Chapter 48
Using the Social Web for Collaborations in Software Engineering Education

Pankaj Kamthan
Concordia University, Canada

ABSTRACT

The discipline of software engineering has been gaining significance in computer science and engineering education. The technological environment in which software engineering education (SEE) resides and thrives has also been changing over the past few years. A technological revitalization of SEE requires a considerate examination from human and social perspectives. This chapter studies the impact of integrating Social Web technologies and applications based on these technologies in collaborative activities pertaining to SEE. In particular, teacher–student and student–student collaborations, both inside and outside the classroom, are highlighted. In doing so, the feasibility issues in selection and adoption of technologies/applications are emphasized and the use of pedagogically-inclined patterns is made. The potential prospects of such an integration and related concerns are illustrated by practical examples.

INTRODUCTION

In the last decade, software engineering has been playing an increasingly prominent role in computer science and engineering undergraduate and graduate curricula of Universities around the world (Rezaei, 2005; Surakka, 2007). As software engineering matures, the question of how its body of knowledge is shared, communicated, and consumed arises. In particular, the role of collaboration is likely to remain crucial to software engineering in the foreseeable future (Whitehead, 2007).

Like other disciplines, software engineering education (SEE) is prone to evolution. To do that, it needs to be sensitive to the variations and evolution of the social and technical environment around it. In particular, any changes in the information technology (IT) environment need to be reflected in SEE, if it leads to viable opportunities and demonstrated benefits (Kamthan, 2008b).

The Social Web, or as it is more commonly referred to by the pseudonym Web 2.0 (O'Reilly,

DOI: 10.4018/978-1-60566-368-5.ch048

2005), is the perceived evolution of the Web in a direction that is driven by 'collective intelligence,' realized by information technology, and characterized by user participation, openness, and network effects. In particular, the Social Web has enabled new avenues for collaborations (Coleman & Levine, 2008).

The aim of this chapter is to assess the implications of the Social Web for teacher–student and student–student collaborations in SEE, and underscore the prospects and concerns in doing so. It assumes a basic background in software engineering (Ghezzi, Jazayeri, & Mandrioli, 2003) on part of the reader. The terms Social Web and Web 2.0 are used interchangeably. For the sake of this chapter, it is also acknowledged that the notions of *coordination* and *cooperation* differ from the term collaboration and are subsumed by it.

The rest of the chapter is organized as follows. First, the background necessary for later discussion is provided and related work is presented. This is followed by a proposal for a systematic introduction of the Social Web technologies/applications for collaborations in SEE, labeled SW4CSE2 henceforth. The prospects of SW4CSE2 are illustrated using practical examples. The limitations of the underlying Social Web technologies/applications are highlighted. Next, challenges and directions for future research are outlined. Finally, concluding remarks are given.

BACKGROUND

For the sake of this chapter, collaboration is defined as collective work to achieve common goals. In this section, motivation for collaboration in software engineering is provided and related work in the area is briefly highlighted.

Motivation for Collaboration in Software Engineering Education

The human and social aspect of software engineering has been known since for quite some time (Weinberg, 1998). There is a need to foster a collaborative environment in SEE at several different levels for a variety of different goals (Whitehead, 2007), which is discussed briefly in the rest of the section.

The development of large-scale software has reached a point that it is no longer possible for an individual to grasp its size and complexity. This has necessitated (1) the use of computer-aided software engineering (CASE) tools and (2) carrying out a software project in teams, both of which require dedicated collaboration. Indeed, currently deployed commercial CASE tools such as Microsoft Project, IBM/Eclipse, and IBM/Rational Method Composer, and non-commercial CASE tools such as Subversion need certain degree of collaboration among their users.

The need for collaboration is relevant to the situative/pragmatist–sociohistoric theory of learning (Bennedsen & Eriksen, 2006). Indeed, group assignments and team projects are two common approaches to induce collaboration among students.

It is often the case that a software engineering course is equipped with a team project intended to prepare students for a similar environment later in their careers, including industrial software development. In a team project, there is a need for collaboration *throughout* the duration of the project at two levels: (1) *internal*, that is, collaboration among participating students themselves and (2) *external*, that is, collaboration between students and the (individual(s) playing the role of a) client.