Midterm Exam Review

CS 3398
Fall 2012
Jill Seaman

Midterm Exam

• Wednesday, October 9
• Closed book, closed notes, clean desk
• Chapters 1 through 4
• 25% of your final grade
• I recommend using a pencil (and eraser)
• I will bring extra paper and stapler.

Exam Format

• Multiple choice questions
• Problems
  - write (or modify) some requirements
  - give a scenario or use case
• Written (short essay) questions
  - 3 to 5 sentences, generally
  - Define, explain, compare, evaluate
• Each question will indicate how many points it is worth (out of 100)

Ch 1: Introduction

• Software Engineering: what is it?
  - Generic vs customized software
• Essential attributes of good software
  - Maintainability, Dependability, Efficiency, Acceptability
• Software process activities
  - Specification, Development, Validation, Evolution
• 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

Models

- Software process
  - plan-driven vs agile (definitions)
- Software process models
  - Waterfall model
    - main drawback: response to change
  - Incremental development model
    - series of incomplete versions
    - refactoring
  - Reuse-oriented software engineering
    - web services vs frameworks vs COTS
    - pros and cons

Ch 2: Software Processes

Activities and Change

- Software process activities
  - specification (requirements)
  - development (design and implementation)
  - validation (testing and reviews)
  - evolution (maintenance)
- Coping with change:
  - change avoidance and prototyping
    - how prototyping is used
  - change tolerance and incremental delivery
    - how different from incremental development
    - pros and cons

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

Ch 3: Agile software development

- Agile development: why, manifesto, 5 principles
- Agile vs plan-driven, when to use
- 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
Ch 4: Requirements Engineering

- Requirements (definition)
  - Levels: Business, user, system
  - Functional vs non-functional
  - Desired qualities: correct, unambiguous, complete, consistent, verifiable

- Software Requirements Specification Doc
  - Contents
  - Users and uses

Ch 4: Example Problem

1. Using a natural language format (see figure 4.1, or lecture Ch 4 slide 6) write user and system requirements for the cash-dispensing function in a bank ATM. You should assume the user’s card has already been submitted, the PIN verified, and “withdraw cash” is the chosen operation. Consider what conditions must be checked, and what information must be recorded.