大而不断上涨。

软件危机的表现 Representations of Software Crisis

- (7) 软件开发生产率的提高赶不上硬件的发展和人们需求的增长。
- (8) 软件可维护性差。

- ▶ IRS (the United States Internal Revenue Service) hired Sperry Corporation to build an automated federal income tax form processing process
 - An extra \$90 M was needed to enhance the original \$103 M product
 - 工作量估计不足
 - IRS lost \$40.2 M on interests and \$22.3 M in overtime wages because refunds were not returned on time
 - 间接损失
- ▶ Malfunctioning code in Therac-25 killed several people
 - 箭头的非标准使用时的错误
- ▶ Reliability constraints have caused cancellation of many *safety critical* systems
 - *Safety-critical*: something whose failure poses a threat to life or health



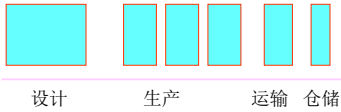
软件产品需要分阶段的定义、设计和开发，严格的产品质量控制，完整的文档记录。

软件危机产生的原因 Motivations of Software Crisis

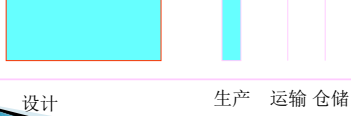
- 软件本身的特点
(Characteristics of Software)
- 对软件开发与维护存在许多错误认识和做法
(Incorrect Opinions for Software Development and Maintenance)
- 软件开发与维护的方法不正确
(Incorrect Method of Development and Maintenance)

软件是被设计的

制造业:



软件业:



软件的特性

Characteristics of Software

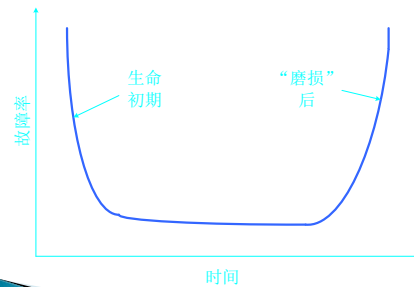
- 软件是一种逻辑实体，具有抽象性。
这个特点使它与其它工程对象有着明显的差异。人们可以把它记录在纸上、内存、和磁盘、光盘上，但却无法看到软件本身的形态，必须通过观察、分析、思考、判断，才能了解它的功能、性能等特性。
- 软件没有明显的制造过程。
一旦研制开发成功，就可以大量拷贝同一内容的副本。所以对软件的质量控制，必须着重在软件开发方面下工夫。

软件的特性

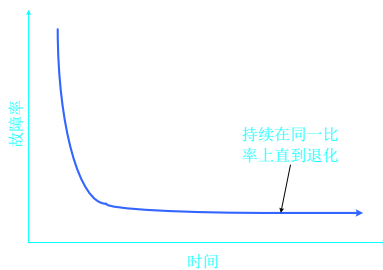
Characteristics of Software

- 软件在使用过程中，没有磨损、老化的问题。
软件在生存周期后期不会因为磨损而老化，但会为了适应硬件、环境以及需求的变化而进行修改，而这些修改有不可避免的引入错误，导致软件失效率升高，从而导致软件**退化**。当修改的成本变得难以接受时，软件就被抛弃。
- 软件对硬件和环境有着不同程度的依赖性。这导致了软件移植的问题。
- 软件的开发至今尚未完全摆脱手工作坊式的开发方式，生产效率低。

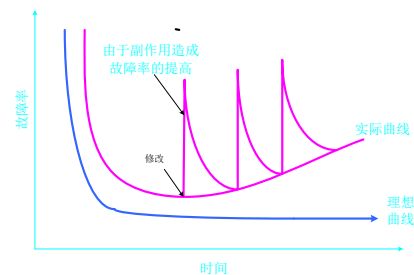
硬件的故障率曲线（浴缸曲线）



软件的故障率曲线(理想情况下)



软件的故障率曲线(实际情况下)



软件的特性 Characteristics of Software

■软件是复杂的，而且以后会更加复杂。

软件是人类有史以来生产的复杂度最高的工业产品。软件涉及人类社会的各行各业、方方面面，软件开发常常涉及其它领域的专门知识，这对软件工程师提出了很高的要求。

■软件的成本相当昂贵。软件开发需要投入大量、高强度的脑力劳动，成本非常高，风险也大。现在软件的开销已大大超过了硬件的开销。

■软件工作牵涉到很多社会因素。

许多软件的开发和运行涉及机构、体制和管理方式等问题，还会设计到人们的观念和心理。这些人的因素，常常成为软件开发的困难所在，直接影响到项目的成败。

软件危机产生的原因

- 对软件开发与维护存在许多错误认识和做法
(Incorrect Opinions for Software Development and Maintenance)

忽视软件需求分析的重要性；对软件与程序的概念不清；轻视软件维护。

- 软件开发与维护的方法不正确
(Incorrect Method of Software Development and Maintenance)

对系统需求没有清楚和准确的认识就进入开发阶段，忽视对软件开发过程的管理；……

软件面临的新挑战

- 软件复杂性的增加
- 软件规模的不断扩大
- 软件环境的变化
- 遗留系统（Legacy System）的集成和复用
- 软件开发的高质量和敏捷性要求
- 分散的开发团队的协同

1.2 软件工程(Software Engineering)

解决软件危机的途径
The method for resolving Software Crisis

软件工程（Software Engineering）采用工程的概念、原理、技术和方法来开发与维护软件的过程。

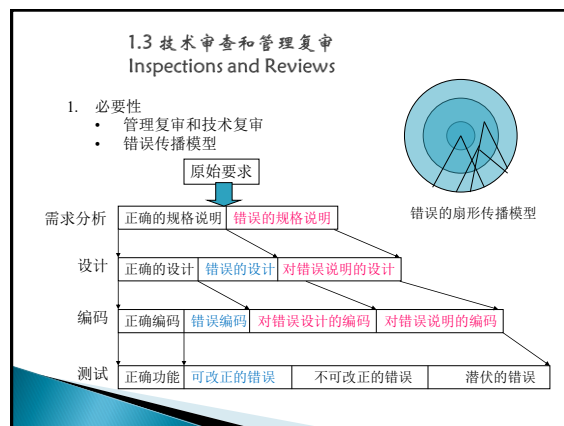
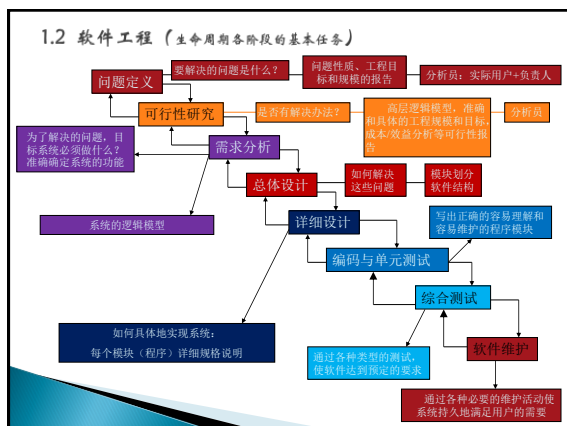
- 软件工程学是一门研究如何用系统化、规范化、数量化等工程原则和方法去进行软件的开发和维护的学科。

1、软件工程的本质特性

软件工程关注大型程序的构造；
软件工程的中心课题的控制复杂性；
软件经常变化；
软件开发效率很重要；
良好的合作是软件开发的关键；
软件必须有效地支持它的用户；
软件工程领域通常由一种专业人士替另一种专业人士创造产品。

2. 软件工程的传统途径—生命周期方法学(Software life cycle)

- 生命周期方法学的基本内容
 - 从时间角度对软件开发和维护的复杂问题进行分解，把软件生命的漫长周期依次划分为若干个阶段，然后逐步完成。
- 生命周期方法学的应用方法
 - 从对任务的抽象逻辑分析开始，一个阶段一个阶段地进行开发；前一个阶段任务的完成是后一个阶段工作的前提和基础，而后一个阶段任务通常是使前一阶段提出的解法更进一步的具体化。
- 阶段过渡方法
 - 每一个阶段的开始和结束都有严格标准，前一阶段结束的标准是后一阶段工作开始的标准。技术审查和管理复审。
- 基本概念
 - 软件的定义（Boehm）：软件是程序以及开发、使用和维护程序需要的所有文档。

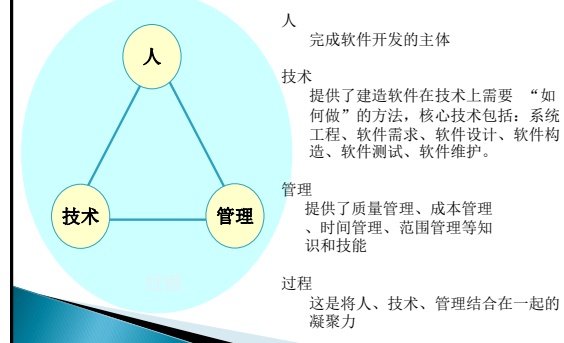


技术审查和管理复审

- 管理复审的基本任务：在软件生命周期的每个重要的里程碑（一般是每个阶段计划、需求分析、设计、编码、维护）对工程项目的成本、实际花费、投资回报的前景等从管理的角度进行审查。
- 技术审查的基本任务：在软件生命周期的每个阶段进行正式而严格的技术审查，尽量发现隐藏的错误。
- 正式技术评审是软件工程实践者实施的一项软件质量保证活动

评审办法：评审会议、评审记录、评审指南

1.4 软件工程的金三角



1.5 控制软件开发的复杂性

技术的复杂性、需求的复杂性、人的复杂性。

1、抽象

抽出共同的、本质的特征，包括两种：过程抽象与数据抽象。

2、分解

分而治之，把复杂问题分成若干简单的子问题。

3、迭代

反复、循环求解，当无法正确抽象和分解时采用迭代的方法，以积累经验。



图 1-3 抽象层次