A Combined Curriculum Research and Curriculum Development Approach to Software Engineering Education

Barry Boehm, Gail Kaiser, Daniel Port
USC, Columbia University
{dport, boehm}@sunset.usc.edu, kaiser@cs.columbia.edu

Abstract

Since 1996 the authors have had great success combining software engineering research and software engineering education. This effort introduces new research and experimental areas into the curriculum along with the means to evolve and refine current practices. The curriculum itself becomes a valuable research area with direct applications to “real-world” practice. The approach in particular tries to deal with four fundamental issues within software engineering and software engineering education. There are Exposing students to a “no-surprises” full lifecycle software engineering experience. Dealing with the increasing distribution and diversity of students. Providing students a non-trivial “real-world” customer-based development experience within the constraints of the university. Introducing software engineering research and industry best practices into the curriculum.

There are a myriad of challenges facing software engineering education. We believe some of the most critical are:

1) Exposing students to a “no-surprises” full lifecycle software engineering experience. The recent PITAC report on Information Technology Research emphasized the fragility of currently produced software, and identified software as the highest-priority area for increased IT research. The corresponding NSF Software Research Workshop Report emphasized the need for techniques that improve our ability to produce “no-surprise” software.

Some major sources of software surprises are the hidden conflicts among the models that the software system stakeholders (users, customers, developers, maintainers, marketers, and others) bring to a software project. Since 1996, USC has been using a combined research and team-project curriculum development (CRCD) approach to develop and refine a methodology called MBASE – Model-Based (System) Architecting and Software Engineering – to eliminate these conflicts or “model clashes.” This approach was introduced at Columbia in Spring 1999.

Current leading software development methods are particularly vulnerable to model clashes. Their product and process models focus almost exclusively on de novo logical product structuring, and do not deal with such property-model and success-model concerns as performance, cost, schedule, COTS selection and integration, and legacy systems. For example, in a recent survey of 16 object-oriented development books, we found only 6 with the term 'performance' in their index, and only two with the term 'cost'. Software engineering education must provide students an understanding of the theory and applications of product, process, property, and success models.

2) Dealing with the increasing distribution and diversity of students. University education has diversified greatly in the past few decades. Students may be part-time, off-campus, working professionals. Distance learning programs are a necessity. This presents a particular challenge to team oriented, software engineering course projects where collaboration is essential. As in industry, development may be critically hampered by the communications overhead in dealing with distributed stakeholders. Explicit structural guidance in stakeholder interaction such as that found in MBASE must be introduced into the core curriculum along with...
support from distributed collaboration tools such as offered through immersive virtual environments (CHIME), Internet-capable software development environments (Oz, OzWeb), and negotiation aids (WinWin, DCPT).

3) Providing students a non-trivial “real-world” customer-based development experience within the constraints of the university.
Real-client software development involves a number of built-in stakeholder model clashes. A key MBASE strategy to avoid model clashes on a software project is to promote collaboration toward a realistically negotiated critical stakeholder win-win (mutually satisfactory) project solution. Software engineering courses should avoid using pre-canned “artificial” projects that set up an overly simplified and ideal development scenario. Challenging students with an actual development project involving a multitude of stakeholders (in addition to the students and the instructors) provides a rich, meaningful experience that inevitably leads to greater value for all involved. At Columbia and USC we use a stakeholder win-win approach to address real-world software development challenges based on an analysis of the primary win conditions of such stakeholders as students, faculty, team-project clients, industry, and the overall community of software practitioners and users.

4) Introducing software engineering research and industry best practices into the curriculum. The gulf between software engineering research and practice is difficult to close. The negative effects of this cannot be understated. Students learn untested research techniques; industry best practices are kept proprietary, students are “re-educated” when hired into industry (and increasingly thought to be unprepared for industry development efforts), techniques are re-created over and over, cutting edge technologies are reluctantly applied, and software quality leaves users increasingly frustrated. Utilizing industry best practices such as UML modeling, architecture review boards, and risk-driven development provides students with a recognizable skill set. Exposing students to current research such as software architectures and formal methods provide them with increased analytical skills and competitive advantage while directly advancing software development state of the art (and perhaps spawning future software engineering researchers). Through MBASE, USC and Columbia use a combined Curriculum Research and Curriculum Development (CRCD) approach to software engineering education where research and the curriculum are intimately intertwined with direct influence from real-world customers from the university and industrial affiliates.