Software Engineering I

CS 4320/7320 – Fall 2013

Introduction

This course will provide in-depth coverage of software engineering. As a writing intensive course, many of the exercises and analysis of software requirements and specifications will be writing assignments. The concepts presented are applicable to all types of software systems. Additionally, the course will develop the student's skills of interpersonal communication, team collaboration, project management, requirements analysis and research. After successfully completing this course, you will be able to apply fundamental software engineering concepts including analyzing, specifying, and designing large-scale systems; system implementation, testing and deployment; and operational system configuration management and maintenance.

Graduate students enrolled in the course will be perform additional literature review and writing exercises; typically reviews or critiques of scholarly software engineering articles.

Topics

1. Overview of software engineering, systems engineering, systems development life cycle
2. Requirements Analysis, Modeling Requirements and Systems specifications
3. Project Management: Collaborative development, Software economics
4. Critical Systems & Software Processes
5. Usability Issues in design and testing
6. Traditional and Object Oriented Modeling of software systems
7. Design: User goals, architectural, interfaces, controls
8. Software Development and Testing, Implementation and Support
9. Interactive Design Issues
10. Security and Ethical Issues

Writing Support

Writing laboratory personnel are available in the Student Success Center. Special appointments to work individually are available at website: *https://writery.missouri.edu/* or call 882-2496 to schedule an appointment or use the "Online Writery" for individual help. There is the Tutorial Reference Service in Brady Commons for one-on-one tutoring.

Group Work

One of the stated purposes of this course in the computer science curriculum is to introduce students to the management of group work. As a consequence of what normally happens in a business setting, students will be assigned to a team, following an application process. The team will work on one long project throughout the semester. The team management and your performance as a team member will be evaluated at least twice during the semester. A sheet with teamwork criteria will be posted on Blackboard in the project folder so that you will know on what you will be graded for both your self-evaluation and the team evaluation. These evaluation scores will be a part your overall grade for the course. These evaluations may be submitted electronically to maintain confidentiality.

Instructor

Dr. Grant Scott
Email: GrantScott@missouri.edu
Phone: 884-6400
Office Hours: *By Appointment*

Meeting Schedule

Tuesday and Thursday
2:00 – 3:15 PM
Neff Hall, rm 204

Grading

1. 10% Class Participation
2. 40% Writing Assignments
3. 30% Group Project Work
4. 10% Midterm Exam
5. 10% Final Exam

Grades will be +/- for both Undergraduates and Graduate Students

Course Prerequisites

CS 2050 & experience with C programming language, CS 3380 concurrently

Course Topics Resources

It is expected that students are proficient with performing basic web-based and Engineering Library (http://mulibraries.missouri.edu/engineering/) searches for information. Throughout the semester, students will be assigned reading from the IEEE Computer Society's *Software Engineering Body of Knowledge*: http://www.computer.org/portal/web/swebok

Additionally, IEEE, ACM and other open source papers providing thorough coverage of Software Engineering and the state of best practices will be assigned reading throughout the semester.

Course Learning Objectives from Departmental Curriculum Guidelines:

- Students will learn of ethical and professional issues related to software engineering
- Students will be able to analyze, interpret and describe the process of software engineering with respect to both large and small information systems, software systems, software development life cycles
- Students will demonstrate the ability to function effectively on a team.
- Students will work with a team of peers to analyze problems associated with a software solution, identify and define the requirements documents and models needed to determine possible solutions for the problem, and determine software and user-interface design.
- Students will revise and rewrite documents associated with technical writing (proposals, memos, business letters, executive summaries and project reports) based on peer and instructor editorial analysis resulting in precise and concise technical documents.
- Students will refine presentation skills to prepare for the capstone sequence resulting in professional presentations.
- Students will be familiar with object oriented analysis and design issues using Unified Modeling Language (UML).
- Students will be introduced to the preparation of a detailed problem definition incorporating needs analysis, market assessment, and scope and feasibility analysis for a desirable solution.
- Students will be encouraged to use a variety of tools in order to complete a project, managing the development, creating the design, and developing a solution.

Recommended Text Resources

Software Engineering :

Sommerville, Ian. *Software Engineering, 9th Edition.* Harrow, England: Addison-Wesley Publishing, 2011. ISBN: 978-0-13-703515-1; editions 7th or 8th are fine.

Other Software Engineering Texts : Some of these texts will be on reserve in the Engineering Library in Lafferre Hall. Ask for this class when you wish to find software engineering texts.

Van Vliet, Hans. *Software Engineering: Principles and Practice. 3rd Edition.* Chicester, England: John Wiley & Sons, Ltd, 2008. ISBN: 978-0-470-03146-9 E-version available through Merlin.

Pfleeger, Shari and Atlee, Joanne. *Software Engineering: Theory and Practice, 3rd Edition.* Upper Saddle River, NJ: Prentice Hall, 2006 (2001, 1998). ISBN: 0-131-46913-4 {any edition is fine}

Schach, Stephen. *Object-Oriented & Classical Software Engineering, 7th Edition.* Boston: McGraw-Hill Higher Education, 2007. ISBN: 0-07-319126-4. {new edition out – any edition is fine}

Pressman, Roger. *Software Engineering: A Practitioner's Approach, 6th Edition.* Boston: McGraw-Hill Higher Education, 2005. ISBN: 0-07-337597-7{Last few editions are fine}

Braude, Eric and Bernstein, Michael. *Software Engineering: Modern Approaches, 2nd Edition.* Hoboken, JN: John Wiley & Sons, Inc., 2011. (2001). ISBN: 978-0-471-69208-9 – features Java code in exercise applications

Agarwal, B.B., M. Gupta and S.P. Tayal *Software Engineering & Testing.* Sudbury, MA: Jones & Barlett Publishers, 2010. ISBN: 978-1-934015-55-1. E-version available through Merlin.

Mohapatra, Pratap K.J. *Software Engineering (A Lifecycle Approach).* Daryaganj, Delhi, India, 2010. ISBN: 9788122427219. E-version available through Merlin.

Unified Modeling Language (UML) :

Fowler, Martin. *UML Distilled*. Third Edition: A Brief Guide to the Standard Object Modeling Language. Boston, MA: Addison-Wesley, 2004. ISBN: 0-321-19368-7

Technical Writing and Style :

Strunk, W & E.B. White. *The Elements of Style*. NY: Allyn & Bacon Publishing, 1999. ISBN: 0-205-30902-X. [or latest version]

Haile, J.M.. *Technical Style: Technical Writing in a Digital Age*. Central, SC: Macatea Productions, 2001. ISBN: 0-9715418-0-9

Liebowitz, Jay, William Agresti and G. Reza Djavanshir. *Communicating As IT Professionals*. Upper Saddle River, NY: Pearson Prentice Hall, 2006. ISBN: 0-13-146828-6

Departmental Curriculum Guidelines : Course Learning Objectives

1. [Writing Intensive] Students will revise and rewrite documents associated with technical writing (proposals, memos, business letters, executive summaries and project reports) based on peer and instructor editorial analysis resulting in precise and concise technical documents.
2. Students will be able to analyze, interpret and describe the process of software engineering with respect to both large and small information systems, software systems, software development

life cycles

3. Students will be introduced to the preparation of a detailed problem definition incorporating needs analysis, market assessment, and scope and feasibility analysis for a desirable solution.
4. Students will be encouraged to use a variety of tools in order to complete a project, managing the development, creating the design, and developing a solution.
5. Students will demonstrate the ability to function effectively on a team.
6. Students will work with a team of peers to analyze problems associated with a software solution, identify and define the requirements documents and models needed to determine possible solutions for the problem, and determine software and user-interface design.
7. Students will refine presentation skills to prepare for the capstone sequence resulting in professional presentations.
8. Students will be familiar with object oriented analysis and design issues using Unified Modeling Language (UML).
9. Students will learn of ethical and professional issues related to software engineering

Writing Intensive

This course is to be designated as **Writing Intensive**. You will develop and demonstrate your understanding of software engineering and the development cycle through revisions of your ideas and written work. In support of rethinking, restructuring, and revising your work, you will receive specific individual feedback from your peers and your instructors. Four primary writing assignments will be integrated with the topics of the course:

1. Research and develop a technical/business requirements analysis document based on a current software engineering problem.
2. Development of a software project proposal and feasibility analysis document to empower management to make informed decisions regarding the proposed project.
3. Develop software design models with detailed narrative explanations of the interdependencies of the various system modules.
4. Write detailed software documentation and/or user guides for the resulting software.

(The end of the semester will include group-based oral presentations)

By applying critical thinking skills to construct and revise technical writing drafts you will strengthen and reinforce your conceptual understanding of the course topics. Conceptualization of the writing will be done in small group settings, e.g., your development team. Drafts will be written by each individual student, then peer-reviewed by the small group and eventually reviewed and graded by the instructors. The extensive technical writing will allow you to improve your written communication skills, which are necessary for all future career endeavors.

This portion of the course grade will determined by your ability to compose clear, concise, technical writing; and deliver professional quality reports and presentations. These skills are critical in real-world software engineering settings, where your professional writing and presentations reflect upon you and your organization. The instructors will utilize the point of view of clients and users of your software projects. As such, your technical writing should be professionally styled as it would for a future business employer or academic mentor.

Academic Dishonesty

Academic integrity is fundamental to the activities and principles of a university. All members of the academic community must be confident that each person's work has been responsibly and honorably acquired, developed, and presented. Any effort to gain an advantage not given to all students is

dishonest whether or not the effort is successful. The academic community regards breaches of the academic integrity rules as extremely serious matters. Sanctions for such a breach may include academic sanctions from the instructor, including failing the course for any violation, to disciplinary sanctions ranging from probation to expulsion. When in doubt about plagiarism, paraphrasing, quoting, collaboration, or any other form of cheating, consult the course instructor.

ADA Notice

Students with Disabilities:

If you anticipate barriers related to the format or requirements of this course, if you have emergency medical information to share with me, or if you need to make arrangements in case the building must be evacuated, please let me know as soon as possible. If disability related accommodations are necessary (for example, a note taker, extended time on exams, captioning), please register with the Office of Disability Services (http://disabilityservices.missouri.edu), S5 Memorial Union, 882-4696, and then notify me of your eligibility for reasonable accommodations. For other MU resources for students with disabilities, click on "Disability Resources" on the MU homepage.

Intellectual Pluralism

The University community welcomes intellectual diversity and respects student rights. Students who have questions or concerns regarding the atmosphere in this class (including respect for diverse opinions) may contact the Departmental Chair or Divisional Director; the Director of the Office of Students Rights and Responsibilities (http://osrr.missouri.edu/); or the MU Equity Office (http://equity.missouri.edu/), or by email at equity@missouri.edu. All students will have the opportunity to submit an anonymous evaluation of the instructor(s) at the end of the course.

Recording Course Activities

University of Missouri System Executive Order No. 38 lays out principles regarding the sanctity of classroom discussions at the university. The policy is described fully in Section 200.015 of the Collected Rules and Regulations. In this class, students may make audio or video recordings of course activity unless specifically prohibited by the faculty member. However, the redistribution of audio or video recordings of statements or comments from the course to individuals who are not students in the course is prohibited without the express permission of the faculty member and of any students who are recorded. Students found to have violated this policy are subject to discipline in accordance with provisions of Section 200.020 of the Collected Rules and Regulations of the University of Missouri pertaining to student conduct matters.

Graduate Student / Credit (CS 7320)

Students taking Software Engineering for graduate credit (CS 7320) will complete all the same work as the undergraduate students. Additionally, graduate students will be assigned additional reading assignments related to position papers and emerging research in the field of Software Engineering. You will write additional papers that are either multi-article compare and contrast reviews or single article critiques. These writings will be counted as part of the 40% course grade that is allocated to *Writing Assignments*.