Chapter 22
Incorporating Free/Open-Source Data and Tools in Software Engineering Education

Liguo Yu
Indiana University South Bend, USA

David R. Surma
Indiana University South Bend, USA

Hossein Hakimzadeh
Indiana University South Bend, USA

ABSTRACT

Software development is a fast-changing area. New methods and new technologies emerge all the time. As a result, the education of software engineering is generally considered not to be keeping pace with the development of software engineering in industry. Given the limited resources in academia, it is unrealistic to purchase all the latest software tools for classroom usage. In this chapter, the authors describe how free/open-source data and free/open-source tools are used in an upper-level software engineering class at Indiana University South Bend. Depending on different learning objectives, different free/open-source tools and free/open-source data are incorporated into different team projects. The approach has been applied for two semesters, where instructor's experiences are assembled and analyzed. The study suggests (1) incorporating both free/open-source tools and free/open-source data in a software engineering course so that students can better understand both development methods and development processes and (2) updating software engineering course regularly in order to keep up with the advance of development tools and development methods in industry.

DOI: 10.4018/978-1-4666-5800-4.ch022

1. INTRODUCTION

Software engineering is considered one of the most difficult topics in computer science program. Its difficulty is not like theory courses, such as algorithm analysis, nor programming courses, such as data structures. Software engineering is an empirical course. Students should learn software engineering methods through hands-on experience, which might include real-world software development, real-world customer interaction, real-world planning and estimation, and real-world decision-making and problem-solving.

However, given the limited resources in academia, it is hard for students to learn hands-on experience in a classroom environment. Software engineering educators have been working on this issue for years and various approaches have been adopted to overcome this hurdle. For example, in some programs, industry projects are introduced into the classroom (Hayes, 2002), where students practice software engineering principles through solving challenging and complicated real world-problems. In other programs, students are asked to participate in open-source software development (Lundell et al., 2007; Stamelos, 2008; Jaccheri & Osterlie, 2007), where the source code is available for analyzing and testing. In some cases, students could be assigned to tackle a reported bug. For example, Papadopoulos et al. (2012; 2013) have used free/libre open source software (FLOSS) projects to assist teaching software engineering for at least four years. Their experiences are well documented and analyzed.

The two methods described above are proven approaches that can better integrate software engineering education with software industry practices. They all can be classified as real-world project-based software engineering education.

The software engineering course offered at Indiana University South Bend is tool and data based, where students learn software engineering methods through using software tools and analyzing software data, more specifically, free/open-source tools and free/open-source data. In this chapter, we describe how free/open-source tools and free/open-source data could be used in software engineering education to reduce the gap between industry expectations and what the academia can deliver.

The remaining of the chapter is organized as follows. In Section 2, we review related work and introduce our teaching approach. In Section 3, we describe our software engineering class, including the teaching method and the teaching experience. In Section 4, we summarize the analysis of our teaching approach. Conclusions and the improvement plan are presented in Section 5.

2. RELATED WORK AND OUR TEACHING APPROACH

Open-source software has been widely used in education (Lazic et al., 2011; Hoeppner & Boag, 2011), especially in computer science education. In software engineering field, open-source software has special usages. Because nowadays, software development largely depends on tools, which are computer software program that can facilitate the analysis, design, implementation, testing, and project management in software development. In other words, to be considered as a modern software engineer, one must know how to use various CASE (computer aided software engineering) tools.

Given the limited resources in academia, it is unrealistic to purchase all the latest commercial development tools for classroom usage. Therefore, open-source tools provide an opportunity for students to explore the latest technology development in software industry. Moreover, both the commercial software source code and commercial software development data are not accessible for most academic institutions. Without examining real-world source code and real-world development data, it is unlikely that the academia could