

Midterm Exam Review

CS 3398
Spring 2013

Jill Seaman

1

Midterm Exam

- Wed, March 6 and Thursday, March 7
- Closed book, closed notes, clean desk
- Chapters 1 through 5
- 25% of your final grade
- I recommend using a pencil (and eraser)
- I will provide extra paper.

2

Exam Format

- Multiple choice: 17 questions
- Problems: 3
 - write (or modify) some requirements or give a scenario or use case
 - draw some diagrams/models (chapter 5)
- 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)

3

Ch 1: Introduction

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

Stand-alone applications	Entertainment systems
Interactive transaction-based apps	Systems for modeling+simulation
Embedded control systems	Data collection systems
Batch processing systems	Systems of systems

4

Ch 2: Software Processes

Models/Frameworks/Paradigms

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

Know advantages and disadvantages of each.

5

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

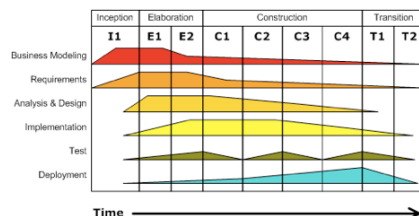
6

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



Time →

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

8

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 (see last slide)
- Software Requirements Specification Doc
 - General contents
 - Users and uses

9

Ch 4: Requirements Engineering

- Requirements Engineering
 - Elicitation,
 - Analysis,
 - Specification,
 - Validation
 - Management
 - Know goals of each sub-discipline
- 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

10

Ch 5: System modeling

- Models
 - Simple context model
 - **UML**: activity diagram, use case diagram, sequence diagram, class diagram, state diagram
 - Aggregation and generalization in class diagrams
- How models are used
 - Requirements development, design and implementation
- Be able to
 - Recognize the models.
 - Draw simple versions of the models given a problem description.
 - Answer multiple choice questions about them.

11

Example Problems

- See the **Review Exercises for midterm exam** handout (these are the modeling problems we did in class)
 - I will put it on Tracs, along with the solutions.
- See the requirements problem on the next slide.

12

Ch 4: Example Problem

1. Using a natural language format (see figure 4.1, or lecture Ch 4 slide 7) 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.