Midterm Exam Study Guide
CSC 4700  Software Engineering
February 12, 2004,  Dr. Thomas Way

Preparation checklist:
✓ Read chapters 1-7
✓ Review the lecture slides and notes you took
✓ Although much of this material has been covered in class, some has not so these items may require additional preparation effort
✓ Make sure you know and understand the following:

1. Be able to define (words/sentence):
   • alpha testing
   • behavioral modeling
   • beta testing
   • business risks
   • CASE
   • compatibility requirements
   • computer system
   • consequential requirements
   • context model
   • COTS
   • data model
   • data-flow model
   • data-flow or activity model
   • deliverables
   • domain requirements
   • emergent properties
   • emergent requirements
   • enduring requirements
   • ethnography
   • feasibility study
   • functional requirements
   • inheritance model
   • milestones
   • mutable requirements
   • non-functional requirements
   • object model
   • product model
   • project risks
   • RE
   • role/action model
   • software
   • software design specification
   • software engineering
   • software process
   • software process model
   • software validation
   • stakeholder
   • state machine model
   • system engineering
   • system procurement
   • system requirements
   • user requirements
   • viewpoint-oriented elicitation
   • volatile requirements
   • workflow model

2. Be able to briefly example (sentence/paragraph):
   • Contrast software engineering, computer science and system engineering
   • State the breakdown of costs of software engineering
   • Describe the 4 essential attributes of good software
   • Describe the 3 key challenges facing software engineering
   • Describe the 4 primary activities of a software process
   • Explain areas of professional and ethical responsibility that apply to software engineering
• Describe 3 closely related influences on the overall reliability of a system
• List 6 types of functional system components
• List 4 fundamental activities common to all software processes
• Discuss the importance of CASE tools and provide a few examples
• Describe 3 ways in which software project management is different from other forms of project management (pp. 72-73)
• Discuss risk management when faced with specific risks (e.g., staff turnover, specification delays, poor CASE tools, underestimated development time, etc.)
• Be able to fill in the “Probability” and “Effects” columns of Figure 4.13, including additional “Risk” items not shown
• List 3 types of non-functional requirements (pp. 102-103)
• Discuss problems that can be encountered when preparing user requirements and system requirements
• If provided with the outline of an SRS (Software Requirements Specification), briefly describe the content expected in each section of the document
• Define and discuss the use of scenarios in the requirements engineering process (i.e., what is scenario-based elicitation and how is it used?)
• List 5 checks that should be included for requirements validation (p. 137)
• List 4 techniques that can be used for requirements validation (pp. 137-138)
• Explain the requirements management process, including why it is necessary (pp. 139-140)
• Explain what a CASE workbench is and what it includes (Section 7.5)

3. **Be able to explain the following figures:**
   For example, you will be presented with a figure and asked to explain it thoroughly.
   • **Chapter 2:** 2.6, 2.8
   • **Chapter 3:** 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.11
   • **Chapter 4:** 4.3, 4.4, 4.5, 4.6, 4.7, 4.11
   • **Chapter 5:** none
   • **Chapter 6:** 6.1, 6.2
   • **Chapter 7:** 7.10, 7.11, 7.12, 7.13