Spring 2009

CEG 460/660-01: Introduction to Software Computer Engineering

John A. Reisner

Wright State University - Main Campus, john.reisner@wright.edu

CEG 460/660: Introduction to Software Engineering

Summer Quarter, 2009

Course Description

This course introduces established practices for engineering large-scale software systems. Emphasis is placed on both the technical and managerial aspects of software engineering, and the software development process. This includes techniques for requirements elicitation, analysis, design, testing, and project management. The course emphasizes object-oriented development with the Unified Modeling Language (UML). Hands-on experience is provided through individual homework problems and a partnered project.

Textbook

System Analysis and Design with UML: An Object-Oriented Approach, by A. Dennis, B. H. Wixom, and D. Tegarden, John Wiley Publishers. This is a required textbook for the course.

Note: The instructor will be using the 3rd edition of this text, however, the 1st edition may also be used (be sure to obtain an edition of the text with the phrase "*with UML*" in its title; the text *Systems Analysis and Design* by Dennis, Wixom, and Ross is NOT an acceptable substitute).

ISBN (3rd edition): 978-0470074787
ISBN (1st edition): 978-0471413875

Reading Assignments

Most weekly content has corresponding readings from the course textbook. Some students might want to accomplish this reading prior to the lecture; others may prefer reading the text after the lecture. Still others might prefer to skim the reading assignment prior to the lecture, with more in-depth reading taking place after the lecture is finished. Students are encouraged to figure out which method works best with their particular style of learning. Whichever method the student decides to use to incorporate assigned readings into the learning process, it should be remembered that the lectures and readings are intended to supplement each other, not act in lieu of each other.

References Available for Student Use

Students may find the following documents helpful:

- *The Rational Unified Process, Version 2002.05.00.25*, IBM Corp, 2003.
  Available at L:\eng students\csce593\RationalUnifiedProcess\index.html.
- *OMG Unified Modeling Language Specification, Version 1.5,* Object Management Group, 2003.
  Available at L:\eng students\csce593\Papers\UML 1.5 Spec 03-03-01.pdf.

Instructor Contact Info

John Reisner
Office Hours by Appointment
Daytime Phone: 255-3636 x7422 (this is a WPAFB phone number)
email: **john.reisner@wright.edu** (it wouldn't hurt to cc: **john.reisner@afit.edu**)
→ or use WebCT email tool

The instructor is an adjunct faculty member. Most contact will be done via WebCT, or in after-class discussions. Other meetings can be arranged.

Course Objectives

At the conclusion of this course, each student should be able to

- Comprehend the advantages and importance of using a sound software engineering methodologies and processes for developing complex software systems.
- Explain the characteristics, strengths, and weaknesses of software process models, and how they evolved.
- Understand the benefits of use-case-driven design.
- Design a system structurally and behaviorally based on developed use cases.
- Apply object-oriented techniques and modeling with UML to the development of software systems.
- Identify and apply appropriate techniques and tools for use at each phase of the software lifecycle.

Grading

Course grades will be weighted as follows:

15% Homework Assignments

- Homework assignments are designed to facilitate deeper comprehension about a lecture topic (in other words, these are "think and respond" assignments).
- There may be up to two assignments per week, but some weeks may have one or zero assignments.
- Answers to these homework assignments generally run about half page to one page in length, and should not take too long to complete, although there may be some exceptions to this guideline. Full details about these assignments will be posted on WebCT.
- Homework assignments, unlike the group project work, are meant to be done *individually.*
- Assignments are due at the start of the class/lab session; please have them printed out and ready to turn in at the start of class. If you are unable to attend class, email will be accepted. Emailed assignments should be timestamped before class time (skipping class does not give you a homework extension).

30% Group Project

- Partnered with another student, you will perform a use-case driven design and refinement project.
- This project will begin during the first week of class, and last for the duration of the course.
- Students will submit their work at various stages of development in order to get some feedback.
- Grade for the project is based on the final turn-in. Modifications can be made to interim work before the final grade is assigned.

25% Midterm Exam

- Mixed-format exam, administered in class.

25% Final Exam

- Comprehensive, mixed-format exam, administered on the last day of classes.

5% Class Participation

- Much of the course lecture time is designed to be interactive and engaging. Students are expected to remain focused on the discussion topics, and contribute constructively. Those who do so will be rewarded with a high participation grade; those who merely "clock watch" or find other ways to pass time without contributing to the betterment of the learning environment will have this portion of their grade adjusted accordingly.

Final course grades will be assigned at the instructor's discretion, after all work has been graded, and the grade distribution has been analyzed. The follow scale is used, but this scale may be (and frequently is) curved:

92-100 = A
84-91 = B
75-83 = C
65-74 = D
< 65 = F

Class Project

The class project schedule is as follows:

Project Milestone	**Initially Due**
Requirements document & project plan	After Week 1
First 5 (or 6)* use-case scenarios	After Week 2
Remaining* use-case scenarios, plus use-case diagram	After Week 3
Use-case diagram, User interface design	After Week 4
Initial class diagram	After Week 5
First 4 sequence diagrams	After Week 6
Remaining sequence diagrams	After Week 7
Test Documentation	After Week 8
Final Report	After Week 9

**Project teams with only 460 students require 12 scenarios minimum. Project teams with 660 students require at least 14 or 16 scenarios, depending upon whether there are 1 or 2 660 students in the group.*

NOTE: Although materials are initially due on the dates indicated, your final project grade is based on the quality of the materials *on the day of the final turn-in.* When a real-world project gets off to a bad start, problems which are identified early in the lifecycle can be rectified, making the overall project a success. Similarly, students working on this class will have the opportunity to address any identified shortcomings without negative repercussions on their final grades.

Factors for the final project grade include quality, consistency, and completeness of the design, organization and readability of the documentation, and apposite leveraging of object-oriented concepts. More information and guidance concerning the class project will be discussed in class and distributed in various handouts.

Course Schedule

SCHEDULE IS SUBJECT TO CHANGE			
Week/Lesson	Date	Lesson Topics	Reading Assignment / Project Milestone Due
1 / 1	Mon	Course Introduction Overview of O-O Software Development Lifecycle Model Evolution	Due on Weds: Project Selection, Partnerships Text Readings: Chaps 1, 4 Due next week: Requirements Specification
1 / 2	Wed	Requirements Analysis	
2 / 3	Mon	Use-Case Driven Development Use Case Scenarios	Due on Mon: Requirements Specification Text Readings: Chap 5 Due next Mon: First use-case scenarios (5 or 6)
2 / 4	Wed	Use Cases: Business vs. System Case Use Case View & Diagrams	
3 / 5	Mon	Structural Modeling Objects and Classes Encapsulation Inheritance & Polymorphism	Due on Mon: First use-case scenarios Text Readings: Chaps 6, 11 Due next Mon: All use-cases & diagram (12-14 minimum)
3 / 6	Wed	Candidate Classes & CRC Cards Entity, Boundary, Control Objects User Interface Design	
4 / 7	Mon	Behavioral Modeling Sequence Diagrams	Due on Mon: All use-case scenarios Text Readings: Chap 7 Due next Mon: Use-case diagram & GUIs
4 / 8	Wed	Activity Diagrams State Chart Diagrams	
5 / 9	Mon	Overview & Review: OO Design Process & Modeling	Due on Mon: Use-case diagram; GUI design Text Readings: Chap 8, 9 Due next Mon: Initial Class Diagram
5 /-	Wed	NO LECTURE MIDTERM EXAM	
6 / 10	Mon	Software Testing The V-Model Unit & Acceptance Testing	Due on Mon: Initial Class Diagram Text Readings: Chap 13 Due next Mon: First half of seq. diagrams
6 / 11	Wed	Test Plan Development & Execution Regression Testing	
7 / -	Mon	In-Class Design Reviews (Pt. 1)	Due on Mon: First sequence diagrams Text Readings: None Due next Mon: Remaining seq. diagrams; Modified class diagram
7 / -	Wed	In-Class Design Reviews (Pt. 2)	
8 / 13	Mon	Software Project Management Work Breakdown Structures	Due on Mon: Extra Behavioral Diagrams Text Readings: Chaps 2, 3 Due next Mon: Test Plan
8 / 14	Wed	Development Strategies Project Risk & Risk Management	
9 / 15	Mon	Software Implementation Software Maintenance	Due on Mon: Test Plan Text Readings: Chaps 13, 14 Due next Mon: FINAL REPORT
9 / 16	Wed	Software Engineering Topics Configuration Management Ethics & Metrics	
10 / 17	Mon	The Big Picture: Software Horror Stories	Due on Mon: FINAL REPORT Text Readings: Chap Due next week: N/A
10 / -	Wed	NO LECTURE FINAL EXAM	