Tailoring a Successful Project-Based Course -- in Which Students Learn to Work in Teams on the Development of Useful Software Products for Real Clients – to the Needs and Resource Constraints of Individual Colleges and Universities

Barry Boehm(Moderator) USC boehm@sunset.usc.edu
Dan Port USC dport@sunset.usc.edu
David Klappholz Stevens Institute of Technology david@cs.stevens-tech.edu

Keywords: project-based software engineering course, software projects, real clients, USC CS577, team projects

Workshop Summary

The purpose of this workshop is to present an in-depth overview of USC CS577, and to start the processes of:

- eliciting from faculty at a variety of colleges and universities the different needs and resource constraints to which they would be subject were they to port CS577 to their home institutions
- planning the types of adjustments that can be made to CS577 to meet various sets of constraints

The ultimate preparation for a career in software development is a project-based course in which students learn to work in teams on the development of useful software products for real clients. Published educational materials provide little support for the educator who wishes to teach such an ambitious course. We have embarked on a project whose goal is the development of materials which will enable educators to adapt Barry Boehm and Dan Port’s USC CS577, one of the most successful such courses, to their special needs and resources, and to disseminate these materials in the form of a textbook, student and instructor manuals, a web-based course delivery framework, videotapes of USC course lectures, a CD archive of completed student projects, and public training in both adaptation and teaching of the course.

Our starting point is the two-semester project-based Introduction to Software Engineering which Profs. Boehm and Port have developed at USC. The course’s effectiveness has been proved and improved over a period of 6 years with over 1000 students, and over 100 real-world customers.

Among the specific issues that will be discussed in the workshop are:

- acquisition of and dealing with clients for student projects -- or training TA’s, or oneself, to act as clients in cases in which the possibility of real clients is precluded.
- instructing students in the dynamics of dealing with clients/customers and other stakeholders to arrive at the specification of a software product which will meet stakeholders’ business, functional, performance, and other needs and to assess the direction of changes in needs after the product’s initial deployment
- instructing students in the preparation of the various documents and software artifacts
necessary for a team to successfully complete a software project and to transition the product to its end users and, ultimately, to those who will maintain and enhance it

- training students in preparing for and conducting various types of review (requirements, architecture/design, code, etc.) so as to detect and correct faults at each stage of the development process
- teaching students how to perform risk analysis, risk assessment, and risk mitigation in order to make choices which arise at each stage of the development process in such way as to maximize the probability of success
- helping students adjust and re-adjust their products’ features and details of the software development process over the project’s lifetime so as to maximize the probability of producing the most useful possible software product within the allocated time
- helping students identify and deal with problems in team/group dynamics
- selecting teaching assistants, if they are to be available, and training them in helping students with all of the above activities
- determining how to grade intermediate work products, including documents, reviews, and periodic progress reports, in a situation in which it is not the quality of the final software product, but rather, learning of the process which is the major criterion for success in the course
- helping students relate principles and lessons learned on team projects to software engineering best practices discussed in lectures and readings