CS 310: SOFTWARE ENGINEERING

SYLLABUS, SPRING 2015

COURSE DESCRIPTION
This course provides an introduction to the software engineering process and software lifecycle. The topics covered include project management, requirements, architecture, design, implementation, testing, and maintenance phase activities.

PREREQUISITE
CSCI 201: Principles of Software Development

COURSE INFORMATION
Time: Mondays and Wednesdays, 10:00pm-11:50pm
Location: SSL 150
Website: Via USC Blackboard
Final exam: Monday, May 11th, 8-10am

INSTRUCTOR
Name: William G.J. Halfond
Office: SAL 330
Office hours: TBD
Telephone: (213) 740-1239
Email: halfond@usc.edu [Preferred]

TEACHING ASSISTANTS
Name: TBD
Office: TBD
Office hours: TBD
Email: TBD

MOTIVATION
Software engineering is the systematic study of processes and techniques for building adaptable and reliable software systems. This is an important engineering discipline because software supports almost all of our economic and social infrastructure, such as banking, transportation, and communication systems. Millions of users interact with software on a daily basis whether it is through their mobile devices, automobiles, or via the Internet. Software developers face an incredibly hard task in ensuring that not only do they build reliable software systems, but that they
build systems that will stand the test of time as new features and capabilities are added to evolve the system.

Despite the best efforts of software engineers, the discipline faces many challenges. The National Institute of Science and Technology (NIST) has estimated that software errors and failures are so prevalent that they cost the US economy almost $59.6 billion annually. The NIST estimates that over a third of that amount, $22.2 billion, could be saved by improved software engineering infrastructure that enables earlier and more effective elimination of defects in the software development lifecycle.

**COURSE OVERVIEW**

This class will cover the key concepts and best practices of the software engineering discipline. Students will learn about the different phases of the classic software engineering lifecycle and the activities that software engineers perform during each of these phases. This will include project management, software requirements specification, architecture, design, implementation best practices, software testing, and maintenance activities.

Students will also participate in two team-based software engineering projects that will span the entire software lifecycle. In the first project, students will use a classic software lifecycle model to learn about each of the phases as they are applied in practice. For the second project, students will learn and apply agile software developments techniques and go through multiple iterations of the agile software lifecycle.

**TOPICS AND SKILLS**

- Understanding of the process of software engineering
- Ability to distinguish software lifecycle activities
- Understanding of the role of each phase’s activities in ensuring quality software
- Familiarity with the tools and mechanisms used to achieve each phase’s goals
- Ability to plan and carry out a team-based software engineering project using traditional and agile development methodologies
- Practical and conceptual knowledge of:
  - Project management plans
  - Software specifications and requirements
  - Software architecture and design
  - Common implementation and configuration management strategies
  - Test suite development, maintenance, and measurement
  - Software maintenance activities

**READINGS**

The following textbook is required. Additional handouts for special topics will be made available as indicated.
### SCHEDULE

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Project Milestone</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Course overview and introduction to SE</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Project management and overview of class project</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Requirements, modeling, and architecture</td>
<td>Kickoff 1</td>
<td>9,10</td>
</tr>
<tr>
<td>4</td>
<td>Software design principles and styles</td>
<td>Plan 1</td>
<td>11,12</td>
</tr>
<tr>
<td>5</td>
<td>Implementation practices</td>
<td>Requirements 1</td>
<td>4,15</td>
</tr>
<tr>
<td>6</td>
<td>Verification and validation techniques</td>
<td></td>
<td>6,13</td>
</tr>
<tr>
<td>7</td>
<td>Software delivery and alternate lifecycles models</td>
<td>Design 1</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>Midterm and review</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>Agile development practices</td>
<td>Implementation 1</td>
<td>Handouts</td>
</tr>
<tr>
<td>10</td>
<td>Software maintenance activities</td>
<td>Testing 1</td>
<td>14</td>
</tr>
<tr>
<td>11</td>
<td>Cleanroom development practices and code reviews</td>
<td>Kickoff 2</td>
<td>Handouts</td>
</tr>
<tr>
<td>12</td>
<td>Software economics, modeling, and certifications</td>
<td>Iteration 1</td>
<td>7, Handouts</td>
</tr>
<tr>
<td>13</td>
<td>Regression testing and formal verification</td>
<td>Iteration 2</td>
<td>Handouts</td>
</tr>
</tbody>
</table>
The grade for this class will be comprised of two projects and two exams. The breakdown for each of these categories is listed below. A more detailed explanation of the grading for each category is also provided.

- Midterm Exam: 20%
- Project 1: 25%
- Project 2: 25%
- Final Exam 30%

**CLASS PROJECTS**
Both projects will have teams work through the software lifecycle from project planning to software delivery. The first project will use a traditional waterfall process for sequencing the lifecycle activities. Deliverables will be due at the end of each lifecycle phase. The second project will use a more modern and popular software lifecycle process based on agile development techniques. Deliverables will be due at the end of each iteration. The project requirements will be negotiated with each team, but will focus on the development of a small client-server web or mobile application.

Grading for the projects will be as follows: For both projects, there will be two submissions for each deliverable. Students will receive feedback on their first submission, and the grade will count for 2/3 of the final grade of the deliverable. Students will have an opportunity to revise their submission, and the grade for the revised version will count for 1/3 of the final grade of the deliverable. After the final deliverable for each project, students will also submit a written evaluation of their teammates, which will be used to weight the final grades assigned to each student for the project.

**EXAMS**
There will be two exams, a midterm and final exam. The midterm will cover all material presented to date in the class. The final exam will be cumulative over the entire semester.

**LATE POLICY**
Late work will not be accepted without prior approval of the instructor.

**ACADEMIC CONDUCT**
Plagiarism – presenting someone else’s ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in SCampus in Section 11, Behavior Violating University Standards https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriate-sanctions. Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct.

Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the Office of Equity and Diversity http://equity.usc.edu or to the Department of Public Safety http://capsnet.usc.edu/department/department-public-safety/online-forms/contact-us. This is important for the safety of the whole USC community. Another member of the university community – such as a friend, classmate, advisor, or faculty member – can help initiate the report, or can initiate the report on behalf of another person. The Center for Women and Men http://www.usc.edu/student-affairs/cwm/ provides 24/7 confidential support, and the sexual assault resource center webpage http://sarc.usc.edu describes reporting options and other resources.

SUPPORT SYSTEMS
A number of USC’s schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the American Language Institute http://dornsife.usc.edu/ali, which sponsors courses and workshops specifically for international graduate students. The Office of Disability Services and Programs http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, USC Emergency Information http://emergency.usc.edu will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.