Chapter VII
The Software Enterprise: Preparing Industry-Ready Software Engineers

Kevin A. Gary
Arizona State University, USA

ABSTRACT

"You can't teach experience" – but you can sure try. At the Polytechnic Campus of Arizona State University, we are developing a learning-by-doing approach for teaching software engineering called the Software Enterprise. The Capstone experience is extended to two one-year projects and serves as the primary teaching and learning vehicle for best practices in software engineering. Several process features are introduced in an attempt to make projects, or more importantly the experience gained from project work, more applicable to industry expectations. At the conclusion of the Software Enterprise students have an applied understanding of how to leverage software process as a tool for successful project evolution. This chapter presents the Software Enterprise, focusing the presentation on three novel aspects: a highly iterative, learner-centered pedagogical model, cross-year mentoring, and multiple projects as a novel means of sequencing learning objectives.

INTRODUCTION

Students must emerge from a "write-a-program-get-a-grade" mentality to a "follow-a-process-produce-a-deliverable" mentality (and eventually to "use-and-improve-processes-to-solve-customer-problems"). This evolution from learner to practitioner is a cultural mindset even at the personal level. Junior professionals are confronted with real-world situations immediately after graduating and entering the workforce. Professionalism challenges junior engineers in a different way than academic ethics. Junior professionals can gain professionalism through formal and informal mentoring relationships in professional settings such as internships, but we should not rely solely

on industry to accept this burden; we must incorporate it into the learning environment.

The Software Enterprise, introduced four years ago in the Division of Computing Studies at Arizona State University's Polytechnic campus (ASU Poly), is our attempt at preparing new graduates for the software profession. In the model of a polytechnic, an increased emphasis is placed on hands-on practice over pure scientific study. The mechanism chosen for this approach is the Capstone project, which traditionally focuses on one or two semester projects required at the conclusion of the undergraduate degree program. The Capstone project, an inherited requirement from engineering disciplines, is often considered more a "rite of passage" than a teaching and learning opportunity. We contend the Capstone experience provides a great opportunity to be the primary teaching and learning model in software engineering. Our solution is a learn-by-doing model called the Software Enterprise.

The Software Enterprise is one part "evolution" and one part "revolution." It leverages some of the better practices we have seen from the multitude of Capstone software engineering projects published over the past decade. In particular, mentoring relationships within student teams are emphasized, as is a careful sequencing of course and project topics. The Software Enterprise also presents a novel pedagogical model geared to accelerate students' comprehension of software engineering. This combination of old and new is wrapped in an applied learning program so as to better prepare new graduates for the software engineering profession.

This chapter is organized as follows. The next section motivates the need for the Enterprise by discussing some perceived shortcomings of new computing graduates. The pedagogical innovation of the Enterprise is presented next, followed by a detailed description of how the pedagogy is implemented at ASU Poly. We conclude with an ongoing evaluation of the Enterprise and a summary.

BACKGROUND AND MOTIVATION

Software engineer ranks as one of the fastest growing occupations (U.S. BLS, 2007) with the highest median salary (Morsch, 2006). Unfortunately, many employers consider new graduates unproductive, while at the same time those graduates feel unprepared for that first job. Traditional computer science education is criticized as outdated, too theoretical, and too fractured. As educators, we should do a better job preparing new graduates for what lies ahead. We should expose students to the true nature of today's computing challenges, strive to ground students in fundamental theory, and provide them the modern tools a modern discipline requires.

The Software Enterprise uses a bottom-up approach to incorporating process best practices and process models via a multi-year Capstone experience. Example best practices include configuration management, unit testing, and code inspections for software development. By software process models we mean the incorporation of accepted process models as a mechanism for teaching and learning software construction, maintenance, and project management. The ability to identify issues, analyze risks, debate, create consensus, and work within a team are examples of managerial skills software engineers require perhaps more than other engineering disciplines due to the unique challenges in developing software products. We also contend there is more in the intersection of emphasizing process execution and project management skills than is given proper due. In other words, how does a learning facilitator demonstrate the need for process structure while at the same time mentor students on the judgment needed to know when to alter the current process instance to ensure project success?

The approach in the Software Enterprise is to provide a process structure, and then give teams "just enough rope" to resolve their own process-related issues. We do this in several ways. Traumatic "real-world events" are injected dur-