绪论

徐本柱 软件学院 2018-09



101

▼软件工程内容

→如何学好软件工程



- ▶绪论 (chap 1&2)
- ▶第一部分 软件过程
- ▶第二部分 建模
- **▽**第三部分 质量管理
- ♥ 第四部分 软件项目管理





要点浏览

- 概念: 计算机软件是由专业人员开发并长期维护的软件产品
 - ❖ 可以在各种不同容量及系统结构的计算机上运行的程序、
 - * 程序运行过程中产生的各种结果以及各种描述信息,
 - ❖ 可以以硬拷贝或是各种电子媒介形式存在
- 人员: 软件工程师开发软件并提供技术支持,产业界中几乎 每个人都间接或直接地使用软件
- ▼ 重要性: 生活中无所不在、日渐深入到商业、文化等各方面
- ▶ 步骤:
 - ❖ 客户和利益相关者表达对计算机软件的要求,
 - 工程师构建软件产品,
 - ❖ 最终用户应用软件来解决特定的问题或者满足特定的要求
- ▼ 工作产品和质量保证措施



引言

- ▼软件工程应用
 - *传统模式(购买、拥有、使用和管理软件)
- ♥网络交付
 - *软件可能运行在遥远的服务器上
 - *看起来就像驻留在每个用户的计算设备上运行一样
- ▼软件部署
 - ❖不必为成千上万个用户交叉访问同一应用系统的5个不同版本担心
- ♥软件维护
 - *仅在服务器上运行当前最新版本的软件
 - *每当对软件做了更新或改正,每个用户都可以立刻拥有最新的版本
- ♥需要软件工程
 - ❖快速、正确或廉价(3选2)→快速、正确



*某些新科技的发明创造

101101

- *给其他一些看似无关的技术领域、商业企业、 公众甚至整个社会文化带来深远而出人意料的影响和作用
- ❖这就是所谓的"意外效应法则"。



1.1软件的本质

- *软件技术具有双重作用
 - *产品和产品交付载体
 - ❖作为产品,显示了由计算机硬件体现的计算能力
 - ❖作为产品生产的载体,软件提供了计算机控制、信息通信以及应用程序开发和控制的基础平台
- ❖软件提供了最重要的产品——信息。
- *软件产业已经成为工业经济中的主导因素



面临的问题

- *软件生产过程中面临的问题:
 - ❖为什么软件需要如此长的开发时间?
 - ❖为什么开发成本居高不下?
 - ❖为什么在将软件交付顾客之前,我们无法找到所有的错误?
 - ❖为什么维护已有的程序要花费高昂的时间和 人力代价?
 - ❖为什么软件开发和维护的过程难以度量?



1.1.1 定义软件

❖软件是:

- ❖(1)指令的集合(计算机程序),通过执行这些 指令可以满足预期的特征、功能和性能需求;
- *(2)数据结构,使得程序可以合理利用信息;
- *(3)软件描述信息,它以硬拷贝和虚拟形式存在 田来描述程序操作和使用
 - ,用来描述程序操作和使用。

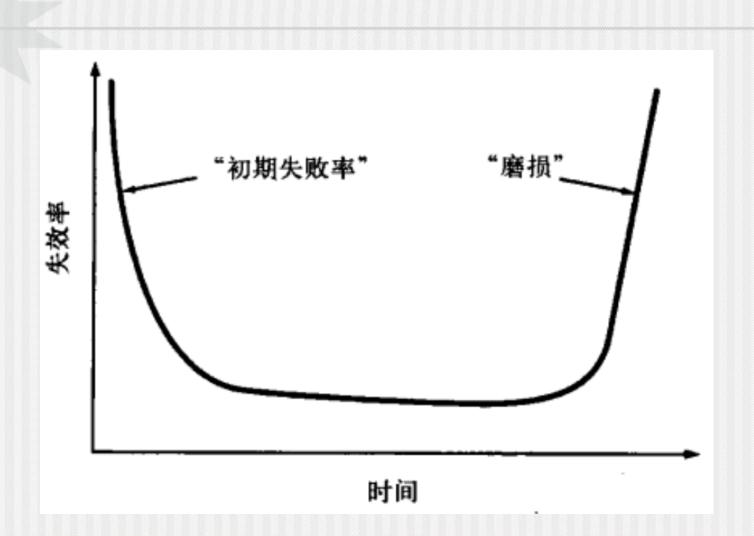


软件的特性

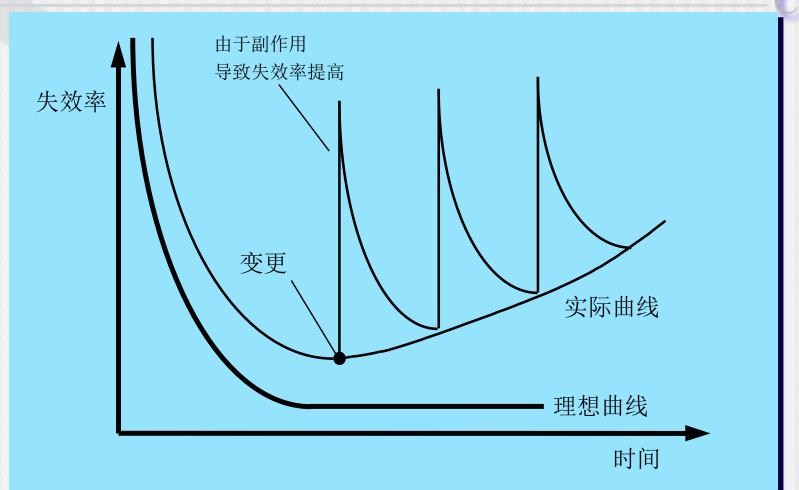
- *1软件是逻辑的而非物理的系统元素
 - •是设计开发的,而非传统意义上生产制造的
- *2 软件不会"磨损"
- *3虽然整个工业向着基于构件的构造模式发展,然而大多数软件仍是根据实际的顾客需求制定的



硬件失效曲线图



软件的特性



1.1.2软件应用领域

- *计算机软件可分为七个大类:
 - *系统软件
 - *应用软件
 - ❖工程/科学软件
 - *嵌入式软件
 - *产品线软件
 - ❖Web应用软件
 - ◆人工智能软件
 - *(包括机器人、人工神经网络和博弈等)



演化的原因?

101101

- ❖软件需要进行适应性调整,从而可以满足新的计算环境或者技术的需求。
- * 软件必须升级以实现新的商业需求。
- ❖ 软件必须被扩展使之具有与更多新的系 统和数据库的互操作能力。
- ❖软件架构必须进行改建使之能适应不断 演化的计算环境。



1.2 软件的变更本质

- WebApp
- ♥移动App
- ▼云计算
- → 产品线软件





软件危机

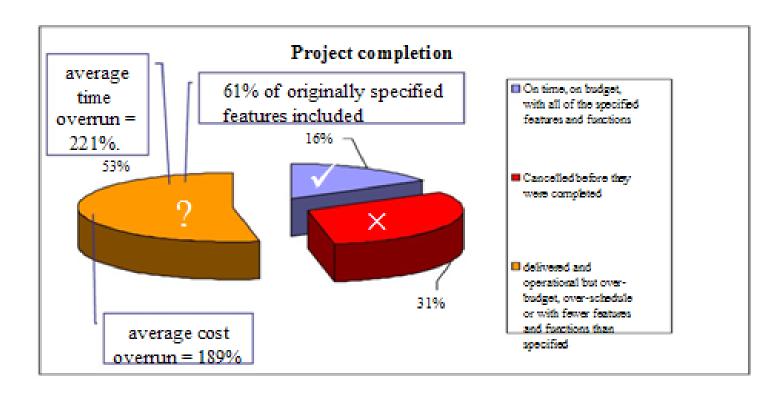
- ❖软件危机(Software Crisis)
 - *软件的开发和维护过程所遇到的一系列严重问题
- * 软件危机的表现:
 - ❖对软件开发成本和进度的估算很不准确,甚至严重 拖期和超出预算;
 - ❖无法满足用户需求,导致用户很不满意;
 - ❖质量很不可靠,经常失效;
 - ❖难以更改、调试和增强;
 - ❖没有适当的文档;
 - ❖软件成本比重上升;
 - * 软件开发生产率跟不上计算机应用迅速深入的趋势



统计数字:混乱的软件开发

Standish Group – 2004 调 查对 象:来 自美 国不同 行业 的 **365个公司的CIO**;**8,380个不同的** 项目:

调查的结果
•预算: 189%
•时间: 221%
•功能: 61%





- ❖客观上: 软件产品开发的复杂度和难度随软件规模呈指数级别增长
- *随着软件规模的急速增长,软件的开发方法已经不适用
- *主观上:软件开发人员缺乏工程性的、系统性的方法论
- ❖程序员具有编程的能力,但对软件开发这一过程性较强的任务却缺乏足够的工程化思维;
- ❖对软件开发的一些认识的误区:软件神话(Software myths);
- ❖没有将"软件产品研发"与"程序编码"区分清楚;
- ❖忽视需求分析、轻视软件维护。



几个简单事实

- →确定软件方案之前,需要共同努力来理解问题
- ♥设计已成为关键活动
- ♥软件应该具有高质量
- ♥软件需具备可维护性
- 掌结论:各种形式、各个应用领域的软件都需要工程化



2.1 定义软件工程学科

- ▼【软件工程是】建立和使用一套合理的工程 原则,以便经济地获得可靠的、可以在实际机 器上运行的软件。
- ♥软件工程: (IEEE)
 - ❖ (1) 将系统化的、严格约束的、可量化的 方法应用于软件的开发、运行和维护,即将 工程化应用于软件
 - ❖ (2) 在(1) 中所定义的过程中的方法研究



- > 软件工程是一种层次化的技术
 - ❖ 任何工程方法必须构建在质量承诺的基础上(根基质量关注点)
- ▶ 软件工程的基础是过程层
 - * 软件过程将各个技术层次结合在一起,使得合理、及时地开发计算机软件成为可能。
- ▶ 软件工程方法为构建软件提供技术上的解决方法
 - ❖ 如何做: 沟通、需求分析、设计建模、编程、测试等
- ▼ 软件工程工具为过程和方法提供自动化或半自动化的支持





2.2 软件过程

- ❖软件过程是工作产品构建时所执行的一系列<mark>活</mark> 动、动作和任务的集合。
- *活动主要实现宽泛的目标
 - ❖与应用领域、项目大小、结果复杂性或者实施软件 工程的重要程度没有直接关系
- ❖动作包含了主要工作产品(如体系结构设计模型)生产过程中的一系列任务
- ❖任务关注小而明确的目标,能够产生实际产品(如构件一个单元测试)



2.2.1 过程框架

过程框架 框架活动

工作任务

工作产品

里程碑&可交付成果

QA 检查点

普适性活动



通用框架活动

- *沟通
- * 策划
- * 建模
 - * 需求分析
 - * 设计
- * 构建
 - * 编码
 - * 测试
- ❖ 部署



2.2.2 普适性活动

- *软件项目跟踪和控制
- *风险管理
- *软件质量保证
- *技术评审
- ❖测量
- *软件配置管理
- ❖可复用管理
- *工作产品的准备和生产



- 活动、动作和任务的总体流程,以及相互依赖关系
- ▶ 在每一个框架活动中,动作和任务细化的程度
- ▶ 工作产品的定义和要求的程度
- ▶ 质量保证活动的应用方式
- 项目跟踪和控制活动应用的方式
- ♥ 过程描述的详细程度和严谨程度
- ▼ 客户和利益相关者对项目参与的程度
- ♥ 软件团队所赋予的自主权
- ▶ 队伍组织和角色的明确程度



2.3 软件工程实践

*实践的精髓

- 1. 理解问题(沟通和分析)
- 2. 计划解决方案(建模和软件设计)
- 3. 实施计划(代码生成)
- 4. 检查结果的正确性(测试和质量保证)



- ◆第1原则:存在价值
- ❖第2原则:保持简洁
- ❖第3原则:保持愿景
- ❖第4原则: 关注使用者
- ❖第5原则:面向未来
- ❖第6原则: 计划复用
- ❖第7原则: 认真思考



2.4 软件神话

- 经理、客户(和其他非技术利益相关者) 和实践者的影响力很大
- 他们通常有一套看似正确的方法 但是...
- 大量软件神话将他们诱入歧途 因此...
- 随着对软件工程理解的深化,无论何时 遇到这些神话,都要不遗余力地揭露



■客户:

- 软件改起来简单,你们先做着,做好了我看看,不合适再改。
- 我就需要你们做一个类似Excel的功能 ,但是要符合我的要求。



软件神话的几个例子

■开发者:

- 客户要什么,我们就做什么;
- 当我完成程序并将其交付使用之后,我的任务便结束了;
- 软件还没开发完呢,好不好用等做完再说;
- •程序好使就行,其他无所谓;
- 软件工程将导致我们产生大量的无用文档, 并因此降低工作效率;



软件神话的几个例子

- ■软件公司的管理者:
 - 我们制定了完善的规范和标准,只要照章执行,软件开发就不会出问题;
 - 软件项目拖期了,再招聘几个人,很 快就能赶上进度;
 - 只要我们使用了最先迚的软件开发语言、开发工具和开发环境,软件项目将会一帆风顺。



2.5 如何开始

SafeHome:

- ■每个软件工程项目都来自业务需求
 - 对现有应用程序缺陷的修正;
 - 改变遗留系统以适应新的业务环境;
 - 扩展现有应用程序功能和特性;
 - 或者开发某种新的产品、服务或系统。



潮扩潮!



101 101101