International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:3, No:5, 2009

Promoting Collaborative Learning in Software Engineering by Adapting the PBL Strategy

Charlie Y. Shim, Mina Choi, and Jung Y. Kim

***Abstract*—**Software engineering education not only embraces technical skills of software development but also necessitates communication and interaction among learners. In this paper, it is proposed to adapt the PBL methodology that is especially designed to be integrated into software engineering classroom in order to promote collaborative learning environment. This approach helps students better understand the significance of social aspects and provides a systematic framework to enhance teamwork skills. The adaptation of PBL facilitates the transition to an innovative software development environment where cooperative learning can be actualized.

***Keywords*—**problem-based learning, software engineering, software process models, teamwork.

I. Introduction

THE Institute of Electrical and Electronics Engineers (IEEE) defines Software Engineering as "the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software" [1]. It is vital for successful industrial hires today to have the ability to understand user needs, work within a team framework, and participate in the overall process of the software engineering development [2]. Therefore, students feel software engineering is complex because it necessitates social aspects, besides technical skills, in circumstances that are sometimes less than easy. As Žagar et al. mention, software engineers are required to possess soft skills – ability to present knowledge, learn from independent sources, and listen to what others have to say [3].

Since students have different backgrounds and skills, an instructor's attempt to form balanced teams could cause even bigger issues. That is, students may end up dealing with team members who they don't want to work with. Consequently, intensive cooperation among team members sometimes cannot occur. However, the significance of supplying students with a real world teamwork environment is largely ignored in academic software development [4]. Instructors typically select a leader for each team in order to promote strong teamwork. In general, this action slightly alleviates the conflict among team members. Other researchers have proposed to use Pair Programming, a technique where two programmers work together at one computer [5]. However, Begel et al. assert that the use of Pair Programming suffers cost-ineffectiveness, scheduling problems, and personality conflicts even though it has advantages of fewer bugs and producing overall higher quality code [6]. Hence, it can be concluded that there is no panacea for lack of social skills.

The main goal of this paper is to discover a strategic method to promote strong collaboration among students in software engineering education. As a proposed solution, the PBL methodology is adapted to software engineering education. The rest of the paper is organized as follows. Chapter Two provides the underlying principles and applications of the Problem-Based Learning method. The next chapter explains the rationale and process for adapting the PBL strategy to software engineering education. The conclusion that summarizes our proposal is described in Chapter Four.

Charlie Y. Shim is with Department of Computer Science, Kutztown Univ. of Pennsylvania, Kutztown, PA 19530 USA (e-mail: shim@kutztown.edu).

Mina Choi is with Graduate School of Education, Cheongju University, Cheongju, 360764 South Korea (e-mail: mina@cju.ac.kr).

Jung Y. Kim is with Department of Computer Science, Utica College, Utica, NY 13502 USA (e-mail: jkim@utica.edu).

II. Problem-based learning

In this chapter, the characteristics of the PBL strategy and popular models to which PBL was applied are described.

A. Concept and Characteristics

Problem-Based Learning (PBL) is the learner-oriented instructional strategy. Barrows et al. defines PBL as the following: "Problem-based learning is the learning that results from the process of working toward the understanding or resolution of a problem. The problem is encountered first in the learning process" [7]. Learning in PBL is the consequence from the practice and teachers play the role of facilitators of learning [8], [9]. PBL has been widely employed since learners need to solve real-world problems by themselves through cooperative working in groups. The main characteristics of PBL are (1) use of real-world problems, (2) encouragement of students' active participation, (3) integration of diverse view points, (4) encouragement of self-directed learning, (5) encouragement of team collaboration, and (6) enhancement of education quality [10]. In the next section, two popular application models to which the PBL strategy was applied are introduced.

B. PBL Models

1) Barrows and Myers Model

This is the first implementation of the PBL strategy in medical school, called McMaster System. Students in medical schools are required to memorize a huge amount of information even though the most important physician's capability is not memory skills [7]. Thus, they applied the PBL approach to

medical education so that students could enhance creative problem solving skills in biomedical field. This model includes five major steps: (1) introducing the PBL concept and forming classroom environment, (2) introducing problem configuration and assigning duties, (3) analyzing collected information critically, (4) deriving a solution for the problem, and (5) abstracting obtained knowledge and self-evaluating [11].

2) Fogarty Model

Fogarty asserts that PBL engages learners in intriguing, real and relevant intellectual inquiry allowing them to learn from these life situations [12]. He suggests seven major steps for the PBL approach: (1) facing a problem: assigning a poorly structured problem to students, (2) defining the problem: restating the problem in their own words, (3) making assumptions: establishing background theories and necessary assumptions, (4) searching: searching and collecting information, (5) modifying: updating the initial problem statements based on the collected information, (6) finding alternative solutions: creating ideas for alternative solutions through communications, and (7) evaluating: evaluating a proposed solution to the problem [12].

The above applications represent common procedures employed in various PBL models and they are as follows: presenting a problem, solving the problem through individual or team activities, and evaluating the output as well as the overall course of actions. However, a series of actions can be adjusted in order to fit a special purpose learning group or environment. In the next section, the PBL approach applied to software engineering education is examined.

III. Adapting the PBL Strategy to Software Engineering

This chapter depicts why and how the PBL strategy is adapted to software engineering education for the purpose of promoting collaborative learning environment.

A. Correlation between PBL and Software Engineering

The underlying principles of PBL and the main characteristics of software engineering projects are naturally correlated. It is because group projects have been widely adopted in many (under)graduate courses such as in Software Engineering, Computer Science, and Information Technology in order to simulate a real-world setting [4]. The relationships between the PBL strategy and projects are summarized as follows.

• Use of real-world problems: Due to the limitations in classroom environment, most software engineering projects assigned to students are smaller than the actual size of projects used in companies. The nature of projects, however, still should be as complete as real-world instances so that students can experience the whole process that is likely to be seen in future working environment.

•Encouragement of students' active participation: Students make final decisions and handle all situations in team projects. Each student is encouraged to strongly participate as a team member, which is essential to the successful completion of a project.

•Integration of diverse view points: Each student as a team member may have a different viewpoint to the given problem. Diverse ideas should be organized and integrated through a proper communication channel in order to effectively solve the problem.

•Encouragement of self-oriented learning: Each student, as an entity to make important decisions, is supposed to find their own way to handle tough situations using their knowledge. Problem solving must be in a self-oriented learning environment. An instructor, as a mentor, is supposed to help students manage the problem without deeply being involved in the team activities.

•Encouragement of team collaboration: software engineering is a classic computer science course in which students are supposed to work on team projects. Students have chances to apply their integrated knowledge to real-world problems. This is a simulation of whole software engineering process where collaboration among team members is expected.

•Enhancement of education quality: software engineering is a cap-stone course designed to simulate the process of software development. In this course, students combine the knowledge that they have learned and apply to a real-world problem. The use of integrated knowledge to solve a particular problem can guarantee students' learning, which enhances the quality of education.

B. Software Process Models

It is proposed to adapt the PBL strategy to a software process model. A software process model is an abstract description that encompasses the fundamental sequence of activities throughout the entire process in developing a software product [13]. Other alternative classifications are possible but general process frameworks include one of the following sequence of activities: communication, planning, modeling, construction, and deployment [13]; specification, development, validation, and evolution [14]; requirements analysis, design and specification, code and module testing, integration and system testing, delivery and maintenance [15]. Common features of the above process models are recognition of requirement analysis and post-activities after the completion of a software product, which naturally necessitates the collaboration among learners. Therefore, based upon the correlation, it is desirable to adapt the PBL strategy to the management of team projects in software engineering education in order to promote social aspects when developing a software product.

C. Adapting PBL to Software Engineering

The main idea is to adapt the PBL principles to Pressman's model, one of the most prominent process standards in software engineering. In this study, Pressman's Modeling and Construction phases are proposed to be combined into a single stage for the purpose of simplicity, which makes the overall process into a total of four phases: Communication, Planning, Modeling & Construction, and Deployment. Fig. 1 illustrates the PBL method adapted to software engineering education.

The main activities in the communication phase are

described as follows. First, an instructor introduces the concept of PBL: introducing learning methods and strategies by explaining the importance of social aspects; explaining how to use PBL to develop teamwork skills in project management; Second, an instructor introduces the roles of the instructor/learners: explaining that students are supposed to perform all the activities by self-regulating in the project management and that instructors just facilitate and coach the course management; having students be ready for critical and flexible thinking; asking students to build up speculation and communication skills so that they can effectively and harmoniously solve potential conflictions caused by the different viewpoints about the problem; Third, learners identify the problem: analyzing the situation of the problem, and recognizing the importance of the problem and key points.

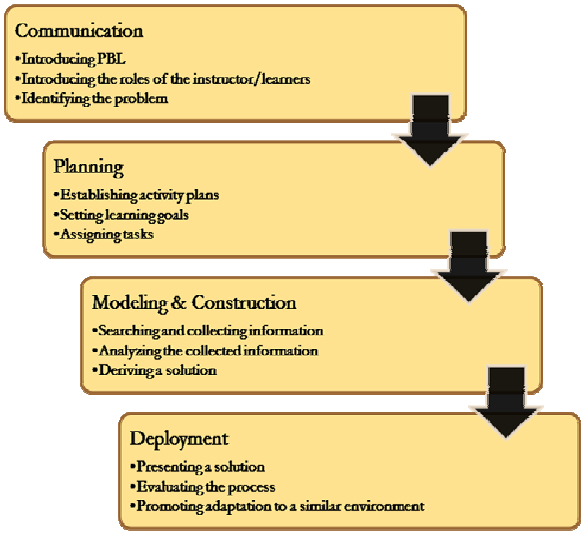

Fig. 1 The PBL method adapted to Software Engineering

The activities in the second phase are described as follows. First, learners establish activity plans: internalizing the problem by analysis; activity planning (e.g. asking team members to plan both teamwork skill development and project management); logging the status of the problem (e.g. creating documents such as requirement documents, work breakdown structure, Pert chart, Gant chart, etc.); bring up the ideas to solve the problem; Second, learners set learning goals and assign tasks (e.g. listing tasks and setting up learning goals); Third, learners assign tasks to individuals or teams.

The activities in the modeling and construction phase are described as follows. First, learners search and collect information: searching and collecting useful information and data to solve the problem using various resources (e.g. literatures, the internet, etc.); Second, learners analyze the collected information: sharing collected information; modifying initial statements based on the shared information and focusing on the core of the problem; Third, learners derive a solution: finding the best solution based on the collected and analyzed information; categorizing possible solutions, alternatives, and resolutions.

The activities in the last phase are described as follows. First, learners present a solution: presenting their solution to the team members; Second, both the instructor and learners evaluate the process: evaluating outputs; evaluating him/herself, peer, instructor, based on the agreed evaluation grids; Third, learners promote adaptation to a similar environment in the future: reflecting learning outputs to learning objectives; applying the output to similar situations.

The above phases represent the overall sequence of activities which reflect the PBL strategy promoting collaborative learning environment in developing a software product.

IV. Conclusion

Students feel software engineering is complicated because it necessitates teamwork skills, aside from technical expertise; thus, it is desirable to integrate human-based aspects into software engineering classroom so that students can be equipped with communication, interaction, and cooperation among team members, which is essential for successful software development.

The PBL methodology is proposed to be integrated into software engineering education. The rationale for this adaption is that the underlying principles of PBL fits collaborative learning environment; They are use of real-world problems, encouragement of students' active participation, integration of diverse view points, encouragement of self-oriented learning, encouragement of team collaboration, and enhancement of education quality. The innovative adaption aims to help students understand the significance of social aspects in software development and improve teamwork skills. This approach is expected to provide a systematic framework to assist learners in preparing for rigorous industrial settings in the future.

References

[1] IEEE *Standards Collection: Software Engineering*, the Institute of Electrical and Electronics Engineers (IEEE) Standard 610, IEEE, 1993.
[2] Petkovic, D., Thompson, G., and Todtenhoefer, R., "*Teaching practical software engineering and global software engineering: evaluation and comparison,*" Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education, pp. 294-298, 2006.
[3] Žagar, M., Bosnić, I., and Orlić, M., "*Enhancing software engineering education: a creative approach,*" Proceedings of the 2008 international workshop on Software Engineering in east and south Europe, pp. 51-58, 2008.
[4] Su, H., Jodis, S., and Zhang, H., "*Providing an integrated software development environment for undergraduate software engineering courses*," Journal of Computing Sciences in Colleges, Volume 23, Issue 2, pp. 143–149, December, 2007
[5] Williams, L. & Kessler, R., "*Experimenting with Industry's "Pair-Programming" Model in the Computer Science Classroom*," Journal of Computer Science Education, March 2001.
[6] Begel, A., and Nagappan, N., "*Pair Programming: What's in it for Me?*" Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement, pp. 120-128, 2008.
[7] Barrows, H.S. & Myers, A.C., "*Problem-based learning in secondary schools*," Springfield, IL: Problem-based learning institute. Lanphier High school and Southern Illinois University Medical School, 1993.

[8] Barrows, H. S., "*Problem-based learning in medicine and beyond: a brief overview,*" New Directions for Teaching and Learning, n68, pp3-12, Win 1996.
[9] Barrows, H. S. and Tamblyn, R. M., *Problem-based learning: An approach to medical education*, Springer Series on Medical Education, Volume 1, Springer Publishing Company, 1980.
[10] Delisle, R., *How to use problem-based learning in the classroom*, Alexandra, VA: Association for Supervision and Curriculum, 1997.
[11] Savery, J. and Duffy, T., "*Problem-based Learning: An instructional model and its constructivist framework*," Educational Technology, 35(5), 35, 1995.
[12] Fogarty, Robin, "*Problem-Based Learning and Other Curriculum Models for the Multiple Intelligences Classroom*," IRI/SkyLight Training and Publishing, 1997.
[13] Pressman, R., *Software Engineering: A Practitioner's Approach*, McGraw-Hill, 7th edition, January, 2009.
[14] Sommerville, I., *Software Engineering*, Addison Wesley, 7th edition, May, 2004.
[15] Ghezzi, C., *Fundamentals of Software Engineering*, Prentice Hall, 2nd edition, September, 2002.