

Chapter 19

Developing Soft Skills by Applying Problem– Based Learning in Software Engineering Education

Liguo Yu

Indiana University South Bend, USA

Raman Adaikkalavan

Indiana University South Bend, USA

ABSTRACT

This chapter describes how problem-based learning is utilized in a software engineering classroom to help students develop soft skills, such as communication skills, documentation skills, team working skills, and interpersonal skills. Case studies are performed at Indiana University South Bend, where experiences of two instructors with similar teaching approaches are analyzed. Studies find that while problem-based learning is promising in engaging students and effectively delivering soft skills, it is more challenging for instructors. Compared to traditional instructor-centered learning approach, problem-based learning requires instructors to spend more effort on problem design/refinement, student coaching, performance evaluation, data gathering, and experience analysis. Despite its difficulties of implementation, problem-based learning is considered an effective approach in software engineering education.

INTRODUCTION

Software engineering education courses are critical to the success of every student who aspires to be part of the computer science discipline as it allows them to apply the skills acquired in previous courses and also educates them with various other aspects of software development. One unique

characteristic of a software engineering course is its student learning outcomes, which not only include technical skills, but also soft (non-technical) skills, such as communication skills, team player skills, leadership skills, time management skills, and interpersonal skills. In addition, the course is expected to educate students on working with problems that are more than simple program-

DOI: 10.4018/978-1-4666-9634-1.ch019

ming assignments. Instead, these assignments should be real-world problems, through working on which, students can also practice customer interaction skills and project management skills (Pyster, 2009). It is hard for students to acquire these soft skills through other computer science courses, which usually focus more on programming, computing theory, and specializations such as databases, networks, security, etc. As soft skills are considered imperative for engineers in general (Becker, 2006) and 21st century software engineers in specific (González-Morales et al., 2011; Diefes-Dux, et al., 2013; Borrego et al., 2013), software engineering education has been considered a challenging task for both educators and students (Mishra et al., 2007).

Within the software engineering community, it is commonly agreed that lecture-based learning is not the most effective approach in helping students achieve the learning objectives, especially soft skills. Problem-based learning (Barrows and Tamblyn, 1980; Jones, 2006), on the other hand, has become a major teaching and learning method used in software engineering courses. However, it is not an easy task to develop a software engineering course based on problem-based learning, as instructors need to create several assignments with appropriate difficulty level and include problems that reflect real-world and at the same time should be able to map these assignments to the learning objectives of the course. Also these problems may have to be refined based on peer-assessment and student evaluations for future offerings of the same course. Considerable teaching experience and research works have been published in software engineering conferences (Frankl et al., 2014; Liem et al., 2014; Nordio et al., 2014), journals, and books (Yu, 2014). Regarding problem-based learning, yet they do not provide real assignment examples that can be replicated in similar software engineering courses taught by other faculty.

In this chapter, we discuss the undergraduate software engineering course offered at Indiana University South Bend. The learning objectives

include imparting technical skills and soft skills. We discuss how we apply problem-based learning in the course. We analyze the challenges we have been facing and present our experience.

BACKGROUND AND OBJECTIVES

Software engineering education is a challenging task for computer science educators and students (Penzstadler and Heymans, 2013). The traditional instructor-centered approach is not effective in delivering career oriented knowledge and skills, because students often consider software engineering principles as mere academic concepts and do not know how to apply them in solving real-world problems. Moreover, it is even harder for the traditional instructor-centered approach to deliver non-technical knowledge and skills, such as communication skills and team working skills.

Problem-based learning is a student-centered learning approach. The experience of solving real-world problems can help students acquire knowledge and/or skills that are hard to gain through lectures (Hmelo-Silver, 2004; Kay et al., 2000). To reduce the gap between industry expectations and what the academia can provide, problem-based learning has been introduced in software engineering classrooms (Richardson and Delaney, 2009; Armarego, 2002; Richardson et al., 2011). Table 1 summarizes the differences between instructor-centered learning and problem-based learning in software engineering education.

In problem-based learning, team projects are the most popular type for students to work on (Piccinini and Scollo, 2006). Because one of the objectives of software engineering education is to get students familiar with the complexity of real-world projects, working on a semester-long team project will allow students to tackle a large and complicated problem. A semester-long team project not only can provide students with the opportunity to experience the entire life-cycle of software development process, it also encourages

12 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/developing-soft-skills-by-applying-problem-based-learning-in-software-engineering-education/140754

Related Content

Qualitative Research Dilemma in the Study of Blog Forums: Practical Issues

Xavier Thayalanand Faridah Noor (2012). *Instructional Technology Research, Design and Development: Lessons from the Field* (pp. 152-167).

www.irma-international.org/chapter/qualitative-research-dilemma-study-blog/61268

Error Analysis of Teacher Technical Writing: The Corpus of EFL Lesson Plans

Banu Çiçek Baaran Uysal (2023). *Cases on Error Analysis in Foreign Language Technical Writing* (pp. 54-73).

www.irma-international.org/chapter/error-analysis-of-teacher-technical-writing/327017

Effectiveness of GSP-Aided Instruction

Chin-Hsiu Tai, Shian Leouand Jeng-Fung Hung (2015). *International Journal of Online Pedagogy and Course Design* (pp. 43-57).

www.irma-international.org/article/effectiveness-of-gsp-aided-instruction/126978

Effective eLearning and Transformative Pedagogical Strategies: STEM Programs

Daniel W. Keeblerand Jessica Huffman (2020). *International Journal of Online Pedagogy and Course Design* (pp. 61-70).

www.irma-international.org/article/effective-elearning-and-transformative-pedagogical-strategies/248016

The Role of Situational Context in High School Teachers Use of Graphing Calculator in Mathematics Instruction

Francis Nzuki (2016). *International Journal of Online Pedagogy and Course Design* (pp. 29-45).

www.irma-international.org/article/the-role-of-situational-context-in-high-school-teachers-use-of-graphing-calculator-in-mathematics-instruction/147744