



中山大學  
SUN YAT-SEN UNIVERSITY

## Chapter 1

# Overview of Software Engineering

**Software Testing: Approaches & Technologies**

School of Data & Computer Science, Sun Yat-sen University

# Outline

---

- 1.1. 软件与软件危机
- 1.2. 软件开发
- 1.3. 软件生命周期模型
- 1.4. 软件质量模型
- 1.5. 敏捷开发



# 1.1 软件与软件危机

---

## 1.1.1 软件

- 计算机体系结构
  - 计算机硬件体系结构
    - ◇ 冯·诺依曼结构
    - ◇ 基本属性
      - 机内数据表示
      - 寻址方式
      - 寄存器组织
      - 指令系统
      - 存储系统
      - 中断机构
      - 输入输出结构
      - 信息保护

# 1.1 软件与软件危机

---

## 1.1.1 软件

- 计算机体系结构

- 计算机软件体系结构

- ✧ 软件体系结构是软件系统的结构、行为和属性的高级抽象，描述构成系统的元素、元素的相互作用、元素集成的模式以及这些模式的约束。软件体系结构给出系统的组织结构和拓扑结构，规定系统需求和构成系统的元素之间的对应关系，并提供设计决策的基本原理。
    - ✧ 描述软件体系结构的模型分类：结构模型，框架模型，动态模型，过程模型，功能模型等。

- 软件 = 程序 + 数据 + 文档 (+ 服务)

- ✧ 能够完成预定功能和性能的、可执行的计算机指令
  - ✧ 使得程序能够适当地操作信息的数据结构
  - ✧ 描述程序的操作和使用的文档

# 1.1 软件与软件危机

---

## 1.1.1 软件

- 软件是逻辑实体 (抽象性), 不是物理实体
  - 软件是逻辑的、知识性的产品集合, 是对物理世界的一种抽象, 或者是某种物理形态的虚拟化
  - 软件是智力成果, 开发成本昂贵, 可以复制 (低边际成本)
  - 软件的问题都是在软件开发和修改过程中引入的; 软件的开发至今尚未完全摆脱手工作坊式的低效率的开发方式
  - 软件的复杂程度持续上升
  - 软件工程与社会工程的交叉

# 1.1 软件与软件危机

---

## 1.1.1 软件

- 软件的分类
  - 系统软件
    - ✧ 操作系统、数据库管理系统、设备驱动程序、通信处理程序等
  - 应用软件
    - ✧ 事务软件、实时软件、工程和科学软件、嵌入式软件、娱乐软件、个人计算机软件、人工智能软件等
  - 工具软件
    - ✧ 文本编辑软件、文件格式化软件、数据传输软件、程序库系统以及支持需求分析、设计、实现、测试和支持管理的软件等
  - 可重用软件

# 1.1 软件与软件危机

---

## 1.1.1 软件

- 软件产品的组成
  - 开发周期
    - ✧ 客户需求文档 - Customer Requirements
    - ✧ 市场需求文档 - MRD (Marketing Requirement Documents)
    - ✧ 软件规格说明书 – Specifications
    - ✧ 技术设计文档 – Technical Design Documents
    - ✧ 测试文档 Test Documents
    - ✧ 在线帮助 - Online Help
    - ✧ 产品发布注释 - Release Notes / Read Me
    - ✧ 产品软件包 - Release Packages

# 1.1 软件与软件危机

---

## 1.1.1 软件

- 软件产品的组成 (续)
  - 销售周期
    - ✧ 帮助文件 Help Files
    - ✧ 示例 Samples and Examples to Illustrate Points
    - ✧ 产品支持文档 Product Support Information
    - ✧ 错误信息 Error Messages
    - ✧ 安装手册 Setup and Installation Instructions
    - ✧ 用户手册 User Manual (s)
    - ✧ 产品标签 Label and Stickers
    - ✧ 产品广告或宣传材料 Ads and Marketing Material ...



# 1.1 软件与软件危机

---

## 1.1.2 软件危机

- 早期软件开发的特点
  - 软件规模相对较小
  - 程序设计是一门技艺
  - 缺少软件开发工具的支持
  - 缺乏软件开发管理的理论与方法
  - 缺乏有效的软件开发后的维护

# 1.1 软件与软件危机

---

## 1.1.2 软件危机

- 软件危机的表现形式
  - 软件开发成本日益增长
    - ✧ 开发成本超出预算
  - 软件开发进度难以控制
    - ✧ 实际进度比预定计划一再拖延
  - 用户对“已完成”系统不满意的现象经常发生
    - ✧ 用户需求不明确
  - 软件产品的质量不可靠
    - ✧ Bug & Patch
  - 软件的可维护程度低
    - ✧ 数量不断膨胀的软件产品缺乏适当的文档资料
  - 软件开发生产率跟不上硬件的发展和人们需求的增长

# 1.1 软件与软件危机

---

## 1.1.2 软件危机

- 软件危机的原因
  - 软件危机的根源
    - ✧ 软件的大量需求与软件生产力效率之间的矛盾
    - ✧ 软件系统的复杂性与软件开发方法之间的矛盾
  - 软件本身的特点
    - ✧ 软件是一种抽象逻辑
    - ✧ 软件是开发人员的智力劳动成果
    - ✧ 软件具备强烈的个性化特征
    - ✧ 软件规模日趋庞大，实现的业务逻辑与流程复杂

# 1.1 软件与软件危机

---

## 1.1.2 软件危机

- 软件危机的原因 (续)
  - 软件开发的客观因素
    - ✧ 系统需求分析不足
    - ✧ 开发周期管理不善
    - ✧ 开发过程缺乏规范
      - 软件开发 =? 程序编写
    - ✧ 质量控制标准规程滞后
    - ✧ 软件维护计划被忽视
  - 产业因素
    - ✧ 软件企业的作坊式管理
    - ✧ 软件企业规模的急剧膨胀

# 1.1 软件与软件危机

---

## 1.1.2 软件危机

- 软件危机的消除

- 解决途径

- ✧ 正确认识计算机软件的内涵 (软件≠程序)
    - ✧ 充分认识到软件开发不是某种个体劳动的神秘技巧，而是一种组织良好、管理严密、协同配合的工程活动
    - ✧ 采用成熟的软件开发技术和方法
    - ✧ 开发和使用适当的软件工具

- 实施办法

- ✧ 组织管理
      - 采用工程项目管理方法
    - ✧ 技术措施
      - 采用软件开发技术、方法与软件工具

## 1.2 软件开发

---

### 1.2.1 软件开发的基本过程

- 软件开发包括计划、需求分析、设计、实现、测试和维护的基本过程
  - 计划
  - 需求分析
    - ✧ 根据客户的要求，清楚了解客户需求中的产品功能、特性、性能、界面和具体规格等，然后进行分析，确定软件产品所能达到的目标。
  - 设计
    - ✧ 根据需求分析的结果，考虑如何在逻辑、程序上去实现所定义的产品功能、特性等，可以分为概要设计和详细设计，也可分为数据结构设计、软件体系结构设计、应用接口设计、模块设计、界面设计等。

## 1.2 软件开发

---

### 1.2.1 软件开发的基本过程

- 软件开发包括计划、需求分析、设计、实现、测试和维护的基本过程
  - 实现
    - ✧ 编写程序将设计转换成计算机可读的形式。
  - 测试
    - ✧ 确认用户需求、对设计和实现结果进行验证的过程。
  - 维护
    - ✧ 维持软件运行，修改软件缺陷、增强已有功能、增加新功能、升级等。

## 1.2 软件开发

### 1.2.2 软件开发成本

- 软件开发成本分布

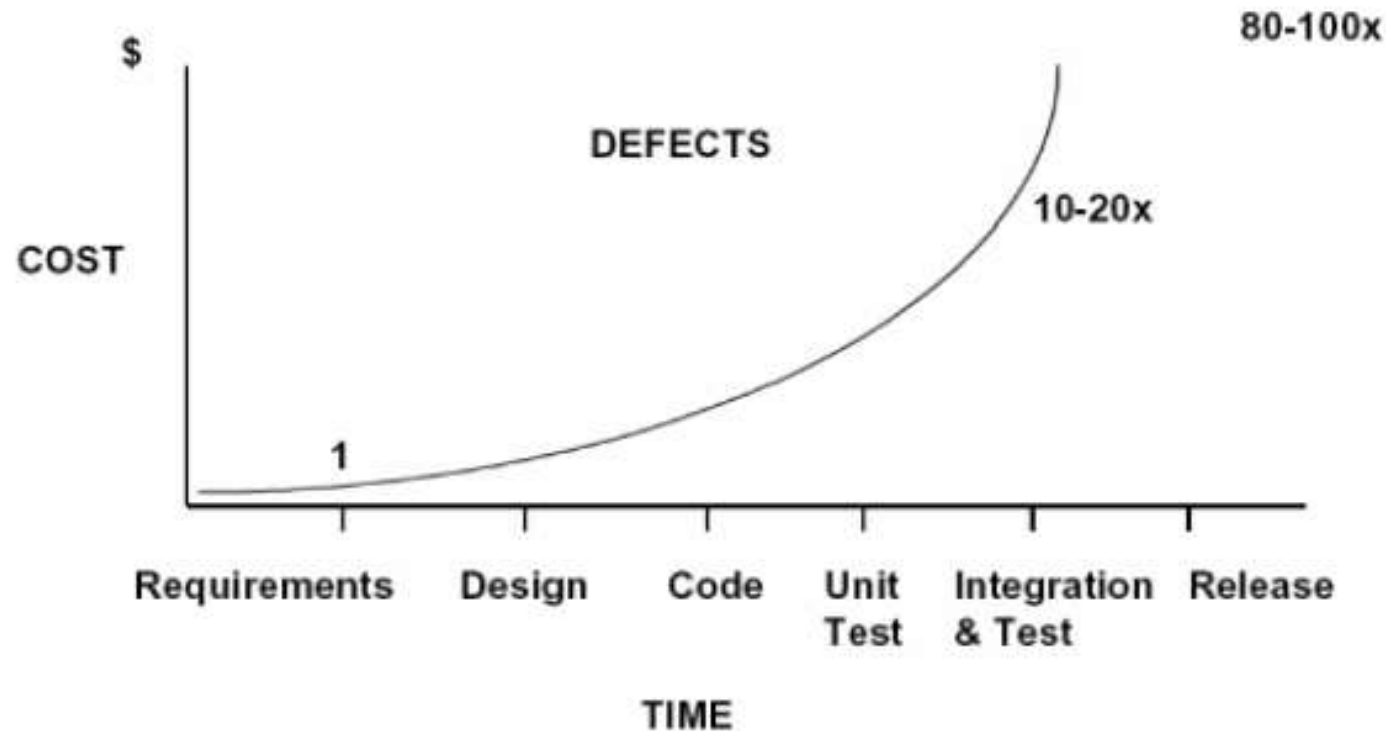
软件类型	开发成本的阶段分布 %		
	需求与设计	实现	测试
控制软件	46	20	34
航空航天软件	34	20	46
操作系统	33	17	50
科技计算软件	44	26	30
商业应用软件	44	28	28



## 1.2 软件开发

### 1.2.2 软件开发成本

- 软件的修改成本随时间变化的趋势



## 1.2 软件开发

---

### 1.2.3 软件工程

- 软件工程概念的提出
  - 为了克服软件危机，北约组织成员国的软件工作者于1968年和1969年两次召开会议 (NATO学术会议)，提出了“软件工程”概念，著名的瀑布模型也同期出现。
    - ✧ 采用工程的概念、原理、技术和方法来开发与维护软件
    - ✧ 把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来
  - 软件工程进展
    - ✧ 软件工程理论与实践水平不断发展。
    - ✧ 软件工程学的重要分支包括软件开发方法学、软件需求工程、软件过程控制与改进、软件质量工程、软件可靠性工程、软件能力成熟度 (CMM)、软件测试与认证以及软件可信计算等。

## 1.2 软件开发

---

### 1.2.3 软件工程

- 软件工程要点

1. 软件开发是一个有计划、分阶段、严格按照标准或规范进行的活动。
  - ✧ 软件工程是指导计算机软件开发和维护的工程学科。
  - ✧ 软件工程方法是结合了工程、管理和技术的方法。
2. 软件工程将系统的、规范的、可度量的方法应用于软件的开发、运行和维护的过程。
  - ✧ 将工程化应用于软件开发过程中，对上述方法进行研究
3. 采用适当的软件开发方法和支持环境以及编程语言来表示和支持软件开发各阶段的各种活动，并使开发过程条令化、规范化，使软件产品标准化，开发人员专业化。

## 1.2 软件开发

---

### 1.2.3 软件工程

- 软件工程要点 (续)
  4. 用工程学的观点进行费用估算，制定进度和计划；用管理科学中的方法和原理进行软件生产的管理；用数学的方法建立软件开发中的各种模型和各种算法。



## 1.2 软件开发

---

### 1.2.3 软件工程

- 软件工程过程
  - 软件工程过程是为了获得软件产品，在软件工具的支持下由软件工程师完成的一系列软件工程活动。
  - 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基本步骤包括：
    1. 软件规格说明 Specification
      - ✧ 问题分析：
        - 需求获取和定义
      - ✧ 需求分析：
        - 生成软件需求规格说明文档
        - 规定软件的功能及其使用限制

# 1.2 软件开发

---

## 1.2.3 软件工程

- 软件工程过程 (续)

### 2. 设计与实现 Development

- ✧ 概要设计：建立整个软件的体系结构，包括子系统、模块以及相关层次的说明、每一模块的接口定义等
- ✧ 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
- ✧ 把设计结果转换为可执行代码，产生满足规格说明的软件

### 3. 确认 Validation

- ✧ 通过对产品的有效性验证，确保软件满足用户的要求

### 4. 演进 Evolution

- ✧ 为了满足客户的变更要求，软件必须在使用过程中进行不断地改进和完善

## 1.2 软件开发

---

### 1.2.3 软件工程

- 软件工程原则
  - 采取适宜的开发模型
    - ✧ 控制易变的需求
  - 采用合适的设计方法
    - ✧ 需要软件模块化、抽象与信息隐藏、局部化、一致性以及适应性等，需要合适的设计方法的支持
  - 提供高质量的工程支持
    - ✧ 软件工具和环境对软件过程的支持
  - 重视开发过程的管理
    - ✧ 有效利用可用的资源、生产满足目标的软件产品、提高软件组织的生产能力等

## 1.2 软件开发

---

### 1.2.3 软件工程

- 软件工程目标
  - 软件工程的目标是生产具有正确性、可用性以及开销适宜、进度保证、并且项目成功的软件产品。
    - ◇ 正确性：
      - 软件产品达到预期功能及性能程度
    - ◇ 可用性：
      - 软件基本结构、实现及文档为用户可用程度
    - ◇ 开销适宜：
      - 软件开发、运行的整个开销 (包括资金开销、时间开销) 满足用户要求的程度
    - ◇ 成功的软件项目
      - 开发成本低、功能与性能满足需求、易于移植、维护方便以及按时完成开发任务并及时交付软件产品



## 1.2 软件开发

---

### 1.2.3 软件工程

- 软件工程目标 (续)
  - 软件工程目标决定了软件工程过程、过程模型 (软件周期模型) 和工程方法的选择。

## 1.2 软件开发

---

### 1.2.3 软件工程

- 软件工程学
  - 软件开发技术
    - ◇ 软件开发方法学
    - ◇ 软件工具
    - ◇ 软件工程环境
  - 软件工程管理
    - ◇ 软件工程管理学
    - ◇ 软件经济学



## 1.2 软件开发

---

### 1.2.4 软件开发技术和方法

- 软件开发技术
  - 完成软件生命周期各阶段的任务所必须具备的技术手段，包括软件开发方法学、软件工具和软件工程环境。
- 软件开发方法
  - 研究如何在规定的投资和时间内，开发出符合用户需求的高质量的软件。
  - 它是一种使用预先定义的技术集及符号来组织软件生产过程的方法。
    - ✧ 包括：项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法的设计、编码、测试以及维护等

## 1.2 软件开发

---

### 1.2.4 软件开发技术和方法

- 软件开发方法 (续)
  - 面向数据流的结构化程序开发方法 (最终关注程序结构)
    - ✧ 指导思想
      - 自顶向下，逐步求精
    - ✧ 基本原则
      - 功能的分解与抽象
    - ✧ 适用性
      - 适合数据处理领域的问题

## 1.2 软件开发

---

### 1.2.4 软件开发技术和方法

- 软件开发方法 (续)
  - 面向数据结构的开发方法 (Jackson 方法, 1975)
    - ✧ JSP (Jackson Structured Programming)
      - 描述问题的输入、输出数据结构，分析其对应性
      - 设计相应的程序结构，从而给出问题的软件过程描述
      - 以数据结构为驱动
    - ✧ JSD (Jackson System Development)
      - 标志实体与行为
      - 生成实体结构图
      - 创建软件系统模型
      - 扩充功能性过程
      - 施加时间约束
      - 系统实现

## 1.2 软件开发

---

### 1.2.4 软件开发技术和方法

- 软件开发方法 (续)
  - 基于模型的方法 (支持程序开发的形式化方法)
    - ✧ 维也纳方法 (VDM, 1969- )
      - 将软件系统当作模型来给予描述, 把软件的输入、输出看作模型对象, 把这些对象在计算机内的状态看作该模型在对象上的操作
  - 面向对象的开发方法
    - ✧ 指导思想
      - 尽可能按照人类认识世界的方法和思维方式来分析和解决问题
    - ✧ 面向对象方法包括
      - 面向对象分析; 面向对象设计; 面向对象实现

## 1.2 软件开发

---

### 1.2.5 软件开发工具和环境

- 软件工具

- 为了支持软件人员开发和维护目标系统而使用的软件

- ✧ 为软件开发方法提供自动的或半自动的软件支撑环境，辅助完成软件开发任务

- ✧ 例：微软的 Visual Studio Team Suite 是基于 Microsoft Solution Framework 的支持工具套件，通过 Team Foundation Server 对项目研发过程中的项目管理、版本管理、工作项跟踪、报告、软件构件等内容的管理，并提供相应的测试工具，是一类软件开发生命周期或应用生命周期管理工具 (SDLC/Application Life Cycle Management)。

- <http://msdn.Microsoft.com/zh-cn/teamsystem/default.aspx>

- ✧ 例：IBM/JAZZ、HP/BTO、MocroFocus/Open ALM 等平台。

## 1.2 软件开发

---

### 1.2.5 软件开发工具和环境

- 软件工程环境
  - 支持软件开发的整个生存周期的资源集合
    - ✧ 用以提高软件开发效率，提高软件质量，降低软件开发成本
- 计算机辅助软件工程 CASE
  - 为软件开发提供一组优化集成的且能大量节省人力的软件开发工具
    - ✧ 实现软件生存期各环节的自动化并使之成为一个整体



## 1.2 软件开发

---

### 1.2.6 软件生命周期

- 软件生命周期的概念
  - 计算机软件有一个孕育、诞生、成长、成熟、衰亡的生存过程，即软件的生命周期 (也称软件开发生命周期 SDLC 或软件开发过程)。软件生命周期被划分为若干阶段，每个阶段有明确的任务，从而使规模、结构和管理复杂的软件开发过程得到适当的控制和管理。
  - 软件生命周期包括可行性分析与开发计划、需求分析、设计 (概要设计和详细设计)、编码实现、测试、运行与维护等活动，将这些活动以适当的方式分配到不同的阶段去完成。

◇ GB/T 8567-2006: 《计算机软件文档编制规范》



## 1.2 软件开发

---

### 1.2.6 软件生命周期

- 软件生命周期的6个阶段

- 可行性分析与计划阶段

- ✧ 确定软件开发的总体目标，给出功能、性能、可靠性以及接口等方面的要求，进行完成可行性分析。
    - ✧ 估计可利用的资源 (硬件、软件、人力等)、成本、效益、开发进度，进行投资-收益分析，制订开发计划。
    - ✧ 提交可行性分析报告、开发计划等文档。

- 需求分析阶段

- ✧ 分析用户提出的要求，给出需求详细定义，确定软件系统的各项功能、性能需求和设计约束，确定对文档编制的要求。
    - ✧ 提交软件需求说明、软件规格说明、数据要求说明等文档和初步的用户手册。

## 1.2 软件开发

---

### 1.2.6 软件生命周期

- 软件生命周期的6个阶段 (续)

- 设计阶段

- ✧ 概要设计：把各项需求转换成软件的体系结构。结构中每一组成部分都是意义明确的模块，每个模块都和某些需求相对应。
    - ✧ 详细设计：对每个模块所要完成的工作进行具体的描述，提供源程序编写的直接依据。
    - ✧ 提交结构设计说明、详细设计说明和测试计划初稿等文档。

- 实现阶段

- ✧ 完成源程序的编码、编译 (或汇编) 和排错调试，得到没有语法错误的程序清单。程序结构良好、清晰易读，且与设计相一致
    - ✧ 编写进度日报、周报和月报 (取决于项目的重要性的和规模)。
    - ✧ 提交用户手册、操作手册等面向用户的文档的编写工作。
    - ✧ 编制测试计划。

## 1.2 软件开发

---

### 1.2.6 软件生命周期

- 软件生命周期的6个阶段 (续)

- 测试阶段

- ✧ 全面测试目标软件系统，并检查审阅已编制的文档，提交测试分析报告。逐项评价所生产的程序、文档以及开发工作本身，提交项目开发总结报告。
    - ✧ 在整个开发过程中 (即前五个阶段中)，开发集体需要按月编写开发进度月报。

- 运行与维护阶段

- ✧ 软件提交给用户后，在运行使用中得到持续维护，根据用户新提出的需求进行必要而且可能的扩充、删改、更新和升级。
    - ✧ 软件维护包括改正性维护 (发现错误)、适应性维护 (适应运行环境变化) 和完善性维护 (增强功能)。

# Thank you!

