

Department of Computer Science and Engineering

College of Engineering, University of Nevada, Reno

CS 691z / CS 791z Topics on Software Engineering

April 1, 2007

Study required for the Midterm Test

The midterm test will be a 70-minute (1 hour and 10 minutes) closed-book exam. No supporting material is allowed. The test will take place on Thursday, April 19, 2007 from 2:30 pm in the regular classroom SEM-201. Its weight is 20% of the course grade.

For this test you are required to study material from the two books used, [Arlow and Neustadt 2002] and [Endres and Rombach 2003] and to know your own project (Concept, Specification and Design parts).

PART I: The chapters required from the [Arlow & Neustadt, 2002] book are the following:

Chapter 1	What is UML?
Chapter 2	What is UP? (sections 2.2., 2.6, 2.7, and 2.8 only)
Chapter 3	The requirements workflow (sections 3.5 and 3.6 only)
Chapter 4	Use case modeling
Chapter 5	Advanced use case modeling
Chapter 13	Activity diagrams
Chapter 17	Interfaces and subsystems
Chapter 19	Basic statecharts
Chapter 20	Advanced statecharts

From the above, the following sections or sub-sections are not required:

Section 1.3	The birth of UML
Section 1.8	UML common mechanisms
Section 2.1	Chapter 2 roadmap
Section 2.3	The birth of UP
Section 2.4	UP and the RUP
Section 2.5	Instantiating UP for your project
Section 2.9	UP phases
Section 3.1	Chapter 3 roadmap
Section 3.2	The requirements workflow
Section 3.3	Software requirements-metamodel
Section 3.4	Requirements workflow detail
Section 13.10	Signals
Subsection 17.6.2	The Façade pattern
Section 20.7	Submachine state

PART II: The laws required from the [Endres & Rombach 2003] textbook are as follows (2 or 3 laws may be added to this list, but no later than a week before the test – that is, no later than April 12):

From Chapter 2:	Glass' law (L1)
From Chapter 2:	Boehm's First law (L2)
From Chapter 2:	Boehm's Second law (L3)

From Chapter 3:	Curtis's law (L5)
From Chapter 3:	Constantine's law (L7)
From Chapter 3:	Fitts-Shneiderman law (L10)
From Chapter 4:	DeRemer's law (L11)
From Chapter 4:	Lanergan's law (L14)
From Chapter 4:	McIlroy's law (L15)
From Chapter 9:	Sackman's second law (L31)
From Chapter 9:	Brook's law (L36)
From Chapter 10:	Kupfmuller's law (L38)
From Chapter 10:	Gestalt laws (L39)

Three or four of the above laws will be given in the test. You will be required to choose two or three (out of the three or, respectively, four given) and comment on them in terms of: applicability, one significant evidence, main theory, and your own personal opinion.

Sample questions for Part I are as follows:

- 1** Briefly describe your project in CS790z (10 to 15 lines description) and give 5 examples of functional requirements and 2 examples of non-functional requirements from your team's project. In the description, indicate the goals of project and its most distinguishing characteristics. Also, indicate why you think yours is an interesting and useful project.
- 2** Explain what is meant by the "4+1 views" architecture in UML. Briefly describe the five views of this architecture.
- 3** Explain what is meant by a *use case* and what is meant by a *scenario*. Indicate what is a *primary scenario* and what are *secondary scenarios*. Give an example of a use case using the template used in the textbook for use case specification. The use case should be related to your project and should have at least 5 steps.
- 4** Indicate what is a *project glossary* and why is it this useful. Give five examples of terms that are (or could be) included in your team project glossary. Explain the meaning of the terms.
- 5** Indicate the UML notations for *activity diagrams* (start and stop states, action states, sub-activity states, transitions, decisions, forks, joins, and swimlanes). Give an example of an activity diagram that includes at least 6 action states (not counting start and stop states), a fork, a join and two swimlanes. Provide some additional (textual) description to your activity diagram to explain its meaning.
- 6** Indicate the UML syntax for *statechart diagrams*, (detailed syntax for states and detailed syntax for transitions) and enumerate the four types of event that trigger transitions (note that you need not explain each type of event). Give an example of statechart diagram that has at least 6 states (not counting start and stop states). Provide some additional (textual) description to your activity diagram to explain its meaning.
- 7** Explain why UML is considered a *unified* modeling language for software systems.
- 8** Describe the layering pattern used in software architecture, as presented in Chapter 17 of [Arlow and Neustadt, 2002].
- 9** Explain what is meant by an *interface*, indicate what are its UML symbols, and explain why interfaces are important models element in software design.
- 10** Explain what is meant by a *subsystem*, indicate what are its UML symbols, and why subsystems are important model elements in software design.