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

Fall 2015

CS 425 Software Engineering

Course Syllabus

Lectures: TR, 9:30 – 10:45 am, LEG-212

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

Teaching Assistant: Ben Brown
Room: ECC
abenyoucantrust@gmail.com

Office hours: **Dascalu:** R 4:00 – 5:00 pm **Brown:** M 10:00 – 10:45 am; W 1:00 – 1:45 pm

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: CS446; ENG102; CH201or 202 or 203; 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-10] Ian Sommerville, Software Engineering, 10th Edition, Addison-Wesley, 2015, ISBN: 0-13-394303-8
- 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/se2015.html
- Ian Sommerville's web-site for the textbook: http://iansommerville.com/software-engineering-book/ [Note: 10th edition]
- The Software Engineering Institute, at Carnegie Mellon University, www.sei.cmu.edu
- The Object Management Group: www.omg.org

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, A#3	15%
• Project (team)	P#1, P#2, P#3, P#4	38%
• Midterm test	T#1	15%
• Final exam (comprehensive)	EXAM	27%
• Class participation	CP	5%

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

A	90 -100	[maximum 100]
A-	87 - 89	
B+	83 - 86	
B	78 - 82	
B-	74 - 77	
C+	70 - 73	
C	65 - 69	
C-	61 - 64	
D+	57 - 60	
D	54 - 56	
D-	50 - 53	
F	< 50	

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 software reuse, component-based software engineering, and service-oriented architecture
- 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).

Academic success services:

Your student fees cover usage of the Math Center (784-4433 or www.unr.edu/mathcenter/), Tutoring Center (784-6801 or www.unr.edu/tutoring/), and University Writing Center (784-6030 or http://www.unr.edu/writing_center/. These centers support your classroom learning; it is your responsibility to take advantage of their services. Keep in mind that seeking help outside of class is the sign of a responsible and successful student.

Statement on audio and video recording:

Surreptitious or covert video-taping of class or unauthorized audio recording of class is prohibited by law and by Board of Regents policy. This class may be videotaped or audio recorded only with the written permission of the instructor. In order to accommodate students with disabilities, some students may be given permission to record class lectures and discussions. Therefore, students should understand that their comments during class may be recorded.

Tentative schedule

Week	Dates (M, W)	Contents
1	Aug 25, 27	Lectures [Introduction]
2	Sep 01, 03	Lectures, Invited talks, A#1 given
3	Sep 08, 10	Lecture, Invited talks, A#2 given *A#1 due*
4	Sep 15, 17	Lectures, Invited talks, A#3 given *A#2 due*
5	Sep 22, 24	Lectures, P#1 given
6	Sep 29, Oct 01	Project meetings *A#3 due*
7	Oct 06, 08	Project meetings, Lecture, Project P#2 given *P#1 due*
8	Oct 13, 15	Lectures
9	Oct 20, 22	Lectures, Project P#3 given *P#2 due*
10	Oct 27, 29	*Midterm [10/27]* Lecture
11	Nov 03, 05	Lectures, Project P#4 given
12	Nov 10, 12	Lectures *P#3 due*
13	Nov 17, 19	Lectures
14	Nov 24, -	Lecture
15	Dec 01, 03	Lectures
16	Dec 08, -	*P#4 due, Demos (12/07-12/09)* *Final EXAM*

Course Assessment Matrix

CS 425 Software Engineering

Program Outcomes	Course Outcomes	Assessment Methods/Metrics	Program Objectives Impacted
4	Students demonstrate the ability to develop a high quality software system while working in a project group.	Operate in teams 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 practical 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 pertaining to professional, ethical and social aspects of the software engineer's job and activities.	4
8	Students are able to analyze the impact of computing and engineering solutions on individuals, organizations, and society	Explore topics relevant to the local and global impact of computing and engineering solutions on individuals, organizations, and the society.	
11	Students are capable to develop their software projects 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

CSE Student 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

CSE Program Educational Objectives

Within 3 to 5 years of graduation our graduates will:

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