

Midterm Exam Review

CS 3398
Spring 2012

Jill Seaman

1

Midterm Exam

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

2

Exam Format

- Multiple choice questions
- Problems
 - write (or modify) some requirements
 - draw some diagrams
- Written answers
 - 1 to 5 sentences, generally
 - Define, explain, compare, evaluate
- Each question will indicate how many points it is worth (out of 100)

3

Example Problems

- See the **Review Exercises for midterm exam** handout
- I will put it on Tracs, along with the solutions.

4

Ch 1: Introduction

- Software Engineering
- Essential attributes of good software
 - Maintainability, Dependability, Efficiency, Acceptability
- Software process activities
 - Specification, Development, Validation, Evolution
- Application types

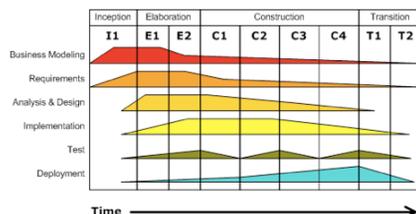
Stand-alone applications	Entertainment systems
Interactive transaction-based apps	Systems for modeling+simulation
Embedded control systems	Data collection systems
Batch processing systems	Systems of systems

Ch 2: Software Processes

- Software process
- Software process models
 - waterfall model
 - incremental development model
 - reuse-oriented software engineering
- Software process model concepts
 - plan-driven vs agile (definitions)
 - refactoring (used in incremental development)
- Coping with change:
 - software prototyping and incremental delivery

Ch 2: Rational Unified Process

- 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: how+why, manifesto, principles
- Agile vs plan-driven, when to use
- Extreme programming
 - 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 (define)
 - Business, user, system
 - Functional vs non-functional
 - Desired qualities
- Requirements Development (and management)
 - Elicitation, Analysis, Specification, Validation (interleaved)
 - Goal: Software Requirements Specification, uses of it
 - Stakeholders and analysts
- Tools, methods
 - interviews, elicitation workshop, ethnography
 - Scenarios, use case diagrams, modeling
 - Prototypes, requirements review, generate test cases

9

Ch 5: System modeling

- Perspectives
 - external, internal, structural, behavioral
- Models
 - UML: activity diagram, use case diagram, sequence diagram, class diagram, state diagram
 - Simple context model, Data-flow diagram (DFD)
 - Aggregation and generalization
- How models are used
 - Requirements development, design and implementation
- Be able to
 - Recognize the models
 - Draw simple versions of the models

10