

## CS Senior Seminar: Collaborative Software Development

*A proposal to shift the periodic Software Engineering special topics course to an annual Senior Seminar in Computer Science.*

The CS program proposes to take our old Software Engineering special topics course, revise it slightly, and turn it into a disciplinary senior seminar in CS that would be required for the major. There are several reasons to do this: 1) We have wanted for a long time to offer the Software Engineering course as a regular course offering so that it would be available to (or required of) all CS students; 2) From the time this course was first offered in the late '90's, it could have been considered a prototypical senior seminar, as it has a strong problem-solving focus, strong student participation and leadership, and stresses integration of the students' undergraduate experiences; 3) After several years of upheaval and changing personnel, we are working to stabilize, evolve, and revitalize the CS program.

**Integration:** The project and presentations that make up this course require students to draw on the skills and techniques they have acquired throughout their undergraduate career, not only from courses in the major, but often also from their study of other fields, internships, SIPs, and other outside activities. In addition, the course gives students a structured environment in which to practice and reflect on the critical life-long learning skills on which computer scientists depend, including learning new languages, methodologies, and frameworks. Thus, this course encourages integration of the student's past, present, and future, applying previous experiences to a current project that models future professional or research environments.

**Student Agency:** Students play a very strong leadership role in this class, including presentation of course material, gathering information about the course project(s), influencing the direction and progress of the project(s), and working collaboratively to achieve an outcome.

**Presentations:** Students present a significant percentage of the material covered in class, on topics such as traditional software engineering, agile development methods, design patterns, database concepts, web-based development languages and tools, version control, testing tools and techniques, open source practices, current trends, and professional ethics. These presentations involve reading, analyzing, synthesizing, and communicating information from books, articles, and online sources representing the (primarily professional) literature of software development.

**Project:** Each year the instructor chooses a topic and one or more possible project ideas. Within the constraints imposed by those initial decisions, student decisions within the class may affect the choice of project, development language, or design methodology, and students always lead the requirements gathering phase and project design for the project implementation.

**Communication Skills:** Students employ and further develop their communication skills through formal class presentations, working with a community representative to develop project requirements, teamwork throughout the quarter as part of project implementation,

presenting weekly oral status reports, developing internal and external documentation for their project, and writing a reflective paper on the process of software development, the responsibilities of developers, and the ways in which this course integrated their undergraduate experiences inside and outside the major.