Chapter 4.10
Rapid Insertion of Leading Edge Industrial Strength Software into University Classrooms

Dick B. Simmons
Texas A&M University, USA

William Lively
Texas A&M University, USA

Chris Nelson
IBM Corporation, USA

Joseph E. Urban
Arizona State University, USA

ABSTRACT

Within the United States, the greatest job growth is in software engineering and information management. Open source software (OSS) is a major technology base for enterprise application development. The complexity of technologies used by industry is often an obstacle to their use in the classroom. In this chapter, a major software development paradigm change that occurred in about the year 2000 is explained. CS education programs have been slow to adapt to the paradigm change due to problems such as the tenure system, inexperienced student laboratory assistants, lack of leading-edge software tool support, lack of software team project servers, unavailability of help and mentoring services, and software unavailability. This chapter explains how these problems can be solved by creating an open source-based shared software infrastructure program (SSIP) sponsored by industry, but planned and implemented by SSIP member universities at no cost to member universities.

INTRODUCTION

Today students are saying no to computer science (Frauenheim, 2004). CS faculty members have panicked in what David Patterson (2005) calls Chicken Little rumor mongering. He tells everyone to stop whining about outsourcing. In our opinion, CS faculty should panic and adapt to a new software development paradigm. Pat-

terson makes an invalid implied assumption for his article in that CS in some way is related to information technology (IT) jobs in U.S. industry (or, for that matter, that CS is useful to a software engineer). He is correct to say that U.S. IT jobs are increasing. Also, software engineering (SE) degree programs and jobs are increasing. His domino theory of job migration is not correct. We agree with Patterson that U.S. programmers should worry about both India and China. We do not agree that either India or China will have to worry much about the Czech Republic. Both India and China have such large populations and low wages that major CS job migration will mainly be to these two countries. The middle processes of a software product development software life cycle (DSLC) may completely migrate from the United States.

Every Fortune 1000 company with which we are familiar takes advantage of low labor costs in India and/or China. Unfortunately for CS, approximately 80% of high-paying CS jobs in the past have been with Fortune 1000 companies. Jobs that will remain in the United States will go to students that are familiar with open standards, a wide variety of solutions including open source solutions, software development tools that support open standard visualization design models and open source integrated development environments. In this chapter, open standards will be defined as standards that are publicly available. The Object Management Group (OMG) (2006) is an example of an organization that was created to produce open standards. OMG is an open membership, not-for-profit consortium that produces and maintains computer industry open standards for interoperable enterprise applications. OMG membership includes virtually every large company in the computer industry and hundreds of smaller ones. OMG's most widely used standard is described by the unified modeling language (UML) specification. UML is used worldwide to model application structure, behavior, architecture, business process, and data structure. We use the term open source software (OSS) to refer to software that has Open Source Initiative (OSI) (2006) licenses. Examples of OSS are Linux, Apache, Eclipse, and Derby. We also include open-standard compliant software that is provided free for classroom use to universities. An example is IBM Rational Software Architect (RSA).

The objective of this chapter is to explain how leading-edge industrial-strength software can be introduced into the university classroom by using OSS, open standards, distance learning, and infrastructure shared among cooperating universities. In this chapter, we will describe the evolution of software development during the 20th century, the paradigm change at the beginning of the 21st century, and the problems with existing university information technology education. Then we will describe a shared software infrastructure program (SSIP) to rapidly introduce leading-edge industrial software solutions into university classrooms at no cost to SSIP member universities.

BACKGROUND

Software education emerged during the last 50 years of the 20th century. During the mid-1900s, computers were applied to create firing tables for the military. Scientists programmed these computers using computational algorithms. Computer memories were small and expensive, and successful software depended on efficient algorithms. As computer use grew, universities began to offer programming courses based on algorithm methodology. The application of mathematical science of algorithms to computers led to a new field called computer science. As demand for computer programmers grew, computer science programs at U.S. universities grew in number. U.S. universities had the computers, while universities outside the United States and Europe did not have access to computers. As the size and complexity of computer systems continued to grow, one could not rely on the theory of algorithms to provide