# Software Engineering Project Management with the Community of Practice Approach: Toward Changes in Software Engineering Education

Charlie Y. Shim
Dept. of Comp. Sci. & Info. Tech.
Kutztown University of Pennsylvania
Kutztown, PA, USA
e-mail: shim@kutztown.edu

Yeungkwun Kim
School of Computer Science
Western Illinois University
Macomb, IL, USA
e-mail: Y-Kim2@wiu.edu

Injoo Kim
Dept. of Comp. & Info. Sci.
East-West University
Chicago, IL, USA
e-mail: injoo@eastwest.edu

*Abstract*—**In this paper, we propose to integrate the Community of Practice (CoP) approach into software engineering classroom environment and develop CoP sessions for software engineering courses. Incorporation of the CoP method rejuvenates the importance of human and social aspects and trains students to be equipped with teamwork skills essential for establishing a mutually supportive and collective learning working environment, such as software development.**

***Keywords – Software Engineering; Community of Practice; Human and Social Aspects; Collective Learning; Classroom Environment.***

## I. INTRODUCTION

Software engineering has frequently been used as a capstone course for many computer science programs in academia. The importance of people-related issues and teamwork skills in software engineering have been recognized and resulted in many researches [1]-[4]. Recent trends in information technology project management, such as globalization, outsourcing, and virtual teams [5] have even been boosting the role of human and social factors. Based on these, it is obvious that students should start to practice teamwork skills that could handle people-related issues and launch working environments for mutually supportive and collaborative learning from their school years.

Students, however, often criticize the classroom environment where instructors discuss people-related issues in an extremely limited way so that students cannot receive enough assistance or instruction on how to work effectively in a team [6]. We also found similar results from informal interviews with students. For the question, “What was the most challenging issue that you faced while working on a capstone project in software engineering course?” students frequently answered they had a hard time in “managing efficient communication among team members”, “addressing conflicts among team members”, and “adjusting unfair contribution among team members”, which all need teamwork skills based on human and social aspects.

It seems that the students’ real problem does not exist in applying software engineering knowledge to a capstone project, but does exist in handling people-related issues properly. Instructors may put this down to many challenges in a classroom environment, such as limits in time and opportunities, differences in students’ background, etc. As we saw in current trends, human and social factors are no longer an option that can be postponed. Students should experience and learn how they effectively work as a team in the classroom environment and be ready to establish mutually supportive and collective learning working environment in the real world. In the swim of this, we should consider integrating a systematic process of teaching and evaluating teamwork skills based on human and social factors as a part of a software engineering course.

The CoP is a group formed by people who regularly engage in sharing and learning, based on their common interests [7]. Many organizations integrated the CoP method as a part of their organization structure with the purpose of improving their performance [8]. Adopting this method into the classroom environment will provide students with chances to build teamwork skills to manage people-related issues needed for software engineering project management. In this research, we propose to adopt the CoP method and develop CoP sessions that are specific to a software engineering course, where we believe is the right place to incorporate this approach.

The rest of this paper is arranged as follows. Section II describes the related concepts of software engineering, human aspects in software engineering, CoP, and key elements of CoP. Section III describes the commonalities between CoP and software engineering project management. CoP sessions designed for a software engineering course and issues to consider are also discussed. The conclusion and future study end the paper.

## II. RELATED WORK

### A. *Software Engineering and Human Aspects*

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” [9]. It is common for many computer science programs to assign two consecutive semesters for a software engineering course. Students learn software engineering skills and knowledge during the first semester, then apply the skills and knowledge to complete a capstone project of software development during the second semester. Since software engineering is where students can simulate real-world working environments through a project, it has frequently been developed as a project-based capstone course in many computer science programs. Human and social factors play as much an important role as academic knowledge in a project-based course like software engineering. Students who successfully complete the course are expected to have the ability to understand user needs, work within a team framework, and participate in the overall process of the project management.

### *B. Community of Practice*

The CoP is an educational literature that refers to a network of peers who share a common interest in a particular topic and come together to fulfill both individual and group goals [10]-[11]. Through such networks, members wish to collaborate with other members by sharing information and developing knowledge [10]. The community of practice consists of three key elements [10]:

- **Mutual Engagement** – Individual understanding should be supported by a practical and meaningful connection among members obtained by real contribution of members to complement each other.
- **A Joint Enterprise** – The community should have a context of being part of a broader system and operated through a process of collective negotiation.
- **A Shared Repertoire** – The community should standardize terms used in discussions.

We believe the CoP approach in the educational technology field can be considered as a possible solution to integrate a systematic process of teaching and evaluating human and social factors as a part of a software engineering course.

## III. INTEGRATING THE COMMUNITY OF PRACTICE APPROACH INTO SOFTWARE ENGINEERING COURSE

### *A. Understanding Software Engineering Project Management in the Context of Community of Practice*

Three key elements of the CoP mentioned in Section II.B are also essential for the area of software engineering, especially for software engineering project management:

- **Mutual Engagement** – A successful project management mandates individual member's clear understanding of the project. This can be accommodated by active communication among team members. The environment of mutually supportive and collaborative learning through active communication can promote a truly cohesive relationship among members and their strong contribution to the project.
- **A Joint Enterprise** – A systems approach is popularly adopted for a successful project management. In this approach, top management and project managers should understand how projects are related to the whole organization. A project should be considered within the larger organizational context and operated in a wide organizational environment.
- **A Shared Repertoire** – Team members' clear understanding of a project is essential for a successful project management. To help their understanding, use of standardized terminology and maintaining efficient communication channels among members are necessary. Records of communication history and definitions of standardized terminologies can be included as a part of the project management document.

### *B. Designing Community of Practice Sessions for Software Engineering Class*

As we discussed in Section III.A, the CoP approach and software engineering project management share similarities in their main goals. Given the similarities, we propose to adopt the CoP approach for the software engineering course. Table 1 shows the outline of the CoP sessions that we designed for a software engineering course.

We assumed a two-semester structure mentioned in II.A. Thus, the activities in Table 1 can be implemented during the second semester while students work on a capstone project using the knowledge obtained from the first semester. For one semester software engineering course, some of the sessions can be combined and the weeks to do the activities can be adjusted to fit into the class schedule. Table 1 lists only CoP activities that will be integrated into an existing software engineering class schedule.

### *C. Issues to Consider*

The core of the commonalities between CoP and software engineering project management exists in the issues of people and their cohesive interactions. The following issues should be considered when integrating CoP for project management into the software engineering classroom:

- **Size and Members** – Each project team formed in a software engineering classroom will work as a CoP. Based on our teaching experience, we propose to form a team of 3 to 5 members. The appropriate size for a team can be properly adjusted according to a classroom environment. The bigger the size, the harder to maintain close cohesion among members. Team members with various backgrounds in their coursework and other experiences may be helpful for accommodating a collaborative learning environment among members.
- **Communication** – Efficient communications among team members are crucial for the success of

TABLE 1. OUTLINE OF COP SESSIONS FOR A SOFTWARE ENGINEERING COURSE

| # | Week | Activity Description |
|---|---|---|
| S1 | 1 | ***CoP Concept***<br>- The instructor introduces the idea and importance of the CoP.<br>- The instructor explains how and where the CoP will be integrated into software engineering project management<br>- The instructor checks students' understanding of the CoP |
| S2 | 1 | ***Review of Teamwork Principles***<br>- The instructor reviews teamwork principles and their importance.<br>- The instructor introduces the common problems that teams from previous semesters had. |
| S3 | 2 | ***Project Concept***<br>- Team members have a general discussion about the nature of the project.<br>- Each member should represent their understanding of the project and get feedback from other members.<br>- Through the discussion, members help each other to understand the project and make sure they are on the same page. |
| S4 | 2 | ***Scope of Assigned Work***<br>- Each team member clearly understands their role in the project and clear scope of the assigned job. |
| S5 | 3 | ***Communication and Evaluation Method***<br>- Team members adopt a standard method of communication among the members.<br>- Team members adopt an evaluation method for each member's contribution. |
| S6 | 8 | ***Intermediate Evaluation for the CoP activities so far***<br>- Team members discuss whether communication among members has been maintained effectively so far.<br>- Team members discuss whether mutually supportive and collective learning environments have been maintained so far.<br>- Team members discuss other project management related issues.<br>- Team members plan on necessary changes for the identified issues.<br>- The instructor performs intermediate surveys on students' teamwork experience. |
| S7 | 15 | ***Final Evaluation***<br>- Team members have a final discussion on their teamwork experience with the project management.<br>- Team members evaluate other members contribution to the project.<br>- The instructor performs final surveys on students' teamwork experience. |

the CoP and the project. Meetings are the most common way of managing communication among members. Keeping meeting minutes is recommended to address possible confusion and misunderstanding. The use of social network services, such as a blog, can also be considered. Creating and maintaining a CoP blog for the project management could be helpful for a team with meeting schedule issues. Instead of face-to-face meetings, members could maintain communication through the blog.

- **Effectiveness** – Measuring the effectiveness of the applied CoP sessions will make the approach more robust and meaningful. In Table 1, two forms of surveys (i.e., intermediate, and final) were proposed for this. Comparisons between the two survey results are expected to demonstrate how students' experience has been improved. Details on the contents of the proposed surveys will not be discussed in this paper. That will be a part of the continued research.

## IV. CONCLUSION

Despite their importance, teamwork skills needed for a project management are currently not practiced enough in the software engineering classroom. To address the issue, we proposed the adoption of the CoP method into the software engineering classroom. We justified the applicability of the CoP method to software engineering classroom environment and developed specific CoP sessions for a software engineering course. With the help of the CoP, students will understand the importance of human and social factors in a project management. Students are also expected to practice people-related issues in a systematic way and be equipped with necessary teamwork skills to accommodate mutually supportive and collective learning environment for software development.

## V. FUTURE STUDY

In our future work, we will develop two surveys for the proposed CoP sessions and apply the sessions to the software engineering course. Students' experience will be assessed through the surveys. We will also discuss a possibility of designing and adopting a formal measure to assess the effectiveness of the CoP sessions integrated into a classroom environment.

## REFERENCES

[1] K. Garg and V. Varma, "People Issues Relating to Software Engineering Education and Training in India," Proceedings of the ACM-SIGSOFT's 1st India software engineering conference (ISEC '08), pp. 121-128, February 2008.

[2] J. Karn and T. Cowling, "A Follow up Study of the Effect of Personality on the Performance of Software Engineering Teams," Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering (ISESE'06), pp. 232-241, September 2006, Rio de Janeiro, Brazil.

[3] M. John, F. Maurer, and B. Tessem, “Human and Social Factors of Software Engineering: Workshop Summary,” ACM SIGSOFT Software Engineering Notes, vol. 30, issue 4, pp. 1-6, July 2005.

[4] J. M. Hogan and R. Thomas, “Developing the Software Engineering Team,” Proceedings of the 7th Australasian Conference on Computing Education, vol. 42, pp. 203 - 210, 2005.

[5] K. Schwalbe, Information Technology Project Management (6th Edition), Course Technology, CENGAGE Learning, 2010, ISBN: 978-0-324-78692-7.

[6] H. Su, S. Jodis, and H. Zhang, “Providing an integrated software development environment for undergraduate software engineering courses,” Journal of Computing Sciences in Colleges, vol. 23, issue 2, pp. 143-149, December, 2007.

[7] E. L. Lesser and J. Storck, “Communities of practice and organizational performance,” IBM Systems Journal, vol. 40, no. 4, pp. 831-841, 2001.

[8] R. McDermott and D. Archibald, “Harnessing Your Staff’s Informal Networks,” Harvard Business Review Magazine, March 2010. [Online] Available from: http://hbr.org/2010/03/harnessing-your-staffs-informal-networks/ar/1/ 2023.03.09

[9] IEEE Standards Collection: Software Engineering, the Institute of Electrical and Electronics Engineers (IEEE) Standard 610, IEEE, 1993.

[10] E. Wenger, Communities of Practice: Learning, meaning and identity, Cambridge University Press, 1998.

[11] E. Wenger, R. McDermott, and W. Snyder, Cultivating Communities of Practice: A Guide to Managing Knowledge, Harvard Business School Press, 2002.