软件质量与管理
课程介绍

南京大学智能软件与工程学院
邵栋、荣国平

课程动机——为什么要有这门课程?

- 核心课程——唯一一门系统讲解软件开发管理的课程
- 对于当前本应掌握的知识，需要一些概念的厘清与脉络的梳理
  - 软件项目管理
  - 软件生命周期
  - 软件过程
  - 软件过程管理
  - 敏捷软件开发
  - CMM/CMMI
  - 瀑布模型
  - ……
- 对于将来的职业生涯和发展，需要有一些必要的准备
  - DevOps
  - ……

课程目标

- 理解项目管理的基本概念，掌握项目管理的常用方法。例如估算和计划跟踪，配置管理，风险管理等。
- 掌握产品质量和过程质量的基本概念，理解通过过程质量管理来保障最终产品质量或服务质量的手段。
- 掌握软件过程的基本概念，了解常用软件过程方法。
- 掌握敏捷软件开发价值观、实践。
- 面临相对复杂项目的时候，能够选择适用的软件过程，对其进行合理组合和裁剪，并在此基础上合理组织和管理项目开发，达到预先设定的项目目标（工期、质量、成本等）。

软件工程的底层元素

- 首先剥离表象，分解软件工程的本质组成部分：
  - 目标：将人类需求转化为可运行的数字化系统
  - 资源：开发者时间、硬件算力、数据、资金
  - 约束：
    - 时间限制：不能一个人开发现代软件
    - 质量要求
    - 团队认知差异
  - 熵增定律：代码复杂度天然趋向混乱（技术债务）

本质问题

- 原始状态假设：如果没有任何管理措施（仅仅编程），会发生什么？

  - 无序开发：每个开发者基于个人习惯编码，系统难以集成
  - 需求黑洞：客户需求被无限次修改，无法收敛
  - 质量失控：BUG随代码量指数级增长（每千行代码约15-50个错误）
  - 资源浪费：80%时间耗费在沟通与返工（布鲁克斯定律）

  - 无管理的软件系统会自发趋向混乱（代码冗余、架构腐化、文档缺失）。

熵增定律（热力学第二定律）

- 熵（Entropy）：衡量系统的无序程度或混乱度。
  - 高熵：系统混乱（如气体扩散、碎片散落）。
  - 低熵：系统有序（如冰块结晶、整齐排列的书架）。
- 熵增定律的核心表述
  - 孤立系统中，熵永不减少：
  - 若没有外界干预（如能量输入或物质交换），系统的熵会自发增加，直到达到最大无序状态（热力学平衡）。

代码维护与熵增

- 未维护的代码 → 孤立系统熵增
  - 代码随着时间推移（新功能添加、多人修改、需求变更）会自发变得复杂、冗余（熵增）。
- 重构 → 注入能量对抗熵增
  - 通过重构（重新组织代码结构）降低混乱度，相当于向系统输入“负熵”。
- 例子：
  - 一个函数最初清晰简洁（低熵），但随着多次紧急修改，逐渐变得冗长且包含重复逻辑（高熵）。定期重构如同整理房间，恢复秩序。

软件工程管理的必然性

- 软件熵增结果：软件死亡（增加新功能成本大于从新开发）
- 对抗熵增的终极手段，通过流程（Scrum）、工具（CI/CD）、规范持续注入负熵。

内容安排 （1）

- 敏捷方法
  - Scrum：三个角色、三个工件、五个价值观、五个事件
  - XP：CICD、重构、简单设计
  - Kanban
  - 敏捷软件开发概述
  - 敏捷软件开发和瀑布模型的区别

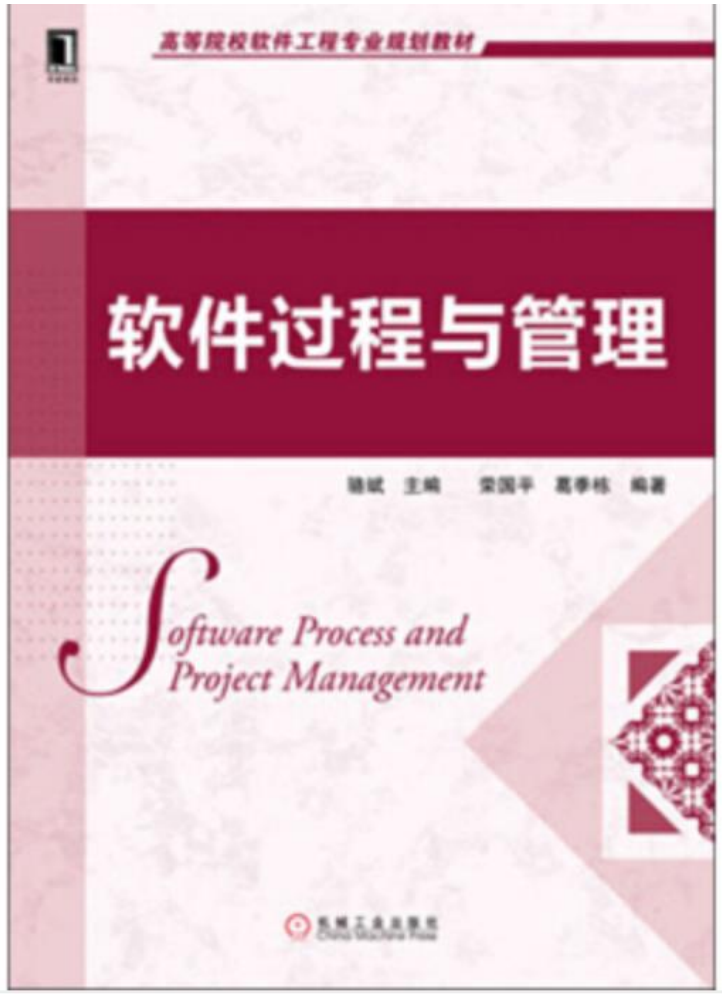
内容安排（2）

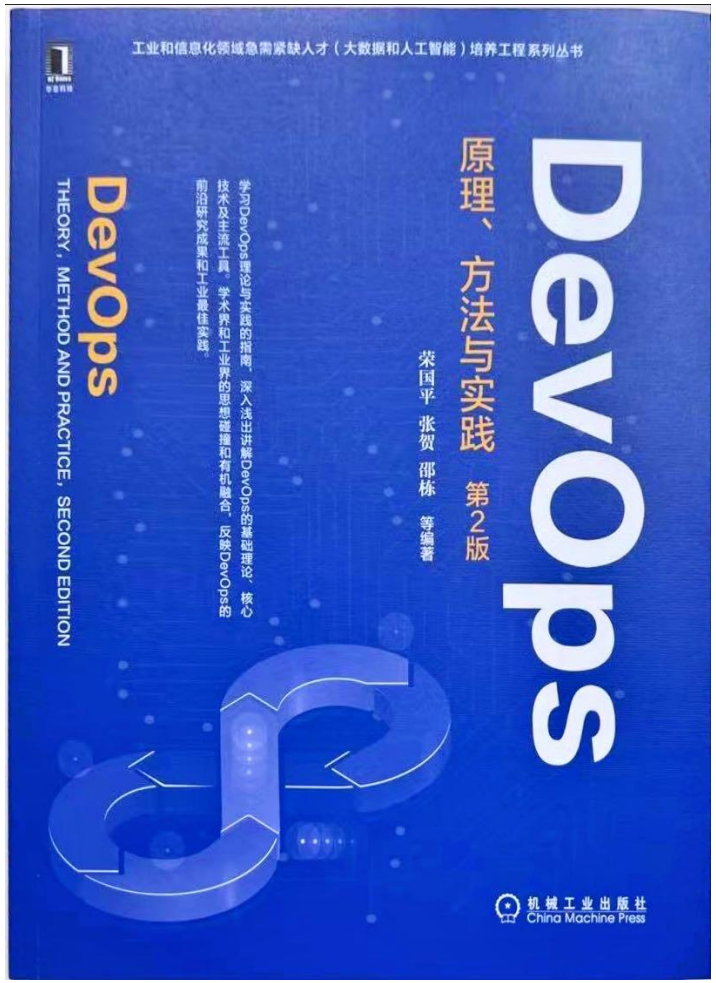
- 基本概念、软件过程的历史演变和经典工作
- 技术人员管理和团队组建
- 估算、计划和跟踪
- 质量管理
- 工程技术管理
- 支持类活动

课程组织

- 课堂上课
- 平时作业组成
  - 读书笔记
  - 课堂练习或课堂讨论，加分奖励**答得好的同学**
  - 学期课程实践
- 期末考试

教材和参考书


高等院校软件工程专业规划教材
软件过程与管理
Software Process and Project Management


工业和信息化领域急需紧缺人才（大数据和人工智能）培养工程系列丛书
DevOps
原理、方法与实践 第2版
荣国平 张贺 邵栋 等编著
DevOps
THEORY, METHOD AND PRACTICE, SECOND EDITION
机械工业出版社
China Machine Press

联系方式

- 邵栋
- 南雍楼 东 209
- dongshao@nju.edu.cn

- 荣国平
- 费彝民楼B-917
- ronggp@nju.edu.cn

THANKS

南京大學
NANJING UNIVERSITY