Chapter LIV
A Perspective on Software Engineering Education with Open Source Software

Pankaj Kamthan
Concordia University, Canada

ABSTRACT

As the development and use of open source software (OSS) becomes prominent, the issue of its outreach in an educational context arises. The practices fundamental to software engineering, including those related to management, process, and workflow deliverables, are examined in light of OSS. Based on a pragmatic framework, the prospects of integrating OSS in a traditional software engineering curriculum are outlined, and concerns in realizing them are given. In doing so, the cases of the adoption of an OSS process model, the use of OSS as a computer-aided software engineering (CASE) tool, OSS as a standalone subsystem, and open source code reuse are considered. The role of openly accessible content in general is discussed briefly.

INTRODUCTION

The steady rise of OSS (Raymond, 1999) over the last few decades has made a noticeable impact on many sectors of society in which software has a role to play. As reflected from the frequency of media articles, traffic on mailing lists, and growing research literature, OSS has garnered much support in the software community. Indeed, from the early days of GNU software to the X Window System to Linux and its utilities, and more recently the Apache Software Project, to name a few, OSS has changed the way software is developed and used.

Software engineering (Ghezzi, Jazayeri, & Mandrioli, 2003) advocates a disciplined and systematic approach to the development of high-quality software within budget, schedule, and other organizational constraints. This chapter discusses the symbiosis between traditional software engineering and open source software development (OSSD) from an educational standpoint.

The organization of the chapter is as follows. We first outline the background necessary for the discussion that follows and state our position. This is followed by a detailed treatment of key software engineering practices that are addressed in light of OSS. We then discuss the use of OSS in software

engineering education (SEE). Next, challenges and directions for future research are outlined, and finally, concluding remarks are given.

BACKGROUND

The concept of open source can mean different things in different contexts (Gacek & Arief, 2004; Perens, 1999). For the purposes of this chapter, we will use "open source" as a single encompassing term that subsumes all of the following: free/freely available or libre/liberated software whose source is available without cost to the user, imposes minimal nonrestrictive licensing conditions, and is based upon nonproprietary technologies. Software that does not fall into this category is termed non-OSS. For example, commercial software is one class of non-OSS.

As the use of OSS in various sectors of society increases, the question of how they are actually engineered garners interest. A software engineering perspective toward OSS is necessary for a variety of reasons: OSS may be adopted and used in critical areas of an organization and thus needs to be carefully examined with respect to non-OSS alternatives; OSS installed in an organization may need to be maintained over time and, therefore, needs to be well understood by maintenance engineers; and current OSS practices could be of interest from an academic (teaching, learning, research) standpoint.

Although OSS itself has a long, rich history, it is only in recent years that a software engineering viewpoint toward it has been taken (Spinellis & Szyperski, 2004; Vixie, 1999). Annual workshops in recent years under the label of Open Source Software Engineering have also created an awareness of this important area.

As OSS becomes prominent, the issue of its outreach in an educational context arises. In this chapter, we take the position that students studying software development should be exposed early to this rapidly growing area. In fact, the use of OSS in computer science education has been emphasized in recent years (Attwell, 2005; González-Barahona et al., 2000; Liu, 2003). It has also been suggested (Cusumano, 2004) that developing OSS could also help students in their future career paths.

However, the current studies of OSS-based education are limited in one or more of the following ways: the discussion is often confined to the case study of a specific OSS, does not highlight the problems associated with introducing OSS, does not address software engineering exclusively, or ignores aspects of software engineering that OSS does not address. One of the purposes of this chapter is to address these concerns.

ELEMENTS OF SOFTWARE ENGINEERING AND ITS EDUCATION AND THEIR MANIFESTATIONS IN OPEN SOURCE CONTEXTS

This section looks at six broadly classified aspects; namely, management, process, modeling/specification, standards, documentation, and quality/measurement, which are common in most SEE contexts, and examines the extent to which they are realized (or not) in an OSS environment. In doing so, we inherently set the limits of the use of OSS in SEE, which is discussed in the following section.

Management

Managing a software project is important for its eventual success. We shall limit our discussion largely to measuring success and team, time, and configuration management.

The goals of developing software in educational and OSS contexts are different. In software engineering, the software product is a means to an end, not an end in itself. It has been reported (Cusumano, 2004) that OSS often lacks precise specification of goals and, as a result, fails to define