Department of Computer Science and Engineering
College of Engineering, University of Nevada, Reno

Fall 2007

CS 425/CS 625 Software Engineering

Lectures: MW, 4:00 – 5:15 pm, SEM 261

Instructors: Sergiu Dascalu
Room SEM-236
Tel: (775) 784-4613
dascalus@cse.unr.edu
www.cse.unr.edu/~dascalus

Office hours: MW 3:00 – 4:00 pm, or by appointment or chance

Catalog description: **CS 425/625 SOFTWARE ENGINEERING**
Lecture + Lab: 3 + 0; Credit(s): 3
Requirements specifications, structured analysis, modeling, top down design, testability, maintainability, portability, verification and validation, modification, configuration, management, reliability, efficiency, complexity, compatibility, modularity, interfacing, hardware and language issues. (Major capstone course.)
Pre-requisite: CS446.

Course outline: This course covers the software development process, from requirements elicitation and analysis, through specification and design, to implementation, integration, testing, and maintenance (evolution). A variety of concepts, principles, techniques, and tools are presented, encompassing topics such as software processes, project management, people management, software requirements, system models, architectural and detailed design, user interface design, programming practices, verification and validation, and software evolution. Although the emphasis will be on modern approaches some more traditional software engineering techniques will also be discussed.

Texts:

- [SE-8] Ian Sommerville, Software Engineering, 8th Edition, Addison-Wesley, 2006, ISBN: 0-321-31379-8
- Lecture notes: include presentations that will be made available by the instructors and notes that you will take during lectures
- Additional material as indicated later by the instructors

Initial web pointers:

- CS 425/625 course website:
  www.cse.unr.edu/~dascalus/SE2007.html
- Ian Sommerville's web-site for the textbook:
  http://www.cs.st-andrews.ac.uk/~ifs/Books/SE8/index.html [Note: 8th edition!]
- The Software Engineering Institute, at Carnegie Mellon University,
  www.sei.cmu.edu
- The Object Management Group:
  www.omg.com

Several other addresses of www sites that contain useful resources (technical documents, tools, etc.) will be indicated by the instructors during the semester.

Grading scheme: (tentative)

•	Assignments (individual)	A#1, A#2	12%
•	Project (team)	P#1, P#2, P#3, P#4, DEMO	38%
•	Midterm tests	T#1, T#2	20%
•	Final exam (comprehensive)	EXAM	24%
•	Class participation	CP	6%

Honors and *graduate* students are also required to complete a technical essay (TESS) worth 10% (see grading scales below). For them, there will also be at least an additional question in each test or final exam.

In order to pass the course you need to obtain at least 50% overall, at least 50% in tests (midterm tests + final exam), at least 50% in applications (project parts P#1, P#2 and P#3 + assignments A#1 and A#2 + class participation CP), and at least 50% in project implementation and demo (P#4 and DEMO).

To obtain grade A you need to obtain at least 90% overall and at least 90% in class participation. Significant lack of class participation can significantly affect your overall grade.

There are no make-ups for homework or tests in this course.

Grading scale [regular CS425]:

A	90 -100	[maximum 100]
A-	87 - 89	
B+	83 - 86	
B	78 - 82	
B-	75 - 77	
C+	71 - 74	
C	66 - 70	
C-	63 - 65	
D+	60 - 62	
D	55 - 59	
D-	50 - 54	
F	< 50	

Grading scale [CS625 and honors CS425]:

A	99 - 110	[maximum 110]
A-	95 - 99	
B+	91 - 94	
B	85 - 90	
B-	80 - 84	
C+	77 - 79	
C	71 - 76	
C-	67 - 70	
D+	64 - 66	
D	60 - 63	
D-	55 - 59	
F	< 55	

Late submissions: Late submissions of homework will be penalized with a deduction of 10% of the grade per late day, to a maximum of two late days for each submission. No material will be accepted after two days past the deadline. For example, an assignment that is worth 90/100 points will receive 90*0.9 = 81/100 points if it is one day late. The same assignment will receive 90*0.8 = 72/100 points if it is two late days and it will not be accepted if it is more than two days late. Note that late days are not divisible in subunits.

On plagiarism and cheating:

Plagiarism and cheating will not be tolerated. It will be dealt with according to the policies of the University of Nevada, Reno regarding academic dishonesty. Please read these policies at www.unr.edu/stsv/acdispol.html

Legal notices on the World Wide Web:

When accessing www resources such as downloadable software, technical reports, papers, on-line tutorials, etc., do not forget to read their accompanying legal notices and comply with their provisions.

Overall course objective:

Coverage of the phases of the software process through study of related concepts, principles and techniques as well as practical software development work using a systematic engineering approach.

Main directions:

- Comprehensive study of software engineering concepts, principles, and techniques
- Extensive coverage of the phases and activities of the software process
- Study of several advanced software engineering topics such as real-time software designs, agile methods, and critical systems
- Practical software development work within the framework of integrated development environments

Disability statement: If you have a disability for which you will need to request accommodations, please contact as soon as possible the instructors or the Disability Resource Center (Thompson Student Services - 107).

Tentative schedule:

Week	Dates (M, W)	Contents
1	Aug 27, 29	Lectures [Overview]
2	- , Sep 5	Lecture [Overview], A#1 given
3	Sep 10, 12	Lecture [Overview], Invited talk [IT]
4	Sep 17, 19	Lectures [Overview, Requirements], A#2 given *A#1 due*
5	Sep 24, 26	Lectures [Requirements], Project P#1 given
6	Oct 1, 3	Lecture [Requirements], Invited talk [IT] *A#2 due*
7	Oct 8, 10	Lectures [Design], Project P#2 given *Project P#1 due*
8	Oct 15, 17	Lecture [Design], Technical essay given [TESS] *Midterm T#1 (10/17)*
9	Oct 22, 24	Lectures [Design], Project P#3 given
10	Oct 29, 31	Lecture [Design], Invited talk [IT] *Project P#2 due*
11	Nov 5, 7	Lectures [Critical Systems]
12	Nov - , 14	Lecture [Verification & Validation], *Project P#3 due*
13	Nov 19, 21	Lecture [Verification & Validation], Project P#4 given *Midterm T#2 (11/19)*
14	Nov 26, 28	Lectures [Managing People]
15	Dec 3, 5	Lectures [Emerging Technologies] *Technical essay (TESS) due*
16	Dec 10, -	*Project P#4 due, Demo (12/10)* *Final EXAM (12/17)*

Note: In the above, Overview, Requirements, Design, Critical Systems, Verification and Validation, Managing People, and Emerging Technologies are the seven parts of the [SE-8] textbook.

Course Assessment Matrix
CS 425 Software Engineering

ABET Criterion 3 Outcomes	Course Outcomes	Assessment Methods/Metrics	CS Program Objectives Impacted	CIE Program Objectives Impacted
c	Students have the capability to specify and design a software system.	Define project concept, elaborate software requirements specification, perform use-case modeling, and develop high-level design, detailed design and user-interface design of the software system.	2, 3	2, 3
d	Students demonstrate the ability to use software engineering concepts, principles, techniques and tools while working in a project group.	Operate as a team to develop the project (a software system), acquire and use resources (references) pertaining to the project's application domain, and demonstrate the project's functionality.	2, 3	2, 3
e	Students have the ability to identify software development needs and/or challenges that require engineering solutions and formulate such solutions.	Define a project topic of good utility and/or interest in a specific area of human activity, assess challenges for developing the project, and outline possible design and implementation solutions.	3	3
f	Students have a thorough understanding of professional and ethical responsibilities	Discuss topics and answer questions regarding professional and ethical aspects of the software engineer's job and activities	4	4
g	Students are capable of expressing their project solutions in clear and precise ways.	Create project deliverables that include documentation and descriptions written in fluent, correct, clear and precise English.	3, 4	3, 4
k	Students demonstrate the ability to apply a range of techniques and tools for developing software systems.	Study, describe and apply software engineering techniques and tools associated with the various phases and activities of the software process: requirements engineering, software design, implementation, integration and testing.	2, 3	2, 3

ABET Criterion 3 Outcomes:

a. an ability to apply knowledge of mathematics, science, and engineering
b. an ability to design and conduct experiments, as well as to analyze and interpret data
c. an ability to design a system, component, or process to meet desired needs
d. an ability to function on multi-disciplinary teams
e. an ability to identify, formulate, and solve engineering problems
f. an understanding of professional and ethical responsibility
g. an ability to communicate effectively
h. the broad education necessary to understand the impact of engineering solutions in a global and societal context
i. a recognition of the need for, and an ability to engage in life-long learning
j. a knowledge of contemporary issues
k. an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice

Computer Science Program Objectives:

Our graduates will have achieved:

1. a broad general education assuring an adequate foundation in science and mathematics relevant to computing.
2. a solid understanding of concepts fundamental to the discipline of computer science.
3. good analytic, design, and implementation skills required to formulate and solve computing problems.
4. the ability to function, communicate, and continue to learn effectively as ethically and socially responsible computer science professionals.

Computer and Information Engineering Program Objectives:

Within 3 to 5 years of graduation our graduates will:

1. be employed as computer engineering professionals beyond entry level positions or be making satisfactory progress in graduate programs.
2. have peer-recognized expertise together with the ability to articulate that expertise as computer engineering professionals.
3. apply good analytic, design, and implementation skills required to formulate and solve computer engineering problems.
4. demonstrate that they can function, communicate, collaborate and continue to learn effectively as ethically and socially responsible computer engineering professionals.