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

Fall 2009

CS 425 Software Engineering

Lectures: MW, 1:00 – 2:15 pm, WRB-2008

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

Office hours: W 5:30 – 7:00 pm, or by appointment or chance

Catalog description:
CS 425 SOFTWARE ENGINEERING
Lecture + Lab: 3 + 0; Credit(s): 3
Software processes, project management, software requirements, system models, architectural design, detailed design, user interface design, implementation, integration, verification, validation, testing, evolution, rapid development, software tools (Major capstone course)
Pre-requisites: CS 446; ENG 102; CH 201; Junior or senior standing

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 course website: www.cse.unr.edu/~dascalus/se2009.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	40%
• Midterm test	T#1	14%
• Final exam (comprehensive)	EXAM	28%
• Class participation	CP	6%

Honors students are also required to complete a technical essay (TESS) worth 10% (see grading scales below).

In order to pass the course you need to obtain at least 50% overall, at least 50% in tests (midterm test + 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. Poor class participation can significantly affect your overall grade.

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

Grading scale [regular CS 425]:

A	90 -100	[maximum 100]
A-	86 - 89	
B+	83 - 85	
B	77 - 82	
B-	73 - 76	
C+	70 - 72	
C	64 - 69	
C-	61 - 63	
D+	58 - 60	
D	54 - 57	
D-	50 - 53	
F	< 50	

Grading scale [Honors CS 425]:

A	99 - 110	[maximum 110]
A-	95 - 98	
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. 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:

- 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 24, 26	Lectures [Overview]
2	Aug 31, Sep 2	Lectures [Overview], A#1 given
3	- , Sep 9	Lecture [Overview]
4	Sep 14, 16	Lectures [Requirements], A#2 given *A#1 due*
5	Sep 21, 23	Lectures [Requirements], Invited talk [IT]
6	Sep 28, 30	Lecture [Requirements, Analysis], Project P#1 given *A#2 due*
7	Oct 5, 7	Lectures [Design], Technical essay given [TESS]
8	Oct 12, 14	Lectures [Design],], Project P#2 given *Project P#1 due*
9	Oct 19, 21	Lectures [Design] *Midterm (10/21)*
10	Oct 26, 28	Lectures [Design, Development], Project P#3 given *Project P#2 due*
11	Nov 2, 4	Lecture [Development]
12	Nov 9, -	Lecture [Verification & Validation], Project P#4 given *Project P#3 due*
13	Nov 16, 18	Lecture [Verification & Validation], Invited talk [IT]
14	Nov 23, 25	Lectures [Managing People]
15	Nov 30, Dec 2	Lectures [Emerging Technologies] *Technical essay (TESS) due*
16	Dec 7, -	*Project P#4 due, Demo (12/7 and 12/8)* *Final EXAM (12/14)*

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

CS and CIE Program Outcomes	Course Outcomes	Assessment Methods/Metrics	CS and CIE Program Objectives Impacted
3	Students have the capability to specify, design and implement a software system that meets specific needs within realistic constraints.	Define project concept, elaborate software requirements specification, develop design models, implement, and test the software system.	2, 3
4	Students demonstrate the ability to develop a high quality software system while working in a project group.	Operate as a team to develop the project, acquire and use resources (references) pertaining to the project's application domain, and demonstrate the project's functionality.	2, 3, 4
5	Students have the ability to identify software development needs and challenges that require various 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.	1, 3
6	Students have a thorough understanding of professional, ethical and social responsibilities	Discuss topics and answer questions regarding professional, ethical and social aspects of the software engineer's job and activities.	4
11	Students are capable of developing their project solutions using modern engineering techniques and tools.	Use modern software engineering techniques and tools associated with the various phases and activities of the software process: requirements engineering, analysis, design, implementation, and testing.	1, 2
13	Students demonstrate the ability to apply a range of design and development principles in the construction of a software system.	Study and apply various high level and detailed design and implementation principles for building a software system.	2, 3

CS and CIE Program Outcomes:

Outcome	Description of Outcome
1	an ability to apply knowledge of computing, mathematics, science, and engineering
2	an ability to design and conduct experiments, as well as to analyze and interpret data
3	an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs, within realistic constraints specific to the field
4	an ability to function effectively on multi-disciplinary teams
5	an ability to analyze a problem, and identify, formulate and use the appropriate computing and engineering requirements for obtaining its solution
6	an understanding of professional, ethical, legal, security and social issues and responsibilities
7	an ability to communicate effectively with a range of audiences
8	the broad education necessary to analyze the local and global impact of computing and engineering solutions on individuals, organizations, and society
9	a recognition of the need for, and an ability to engage in continuing professional development and life-long learning
10	a knowledge of contemporary issues
11	an ability to use current techniques, skills, and tools necessary for computing and engineering practice
12	an ability to apply mathematical foundations, algorithmic principles, and computer science and engineering theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices
13	an ability to apply design and development principles in the construction of software systems or computer systems of varying complexity

CS Program Objectives: Within 3 to 5 years of graduation our graduates will:

1. be employed as computer science 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 science professionals.
3. apply good analytic, design, and implementation skills required to formulate and solve computer science problems.
4. demonstrate that they can function, communicate, collaborate and continue to learn effectively as ethically and socially responsible computer science professionals.

CIE 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.