Software School

Fudan University

# SOFTWARE ENGINEERING
for Undergraduates

SOFT130006.01

pengxin@fudan.edu.cn

COURSE REVIEW 2016

FINAL EVALUATION

- Exercise & Labs (~40%)
- Final Examinations (~45%)
- Classroom Performance (~10%)
- Online Interaction (~5%)

Final Examination

- Date: June 24th (Friday), 2016
- Time: 1:30-3:30 PM
- Place: Z2108

EXAMINATION STRUCTURE

- Multiple choice (3’×10)
- Analysis (~27’)
- Design (~28’)
- Testing (~15’)

OVERVIEW

- Introduction to Software Engineering
- Software Engineering Process
  - Software Process Concept
  - Process Model
  - Agile Process
- Software Engineering Practice
  - Requirements Engineering
  - Design Engineering
  - Software Testing

INTRODUCTION TO SOFTWARE ENGINEERING

- Concepts: software, software engineering
- Hardware Vs. Software
  - Manufacturing vs. Development
  - Wear out vs. Deterioration
- Software Evolution and Evolution Laws
- Software Myths

Software Process

- Concepts: software process, process model
- Software Process Framework
- Process Assessment
  - CMM/CMMI, SPICE, ISO 9001
- CMMI
- PSP/TSP

PROCESS MODEL AND AGILE PROCESS

- Process Model
  - Waterfall Model
  - Incremental Models
  - Evolutionary Models
    - Prototyping Model
    - Spiral Model
  - Unified Process (UP)
- Agile Process
  - Assumptions, manifesto, principles
  - Comparison with traditional process

Requirement Engineering-1

- Concepts: Software Requirements and Requirements Engineering
- Requirements Engineering Activities
- Data Modeling
- Scenario-Based Modeling
  - Use Case and Use Case Elements
  - Use Case Description
    - Text-based Template
    - UML Diagrams: Use Case Diagrams, Activity Diagrams (Swim Lane Diagrams)

REQUIREMENT ENGINEERING-2

- Flow-Oriented Modeling
  - Data Flow Model: Data Flow Diagram
    - DFD Elements
    - Top-level (Level-0) DFD (Context Diagram)
    - Hierarchical Refinement : Level-1 DFD, Level-2 DFD, ……
    - Consistency in DFD Refinement, number……
  - Control Flow Model

REQUIREMENT ENGINEERING-3

Class-Based Modeling

- Analysis classes identification
- Attribute and operation definition
- Class-Responsibility-Collaborator (CRC) Modeling
- Modeling Associations and Dependencies
- UML Diagrams: Class Diagram

Behavioral Modeling

- Behavioral model elements: event, state, transition…
- UML Diagrams: Sequence Diagram, State (Machine) Diagram

DESIGN ENGINEERING-1

- Software Design Layers
  - data, architecture, interface, component-level
- Design Principles: Modular Design
  - Abstraction
  - Information hiding
  - Encapsulation
  - Functional Independence: cohesion, coupling
- Refactoring

DESIGN ENGINEERING-2

- Software Architecture
  - Architectural Styles
  - Architecture Assessment (ATAM)
  - Structured Design
- Component-Level Design
  - OO Design Principles
  - Cohesion Types
  - Coupling Types
  - Component-Level Design Steps

DESIGN ENGINEERING-3

- UML Diagrams for Design Modeling
  - Component Diagram, Class Diagram, Package Diagram, Sequence Diagram, State (Machine) Diagram, Deployment Diagram
- User Interface Design
  - Typical Design Errors
  - UI Design Rules
    - Place the user in control
    - Reduce the user's memory load
    - Make the interface consistent
  - UI Design Models
  - Interface Analysis: people, tasks, content, environment

TESTING STRATEGY

- Concepts: testing, test case, V&V
- Testing Strategy: from Small to Large
  - Unit Test: Unit Test Environment
  - Integration Test: Incremental Integration
    - Top-down, Bottom-up, Combined (Sandwich Testing)
  - High-Order Tests
    - System Test: Recovery Testing, Security Testing, Stress Testing, Performance Testing
    - Validation Test: Configuration review (Audit), Acceptance testing, Alpha/Beta Testing
- OO Testing Strategy
- Debugging

TESTING TACTICS

- Testability
- Test Case Design: objective, criteria, constraint
- White-Box Testing
  - White-Box Testing Criteria
  - Control Structure Testing: basis Path Testing, Loop Testing, Condition testing , Data flow testing
- Black-Box Testing
  - Graph-Based Methods
  - Equivalence Partitioning: equivalence class definition guidelines
  - Boundary Value Analysis: BVA guidelines
- OOT

UML DIAGRAMS

- Use Case Diagram
- Activity Diagram (Swim Lane Diagram)
- State (Machine) Diagram
- Class Diagram
- Deployment Diagram
- Package Diagram
- Component Diagram
- Sequence Diagram

End of Course Review