Midterm Exam Review

CS 3398
Fall 2013
Jill Seaman

Midterm Exam

- Wednesday, October 16
- Closed book, closed notes, clean desk
- Chapters 1 through 4 and 6
- 25% of your final grade
- I recommend using a pencil (and eraser)
- I will provide extra paper.

Exam Format

- Multiple choice: 17 questions
- Problems: 3
  - write (or modify) some requirements or give a scenario or use case
  - draw a use case diagram
  - draw the architecture of a system using an arch. pattern.
- Written answers: 3
  - 3 to 5 sentences, generally
  - Define, explain, compare, evaluate
  - Make a claim and support it.
- Each question will indicate how many points it is worth (out of 100)

Ch 1: Introduction

- Software Engineering: what is it?
  - Why do we need it? Project and Product failures
- Essential attributes of good software
  - Functional Correctness, Maintainability, Dependability, Efficiency, Acceptability
- Two kinds of software products
  - Generic vs customized software
- Application types:
  - Stand-alone applications
  - Interactive transaction-based apps
  - Embedded control systems
  - Batch processing systems
  - Entertainment systems
  - Systems for modeling+simulation
  - Data collection systems
  - Systems of systems
Ch 2: Software Processes

• Software process
  - A structured set of activities used to develop a software system/product.

• Software process activities
  - specification (requirements)
  - development (design and implementation)
  - validation (testing and reviews)
  - evolution (maintenance)

Ch 2: Software Processes

• Software process models
  - Waterfall model
    ✷ separate stages, sequential
    ✷ main drawback: response to change
  - Incremental development model
    ✷ series of incomplete versions
    ✷ refactoring
  - Spiral model
    ✷ risks are explicitly assessed and resolved in each loop
  - Reuse-oriented software engineering
    ✷ web services vs frameworks vs COTS

Know advantages and disadvantages of each.

Ch 2: Rational Unified Process

A hybrid model

• UP is a generic framework, RUP is a refinement of UP and a commercial product
• Must be specialized for each project
• 6 disciplines over 4 phases
  - each phase has goals, complete before next phase
  - each phase has iterations
  - one phase devoted to deployment

Disciplines correspond to the process activities
Ch 3: Agile Processes

- Agile development:
  - why needed?
  - manifesto, principles
- Extreme programming (12 practices)
  - Planning Game: story cards, task list
  - Testing: test-first development, automatic testing
  - Pair programming, continuous integration
  - Refactoring, team code ownership, sustainable pace
- Scrum
  - Project management method for incremental dev
  - Scrum master, sprint cycle, scrum team meeting
- Choosing a process (pros+cons of agile)

Ch 4: Requirements Engineering

- Requirements (definition)
  - Levels: Business, user, system
  - Functional vs non-functional
  - Desired qualities: correct, unambiguous, complete, consistent, verifiable
  - Be able to write user and system level requirements
- Software Requirements Specification Doc
  - General contents
  - Users and uses

Ch 4: Requirements Engineering

- Elicitation,
- Analysis,
- Specification,
- Validation
- Management

- Tools and methods of each sub-discipline
  - interviews, elicitation workshop, ethnography
  - Scenarios, use cases, use case diagrams
  - Prototypes, requirements review, generate test cases
  - Natural language specification, pros and cons

Ch 6: Application architecture

- Introduction
  - Terms: Architectural design, subsystem, service, subsystem interface
  - Using box and line diagrams to represent architecture
- Architectural patterns
  - ModelViewController
  - Layered
  - Client-Server
  - Pipe & Filter
  - Repository
  - Layered + client-server
Example Problems

- Assignment 1, Assignment 3
- Assignment 4: Architectural models and use case diagrams (questions and solutions on TRACS resources folder)

- See the Sample Midterm exam on the website.