

# **Software Engineering**

## **Course Review**

### **2023-06**

软件工程课程复习 2023-06

# Textbook

《Software Engineering A Practitioner's Approach》 10th Edition  
Roger S. Pressman

教科书

《软件工程实践者》  
方法》第 10 版  
罗杰 · S 普雷斯曼

# 1: The Nature of Software

- **1. Software**
  - **Definition of software**
  - **Characteristics of Software**
  - **The difference of software and hardware**
- **2. The changing nature of software**

## 1、软件的本质

- 1. 软件. 软件的定义. 软件的特性. 软件和硬件的区别
- 2. 软件性质的变化

## 2: Software Engineering

- 1. Software engineering – a layered technology:
  - The definition of Software engineering
  - The goal of Software engineering
  - Layer: tools, methods, process and a quality focus
- 2. A process framework
  - The generic five process activities: communication, planning, modeling, construction and deployment
- 3. Software development myths
- 4.Umbrella activities

Software project tracking and control; Risk management; Software quality assurance; Technical reviews; Measurement; SCM; Reusability management; Work product preparation and production;

### 2: 软件工程

- 1. 软件工程-分层技术: 软件工程的定义
  - 软件工程的目标-层: 工具, 方法, 过程和质量焦点
- 2. 过程框架--通用的五个过程活动: 通信、规划、建模、构建和部署
- 3. 软件开发神话
- 4. 伞式活动

软件项目跟踪与控制;风险管理;软件质量保证;技术评审;度量;供应链管理;可重用性管理;工作产品准备与生产;

# 3: Software Process Structure

- **1. Prescriptive models**
  - The function of process models
  - Understand the signification and characteristics of the process models
  - Process model, Pattern, Framework
- **2. The waterfall model**
  - V cycle model
  - 适合需求清楚、熟悉的系统
- **3. Incremental process models (阶段式提交)**
  - 适合需求清楚、周期比较短的项目
  - OO-based
  - Why use incremental model?

## 3: 软件过程结构

- 1. 说明性模型–过程模型的功能–理解过程模型的含义和特征–过程模型、模式、框架
- 2. 瀑布模型– V 循环模型
  - 适合需求清楚、熟悉的系统
- 3. Incremental process models (阶段式提交) – 适合需求清楚、周期比较短的项目
  - 基于 OO–为什么要使用增量模型？

### **3: Software Process Structure**

- **4. Evolutionary process**
  - **Prototyping**
    - 布模型的改进，适合需求不清楚的系统
    - **Process pattern**
  - **Spiral Model** （风险分析）
- **5. Specialized process models**
  - **Component based development** （需要面向对象技术支持）
  - **Object-oriented process models**
- **6. Unified process model(5个阶段)**
  - **Inception**（起始）, **Elaboration**（细化）, **Construction**（构建）, **Transition**（转换）, **Production**（生产）

#### 3: 软件过程结构

- 4. 演化过程-原型
  - 布模型的改进，适合需求不清楚的系统
  - **Process pattern**
- Spiral Model （风险分析）
- 5. 专用过程模型-基于组件的开发（需求面向对象技术支持）
  - 面向对象的过程模型
- 6. Unified process model(5 个阶段) – Inception（起始）, Elaboration（细化）, Construction（构建）, Transition（转换）, Production（生产）

# Agile Development

- 1. What is Agility?
- 2. Agile Process
  - XP (pair programming 结对编程)
  - Scrum

## Agile Development

- 1. 什么是敏捷?
- 2. Agile Process – XP (pair programming 结对编程) – Scrum

## 4: Understanding Requirements

- **1. A bridge to design and construction**
  - The definition of requirements engineering
- **2. seven Requirements engineering tasks**
  - Inception (起始)                      Elicitation (导出)              Elaboration (精化)  
Negotiation (协商)    Specification (规格说明)      Validation (确认)
  - Requirements management (需求管理)
- **3. Initialing the requirements engineering process**
- **4. Eliciting requirements**
  - 通过开发系统原型获取用户需求
- **5. Developing user-case**

### 4: 了解需求

- 1. 设计和施工的桥梁-需求工程的定义
- 2. 7 个需求工程任务
  - Inception (起始) Elicitation (导出) Elaboration (精化) Negotiation (协商)  
Specification (规格说明) Validation (确认)
  - Requirements management (需求管理)
- 3. 初始化需求工程过程
- 4. Eliciting requirements – 通过开发系统原型获取用户需求
- 5. Developing user-case

# 5: Requirements Modeling

- **1. Requirements analysis**
  - The three goals of analysis modeling (Information/Data, Function, Behavioral)
  - The concepts of analysis modeling
  - Specification and Requirements
  - Customer and End-User
- **2. Analysis modeling approaches**
  - The principles of modeling
- **3. Data modeling concepts**
  - E-R diagram, relationship of objects
- **4. Scenario-based modeling**
  - UML
  - Use-Cases in UML: use-case diagram/ activity diagram/ sequence diagram/state diagram/class diagram
  - OO analysis: Behavioral, Class, Use-Case
- **5. Creating a behavioral model**
- **6. Class-based modeling**
  - Identifying analysis classes
  - CRC(class-responsibility-collaborator) Modeling

## 第五章：需求建模

- 1. 需求分析-分析建模的三个目标（信息/数据，功能，行为）-分析建模的概念-规范和需求-客户和最终用户
- 2. 分析建模方法. 建模原理
- 3. 数据建模概念. E-R 图、对象关系
- 4. 基于场景的建模- UML
  - UML 中的用例：用例图/活动图/序列图/状态图/类图
  - 面向对象分析：行为、类、实例
- 5. 创建行为模型
- 6. 基于类的建模识别分析类 CRC (class-responsibility-collaborator) 建模

# 6: Design Concepts

- 1. Design within the context of software engineering
  - Map the analysis model into design model
- 2. Design concepts
  - abstraction, Refinement , architecture, patterns, modularity, information hiding, functional independence, refactoring, design class
- 3. The design model
  - the concepts of the design process
  - four design models : Data Design, Architectural Design, Interface Design, Component-Level Design
  - 4 characteristics of a well-formed design class: Complete and sufficient, Primitiveness (原始性) , High cohesion, Low coupling (高类聚低耦合)
  - Analysis Model and Design Model (二者关系: 过程维度、抽象维度)

## 第六章：设计理念

- 1. 在软件工程的背景下进行设计-将分析模型映射到设计模型
- 2. 设计概念-抽象、细化、架构、模式、模块化、信息隐藏、功能独立性、重构、设计类
- 3. 设计模型-设计过程的概念-四个设计模型：数据设计、架构设计、界面设计、中间层设计-一个良好形式的设计类的 4 个特征：完整和充分、原始性（原始性）、高内聚性、低耦合性（高类聚低合）-分析模型和设计模型（二者关系：过程维度、抽象对象  
维度）

# 7: Architectural Design

- **1. Software Architecture**
  - The definition of architectural
- **2. Data design**
  - The goal of Data Design in the Architectural Design
- **3. Architectural styles and patterns**
  - components, connectors, constraints, semantic (语义) models;
  - Data-centered, Data-flow, Call and return, Object-oriented, Layered architectures
  - Architectural complexity: dependencies (三种依赖关系)

## 7: 建筑设计

- 1. 软件架构-架构的定义
- 2. 数据设计--建筑设计中数据设计的目标
- 3. 体系结构样式和模式-组件、连接器、约束、语义模型；-以数据为中心、数据流、调用和返回、面向对象、分层体系结构-体系结构复杂性：依赖关系（三种依赖关系

系)

# 8: Component-level Design

- **1. What is a component**
  - OO view    Conventional view
- **2. Design Class-based component**
  - **Basic design principles**
    - 4个基本设计原则
      - ①开闭原则。
      - ②Liskov替换原则。
      - ③依赖倒置原则。
      - ④接口分离原则。
    - Two qualitative criteria for measuring module independence:Cohesion & Coupling
    - Analysis Class and Design Class
- **3. Conducting component-level design**
  - The steps of OO Component-level Design
- **4. Design conventional components**
  - flow diagram 流程图

## 第八章：智能级设计

- 1. 什么是组件- OO 视图常规视图
- 2. 基于类的设计组件. 基本设计原则
  - 4个基本设计原则
    - ①开闭原则。
    - ②Liskov替换原则。
    - ③依赖倒置原则。
    - ④接口分离原则。
  - 度量模块独立性的两个定性标准：内聚性和耦合性-分析类和设计类
- 3. 进行组件级设计-OO 组件级设计的步骤
- 4. 设计常规组件-流程图流程图

## 9: User Interface Design

- **1. The Golden Rules**
- **2. User Interface analysis and design**
  - user analysis, task and work environment analysis, Interface design, Interface validation
- **3. Interface analysis**
  - Steps of interface analysis
- **4. Interface design Steps**
  - Design GUI according to Use-Case Diagram

### 第九章：用户界面设计

- 1. 金科玉律
- 2. 用户界面分析和设计-用户分析、任务和工作环境分析、界面设计、界面验证
- 3. 界面分析-界面分析步骤
- 4. 界面设计步骤-根据用例图设计 GUI

# 10: Testing strategies and techniques

- **1. A strategic approach to software testing**
  - Verification and validation (验证与确认)
- **2. Test strategies for conventional software (过程与文档)**
  - Unit testing
  - integration testing
    - Top-Down integration & Bottom-Up integration
    - regression (回归) testing & smoke (冒烟) testing
  - Acceptance testing (Validation testing)
  - System testing
- **3. Validation testing**
- **4. System testing**
  - Use-Case Diagram
  - Function testing, specification
- **5. The art of debugging**
  - The relationship of testing and debugging

## 10: 测试策略和技术

- 1. 软件测试的战略方法-验证与确认（验证与确认）
- 2. 常规软件的测试策略（过程与文档）
  - 单元测试-集成测试

自上而下的集成和自下而上的集成回归（回归）测试和冒烟（冒烟）测试-验收测试（验证测试）-系统测试

- 3. 确认测试
- 4. 系统试验. 第 1 部分: 用例图. 第 1 节:  
功能试验规范
- 5. 调试的艺术-测试与调试的关系

# 10: Testing strategies and techniques

- **6. White-Box testing**
  - Flow Graph Notation
  - Cyclomatic Complexity (圈复杂度与独立路径)
- **7. Basis path testing**
- **8. Control structure testing** (条件/循环)
- **9. Black-Box testing**
  - Equivalence Partitioning (等价类划分)
  - Boundary Value Analysis (边界值分析)
- **10. OO Testing Methods**

## 10: 测试策略和技术

- 6. 白盒测试-流程图表示法
    - Cyclomatic Complexity (圈复杂度与独立路径)
  - 7. 基路径测试
  - 8. 控制结构测试 (条件/循环)
  - 9. 黑盒测试. 等价划分. 边界值分析
- 
- 10. OO 测试方法

# 11: Project Management

- 1. 4 P's
  - People
  - Product
  - Process
  - Project
- 2. SQA
- 3. Risks management
- 4. SCM
  - scm task

## 第 11 章项目管理

- 1. 4P-人

- 产品
  - 过程
  - 项目

- 2. SQA
- 3. 风险管理
- 4. SCM 任务

# 期末考试内容和形式

- 简答论述题（共3小题，共35分）
- 非标准答案题（共1小题，共15分）
- 应用、设计及分析题（共4小题，共50分）

## 期末考试内容和形式

- 简答论述题（共3小题，共35分）
- 非标准答案题（共1小题，共15分）
- 应用、设计及分析题（共4小题，共50分）

# Q & A

Q & A