

# 软件工程引论 - 从程序设计到软件工程

---

## 程序设计

- 程序设计语言的发展：机器语言、汇编语言、面向过程的高级语言、面向对象的高级语言、面向特定领域的现代语言。
- 程序设计方法：结构化程序设计、面向对象程序设计。

## 软件系统的特点

- 逻辑产品，逻辑思维活动的结果。
- 创造过程和制造过程合二为一。
- 管理和控制软件开发过程困难。
- 依赖于开发人员的业务素质、智力、合作、组织和管理。
- 开发质量难以评价，进展情况难以衡量。

## 软件系统的发展和演化

- Cyber-Physical Systems (CPS)、普适计算、基于Internet的计算。
- 软件系统的规模和复杂性不断增大。

## 软件危机

- 供求关系失调、开发费用失控、进度拖延、可靠性差、难以维护。
- 影响：困扰学术界和工业界，用户对软件开发缺乏信心，高投入和高风险。

## 产生软件危机的原因

- 软件开发人员的错误观点。
- 软件本身的特点：进展情况难以衡量、开发质量难以评价、管理和控制困难。

## 解决危机的技术途径和管理途径

- 提出有效的方法和工具支持软件开发。
- 认识到管理的重要性，70%的项目由于管理不善导致难以控制进度、成本和质量。

## 软件工程

- 为克服软件危机而提出的概念，应用计算机科学、数学及管理科学等原理，以工程化原则、方法解决软件问题。

## 软件工程的构成要素

- 过程、方法、工具、人员、软件质量属性。

## 软件工程发展史上的里程碑

- 70年代：程序设计方法学、结构化分析和设计、抽象数据类型、软件工具。
- 80年代：软件开发方法学、软件工程环境（CASE）、面向对象技术。
- 90年代至今：软件过程改进（CMM）、软件复用和软件构件技术、Unified Modeling Language (UML)、Model Driven Architecture (MDA)。

## 软件工程目标

- 可修改性、有效性、可靠性、可理解性、可维护性、可重用性、可适应性、可移植性、可追踪性、可互操作性。

## 软件工程原则

- 抽象、信息隐藏、模块化、局部化、一致性、完整性、可验证性。

## 软件工程道德准则

- IEEE-CS/ACM联合工作组报告提出的软件工程道德规范和专业实践的8条原则。

## 讨论

- 为什么软件产品没有“质保”，没有“三包”？

软件产品没有“质保”或“三包”主要源于其特殊性质和行业惯例：

1. **\*\*无形性\*\***：软件是数字产品，没有物理磨损问题，无法按照传统商品标准提供保修。

2. **\*\*复杂性 & 不可预测性\*\***: 软件可能包含未发现的缺陷，尤其是在复杂系统中，完全无错误几乎不可能。
  3. **\*\*责任界定困难\*\***: 错误可能源于用户操作、硬件环境或第三方组件，难以界定责任。
  4. **\*\*快速更新迭代\*\***: 软件通常通过更新修复问题，新版本往往替代旧版本，提供持续服务而非传统的保修承诺。
  5. **\*\*行业惯例\*\***: 软件协议（如EULA）通常明确限制责任，用户需接受“按现状提供”的使用条件。
- 不过，部分软件提供**\*\*服务保障\*\***，如技术支持、补丁更新和订阅服务，等同于变相的“质保”或“包修”。

- 回顾软件发展历史上的软件效率提升的里程碑，展望未来大模型时代软件工程的发展趋势？

#### ### 软件效率提升的里程碑

1. **\*\*汇编语言到高级语言\*\***: 提升开发效率，减少硬件依赖。
2. **\*\*结构化编程（1970s）\*\***: 引入模块化和流程控制，改善代码可读性和维护性。
3. **\*\*面向对象编程（1980s）\*\***: 通过封装、继承和多态简化复杂系统的建模。
4. **\*\*敏捷开发方法（2000s）\*\***: 强调迭代和用户反馈，提高响应变化的能力。
5. **\*\*DevOps和自动化工具（2010s）\*\***: 通过持续集成和部署优化开发-运维协作效率。

#### ### 大模型时代的软件工程发展趋势

1. **\*\*AI辅助开发\*\***: 代码生成、优化和错误检测，降低人力成本。
2. **\*\*模型驱动架构（MDA）优化\*\***: 用AI模型直接生成和验证系统设计。
3. **\*\*数据驱动工程\*\***: 数据将成为软件开发的核心，训练、验证和维护模型的数据质量尤为关键。
4. **\*\*自动化决策和智能化工具链\*\***: 如智能需求分析、测试生成等。
5. **\*\*新型协作模式\*\***: 开发者与AI共同完成更复杂的任务，重新定义团队角色。

未来，大模型将加速软件工程的智能化，但工程实践的可靠性、安全性和伦理问题需特别关注。