Software Engineering Course Review 2023-06

Textbook

《software Engineering A Practitioner’s Approach》10th Edition
Roger S. Pressman

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

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;

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: 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）

Agile Development

- 1. What is Agility?

- 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

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

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）

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）

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

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

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 & Botton-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: 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

11: Project Management

- 1. 4 P’s
  - People
  - Product
  - Process
  - Project
- 2. SQA

  3. Risks management
- 4. SCM

  scm task

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