

Course Number: CSC 648/848

Course Title: SOFTWARE ENGINEERING/ADVANCED SOFTWARE ENGINEERING

Number of Credits: 3

Schedule: Three hours of lecture/discussion per week.

Prerequisite: a grade of C or better in CSC 413 and in CSC 317, or consent of the instructor

Catalog Description

Practical methods and tools for SW Engineering, including organizational teamwork. Course is paired with CSC 848. Students who have completed CSC 848 may not take CSC 648 for credit.

Expanded Description

- Introduction and motivation for Software Engineering(SE)
- Overview of several basic SE methodologies with emphasis on Iterative and Incremental Development and User Centered Design
- Usability and UI design principles and practice
- Usability evaluation
- Basic components of SW Engineering process: Planning; Requirements and Specifications; Iterative Design, Rapid Prototyping, Mockups; Software Design; Coding and documentation techniques (high level only);
- SW Engineering related to Web application development
- Open source SW development and management
- Software Configuration Management, Code management, Delivery, Installation, and Documentation
- Software Metrics, and Usability Measurements and evaluation
- Software QA and Testing
- Software Maintenance
- Project Management issues
- Teamwork and Communication as integral part of SW Engineering
- Issues related to global SW engineering
- Real life examples and cases from instructor and students
- Guest and student presentations
- Final Team Project including several milestones, interaction with instructor, and final demo and delivery.

Course Objectives and Role in Program

This is a capstone course and paired class with graduate version CSC 848 where students integrate and apply all the knowledge they attained in the program. It offers comprehensive and advanced coverage of practical team oriented methods and tools of full lifecycle SW development. Special emphasis will be on Iterative, Incremental, Agile and User Centered Design team practices and on global SW engineering. Cornerstone of the class is student

team project (in teams of 6-7 students) where students engage in a team project in order to experience and practice key aspects of full SW engineering lifecycle in setting that simulates small SW company. Class teaching is coordinated with student team project needs.

Learning Outcomes

- Knowledge of modern Agile SE engineering methods/processes and best practices for team-centric SW development, and their appropriate application
- Knowledge of basic components and tools for team-centric full lifecycle SW development from use cases and initial specifications to design, implementation, testing, evaluation as well as code management and deployment on cloud systems
- Knowledge of best practices for the design, implementation and evaluation for delivering easy to use and maintainable SW
- Ability to constantly iterate and re-prioritize plans and designs based on user needs, budget, schedule and resources
- Development of significant SE teamwork and project based experience, as close as possible to real life

Suggested Method of Evaluation

<u>Final Exam</u>	40% (multiple choice questions testing basic knowledge of SW Engineering concepts and definitions)
<u>Final Team Project</u>	60% (Design and development of a SW application by student teams.)

Graduate students may have to do a class presentation in order to get graduate credit for 848.

Final Team Project

Objective: By working on developing a SW application in a team setting, students will experience and exercise all steps in full SW Engineering lifecycle simulating real SW company environment. All teams will work on the same application, usually a tree tier media rich WWW application which will be initiated by the instructor in the form of only a very brief description, in order to force the teams to engage in full cycle of SW development - from initial design to delivery.

Organization: Students will work in teams of 6-7. Teams will be selected by instructor based on mandatory student skills and experience survey (used only for team building) in order to build teams with approximately equal skills and experience profile. Each team will have team lead, front end lead, back end lead and github master. In some cases, if possible, we will simulate global SW engineering environment by including teams from other universities, currently from Fulda University in Germany. Instructors will serve as coaches, VP of Engineering, Marketing and Customer representative as well as CTO, and will attempt to create environment that simulates real life SW company. Interaction with instructor will evolve around key SW Engineering process milestones (e.g. infrastructure

selection and setting; high level design; detailed design (UI and system); implementation, testing and evaluation;, and final delivery) and will include extensive instructor feedback, walkthrough sessions and discussion, as well as usability feedback. Instructors will in general not get involved nor help in details of coding and implementation; this will be the responsibility of the students and the team.

SW tools and IT environment: Basic SW tools and environment will be provided using Open Source SW and cloud systems for deployment. Students will have some choice in deciding specific SW stack and cloud systems from a set recommended by instructors (first team milestone). All SW and tools as well as cloud servers for deployment will be open source and free

Deliverables: These include a) Final documents from all the required milestones from the team project (after instructors' feedback has been incorporated), b) code and code management repository (e.g. github; and c) demo of the final product to the whole class. These materials can also serve as student portfolio for job search

Grading Policy: All team members get the same grade except the team members who do not participate as required. Team grade is based on *SE Process* component (adherence to best practices of team-centric SW engineering) and *SW Product* component (quality of team SW deliverable). Grading of submitted documentation will be based only on final document versions; instructor's feedback during the process will not be graded but serve as help and guidance and encouragement for design iterations.

Full participation in all team activities and attendance of all team sessions is mandatory. Students are expected to read and respond as necessary to all class related e-mails from the instructor and their team members in a timely manner. Full student participation and engagement in team project will be checked regularly in team meetings with instructor, instructor's communication with team leads as well as by checking activity in code management systems like github. In case of inadequate student participation that student team grade (but not the team grade of other students in the team) shall be reduced for minimum 10%.

All other mandatory class policies (including those related pot on-line class delivery if applicable) will be introduced in the class in introductory slides.

Suggested (not mandatory) textbook

Ian Somerville: "Software Engineering", 10th edition, Addison Wesley ISBN-13: 978-0133943030

Modified by: D. Petkovic 08-18-20