CSC 331/631 Midterm Study Guide

Instructor: V. Paúl Pauca

Handouts

1. Introduction

   - Professional software development vs. computer programming
   - Types of software products
   - Software engineering as a discipline
   - Importance of software engineering
   - Fundamental software engineering activities
   - Different types of software applications
   - Software engineering ethics

   Review key points and exercises

2. Software processes

   - What is a software process?
   - Software process models: waterfall, incremental development, reuse-oriented
   - Process activities: specification, design and implementation, validation, evolution
   - How to cope with change/uncertainty: prototyping and incremental delivery
   - Boehm's spiral model and relation to risk analysis
   - The rational unified process

   Review key points and exercises

3. Agile software development

   - Agile methods, incremental development and delivery
   - Pros and cons of the agile approach
   - Extreme programming principles, testing, pair programming, refactoring
   - Agile project management and scaling

   Review key points and exercises

4. Requirements engineering

   - Functional vs. non-functional requirements, examples of good and bad requirements
   - Types of non-functional requirements
   - The software requirements document and requirements specification
   - Requirements elicitation and analysis
   - Requirements discovery: interviewing, scenarios, use cases, ethnography
   - Requirements validation and management

   Review key points and exercises

Slides

1. Fundamental object-oriented programming concepts
    - Subclassing, "is a" relationship
    - Data member object references and pointers, "has a" relationship
    - UML class design
    - Polymorphic behavior and dynamic binding
    - Pure abstract classes and interfaces
    - The adapter pattern class structure and purpose
2. Software processes
    - Software life cycle and life cycle models: waterfall, incremental, spiral
    - Rapid prototyping model
    - Agile development: twelve principles
    - Extreme programming
3. System modeling
    - External, interaction, structural, and behavioral perspectives
    - UML diagrams and perspectives where they are applied
    - UML use cases and their application for capturing requirements
    - UML use case descriptions
4. Project planning and Scrum
    - The Scrum development framework
    - The Scrum process: sprint, sprint meetings, roles
    - Scrum artifacts: product backlog, sprint backlog, burndown chart
    - Conducting a sprint planning: estimating work capacity, selection of backlog items, task breakdown
    - Task board and project management