Proceedings of the 3rd International Conference on Computing and Informatics, ICOCI 2011,8-9 June, 2011 Bandung, Indonesia

Paper No. 011

PAIR PROGRAMMING IN INDUCING KNOWLEDGE SHARING

Mawarny Md. Rejab, Mazni Omar, Mazida Ahmad, Khairul Bariah Ahmad

Universiti Utara Malaysia, Malaysia, { mawarny@uum.edu.my, mazni@uum.edu.my , mazida@uum.edu.my, kbariah@uum.edu.my }

ABSTRACT. The Pair programming as a part of the Agile software development has been gaining acceptance among practitioners and software development community. This successful leads a wide use of pair programming in educational setting as pedagogy in programming course. Pair programming can foster knowledge sharing among students. Many studies have been done with pair programming in education however most of them do not highlight internalized knowledge particularly tacit knowledge from knowledge sharing processes between students who act as driver and navigator in pair programming practice. Thus, this paper will discuss knowledge internalization based on the knowledge sharing activities in pair programming practices by employing the process of Socialization, Externalization, Combination and Internalization (SECI). The sample of the study consisted of 60 students who were actively engaged in pair programming practices. The factors investigated were types of internalized tacit knowledge in the form of learning, thinking and decision making skills among the students. Online questionnaires were adapted from SECI model into educational context. T Test technique was used to analyze the data. This study is expected to contribute a better understanding of important knowledge sharing activities to construct student's skills during Internalization process through pair programming. This study's result will be considered for future rigorous theoretical framework for constructing tacit knowledge among the students in pair programming environment.

Keywords: pair programming, knowledge sharing, SECI

INTRODUCTION

Pair programming as one of the key practices in Extreme Programming has been gaining acceptance among practitioners and software development community. This successful leads a wide use of pair programming in educational setting as a computer science or software engineering pedagogical tool especially in programming courses (Canfora et al., 2003; Brereton et al., 2009, Cliburn, 2003; Mendes et al., 1997). Various studies have been done on determining the usefulness and effectiveness of Pair Programming as pedagogical tool and indicated the following positive results:

a) Pair programming can improve students' performance by gaining higher scores on programming assignment (Werner et al., 2004; McDowell et al., 2003; Cliburn, 2003; Slaten et al., 2005).

b) Pair programming can increase student's confidence and satisfaction (Werner et al., 2004; McDowell et al., 2003; Cliburn, 2003; Slaten et al., 2005).

c) Pair programming can encourage students to complete the programming course (Werner et al., 2004; McDowell et al., 2003)

Pair programming shifts the programming learning from solitary activity into a collaborative learning process (McDowell et al., 2003). It involves two students who act as a driver and navigator working on the same problem from design to testing phase. In general, driver is a person who involves in creating and implementing the code, whereas navigator who is responsible in checking errors and suggesting the implementation technique. Navigator provides alternative solution to the given problem and assists the drivers to solve the problem. Meanwhile, driver fully controls all input through the keyboard or mouse and come out with solution based on his/her idea or navigator's suggestions (Williams and Kesler, 2000; Beck, 2005).

Besides roles, switching partners is an important issue that should be considered in implementing the pair programming. Switching partners and roles rotation can induce knowledge sharing among students (Chau and Maurer, 2004; Beck, 2005). This leads to exchange or spread information and knowledge throughout the whole team of software development (Muller and Tichy, 2001). Indeed, a better structured pair interaction is required by having a proper communication within a pair (Gallis et al., 2003; Beck, 2005). Pair programming involves an informal and spontaneous communication as relies on face-to-face communication between driver and navigator (Chau and Maurer, 2004). However, frequent switching of partners is required in achieving knowledge sharing (Gallis et al., 2003).

Pair programming can foster knowledge sharing among students. pair programming is usually performed by students, as novice programmer to develop small programming tasks, which improves knowledge transfer and quality (Vanhanen and Korpi, 2007). Many studies have been done with pair programming in education however most of them do not highlight internalized knowledge particularly tacit knowledge from knowledge sharing processes between students who act as driver and navigator in pair programming practice. Thus, this paper will discuss knowledge internalization based on the knowledge sharing activities in pair programming practices by employing the process of Socialization, Externalization, Combination and Internalization (SECI). For completeness, the overview of knowledge sharing will be discussed more details in section 2. The third section describes the method used in the implementation of this study. The last section describes the result and discussion of this study.

KNOWLEDGE SHARING

Knowledge management is needed to properly manage the knowledge shared within an organization. Knowledge management is a systematic process involving exploration, choosing, rearranging, repairing and information delivering that can uplift individual ability in attended field (Sommerville and Craig, 2006). Apart from managing the knowledge, knowledge management also covers knowledge supervising processes consisting of knowledge creation process, knowledge storing process and knowledge application process and become the basis to new knowledge creation (Natarajan and Shekar, 2001). Generally, knowledge management covers obtaining process, sharing, utilizing and storing knowledge among individuals in an organization. Knowledge sharing is the important part of knowledge management and crucial task in agile software development processes. It is promoting the knowledge transmission among individuals in community or organization and normally supported by the knowledge sharing mode (Fengjie et al., 2004). There are various kind knowledge management modes that enable the individuals to exchange knowledge such as face-to-face communication, conference, knowledge network, and organization learning. However, this study focuses on the face-to-face communication as a knowledge sharing mode in co-located pair programming practices.

According to Fengjie et al.(2004), knowledge sharing process involves two main parties namely contributor and receiver. Contributor contributes a part of his/her knowledge and transmits to the receiver. The receiver will receive the knowledge and try to add his/her understanding and transform it into his/her knowledge. This scenario similar to the pair programming practices where the navigator plays as a contributor and the driver as receiver.

Navigator will provide suggestion in assisting the driver to solve the given problem in implementing the program whereas the driver will use the suggestion given and blend with his/her own knowledge to come out with the best solution.

There are three steps involves in transmitting the fluent knowledge. Firstly, the receiver not just the knowledge beneficiary but also the knowledge provider during the knowledge sharing process. In pair programming, a tendency to avail the knowledge sharing should be overcome by promoting them a reward (Yin and Zhang, 2005). In learning environment, a grade satisfaction is a reward to encourage them sharing their knowledge in achieving a good solution in programming. Besides, a well-designed knowledge sharing space is required to assist knowledge transition such as NetMeeting, Yahoo/MSN Messenger, web-based knowledge sharing system and others. There is no restriction of time and place especially to implement the distributed pair programming. Students need the virtual collaboration environment when their schedules are conflicts and they cannot get physically together in finishing the programming assignment (Ho et al., 2003). Lastly, a proper way is needed to ensure the knowledge is easy to understand by having an effective communications.

During the pair programming process, some explicit and mostly tacit knowledge is shared between the driver and navigator (Chau and Maurer, 2004). Explicit knowledge is easy to share because it can be expressed in words and numbers (Nonaka and Konno, 1988; Ho et al., 2003; Fengjie et al., 2004). However, the representation of explicit knowledge is easy to understand and convenient to retrieve should be considered during explicit knowledge sharing process (Fengjie et al., 2004). Meanwhile, more efforts are required to gain tacit knowledge due to very hard to formalize and difficult to codify the tacit knowledge. Tacit knowledge is a human judgment and strategic decision making (Brockmann and Simmonds, 1997; Guthrie, 1995). The main sources of tacit knowledge are experience and thinking (Gerard, 2003). Tacit knowledge is related to teaching and learning process and also generated through learning experience (Gerholm, 1990). Thus, tacit knowledge will be obtained through pair programming practices between pairs to generate learning, thinking and decision making skill.

Oppose to explicit knowledge, tacit knowledge is hard to share due to it's difficulty to express it in language (Fengjie et al., 2004). Thus, Socialization, Externalization, Combination and Internalization (SECI) is adopted in this study to facilitate knowledge conversion between tacit and explicit knowledge and also to promote knowledge sharing between partners during pair programming practice. Socialization is a process of sharing experiences and thereby creating tacit knowledge such as shared mental models and technical skills. Externalization means the process of articulating tacit knowledge into written form or explicit knowledge but still inconsistent condition. So that it can be shared by others and become the basis of new knowledge. Combination refers to the process of converting explicit knowledge that is inconsistent into a more complex and systematic sets of explicit knowledge. During internalization process, the experiences will be converted into individual knowledge (Nonaka and Takeuchi, 1995). The spiral indicates the spread of knowledge among colleagues and emphasizes the importance of interaction between tacit and explicit knowledge dynamically and consistently. In pair programming practices, knowledge sharing involved social interaction, sharing and constructing knowledge between the partners. Thus SECI model is applicable to promote sharing and constructing knowledge between partners in generating learning, thinking and decision making skill.

This paper discusses on internalization based on the knowledge sharing activities in pair programming practices by employing the process of SECI. The factors investigated were types of internalized tacit knowledge in the form of learning, thinking and decision making skills among the students. In order to assess empirically the effect of knowledge sharing amongst programmers using pair programming, the following hypotheses has been formulated.

- Ho: There is no difference state of learning activities in internationalization knowledge sharing between the pair programmers and non-pair programmers
- H_1: There is significance difference state of learning activities in internationalization knowledge sharing between the pair programmers and non-pair programmers
- Ho: There is no difference state of thinking activities in internationalization knowledge sharing between the pair programmers and non-pair programmers
- H_1: There is significance difference state of thinking activities in internationalization knowledge sharing between the pair programmers and non-pair programmers
- Ho: There is no difference state of decision making activities in internationalization knowledge sharing between the pair programmers and non-pair programmers
- H_1: There is significance difference state of decision making activities in internationalization knowledge sharing between the pair programmers and non-pair programmers

METHOD

Procedure and Sample

The sample of the study consisted of undergraduate College of Arts and Sciences (CAS) students at Universiti Utara Malaysia (UUM) enrolled in *Basic Programming* course. *Basic Programming* course is a compulsory course for first year student in information technology (IT), multimedia, and education in IT. Each week students attend two hours of lectures and two-hour laboratory session. In the laboratory, students were required to solve programming assignments assigned by the lecturer. Students were divided into two groups; pair programming group and non-pair programming group to work on the assignments. During the lab session, an instructor was assigned to assist and support the students to solve programming problems for both groups.

At the mid of the semester, 119 questionnaires were distributed to the students who were actively engaged in pair programming practices and had experience applying non-pair activities. Students were required to complete ten-minute survey to determine level of knowledge sharing amongst pair and non pair programmers. All questionnaires were returned completed, representing an acceptable response rate. Of the 119 questionnaires administered, 77 students from the pair programming groups and 42 from the non-pair programming groups completed the survey. To ensure the validity of knowledge sharing scores, outlier data was excluded in the analysis, resulting in data set of 118 respondents. The age of respondents ranged from 20 to 25, with a mean age of 18.7 years. Slightly more than 65.5% of the respondents were female.

Measure

In order to test the hypotheses, a survey study was conducted. The questionnaire was adapted from SECI model in educational context (Mazida, 2010) particularly focuses on internalization factor. The validity and reliability of this questionnaire was demonstrated in other study (Mazida, 2010).The factors investigated were types of internalized tacit knowledge in the form of learning, thinking and decision making skills among the students. All items in the questionnaire were measured using a five point Likert scale ranged from “1-Strongly disagree”, “2-Disagree”, “3-Don’t Know”, “4-Agree“, and “5-Strongly agree”.

Independent t-test was conducted to measure level of knowledge sharing between pair programming and non-pair programming groups. Independent t-test was used to compare the two groups’ level of knowledge sharing in terms of learning, thinking and decision making skills between those groups. SPSS tool was used to analyze the data. Reliability analysis for this questionnaire was 0.7, which exceeds minimum requirement of Cronbach Alpha, 0.6 (Nunally, 1978).

RESULT AND DISCUSSION

Data were analyzed in terms of learning, thinking and decision making skills. The main goal was to demonstrate that pair programming practice is a viable tool to promote knowledge sharing activities, particularly amongst the programming students.

Learning

There was no significance difference in score of learning for pair programming (M =20.24, SD=3.40), and non-pair programming groups [M =19.43, SD=3.76; t(116)=1.19 , p=0.24]. This can be illustrated in Table 1.

Table 1. Group Differences for Learning between Pair Programming and non-pair programming groups

	Pair Programming		Non-Pair Programming		
Learning	M	SD	M	SD	T
	20.24	3.40	19.43	3.76	1.19

*p < 0.05

In education context, pair programming involved two novices that need guidance from the lecturer as an expert. Even though they can do the job but still facilitation from the lecturer exceeds what can be attained in pair (Vygotsky, 1978). Level of students' potential ability is uplifted to a higher level with the guidance from the expert compared to self learning (Holzman, 2009). The finding is supported by a claim that students require strong support from the instructor for a better education (Heywood, 1992). With the continuous guidance from the lecturer, lecturer's tacit knowledge is transferred and the knowledge is shared amongst the pair.

Thinking

There was significance difference in score of thinking for pair programming (M =20.24, SD=3.40), and non-pair programming groups [M =19.43, SD=3.76; t(116)=2.47 , p=0.015]. This can be illustrated in Table 2.

Table 2. Group Differences for Thinking between Pair Programming and non-pair programming groups

	Pair Programming		Non-Pair Programming		
Thinking	M	SD	M	SD	T
	18.97	2.26	17.90	2.25	2.47*

*p < 0.05

Students applying pair programming showed higher order thinking skills compared to solo programmers (Williams, 2002). This is because they can share knowledge to solve programming assignment with their partner during pair programming. Therefore, the students are more independent without fully relying on instructor to discuss solutions during lab session. In addition, by doing programming in pair, tacit knowledge instilled in brain can be transferred among the students, which encourages intrinsic motivation among team members (Mazni et al., 2009). In pair programming, students need to be alert and attentive to check and review their partner's code program. This situation encourages them to think more compared to non-pair programmer, which develops codes in isolation. When this happens, logical thinking amongst the pair increased and assisted them to broaden way of thinking, which improve creativity during programming activities.

Decision Making

There was no significance difference in score of decision making for pair programming (M =17.82, SD=2.28), and non-pair programming groups [M =17.76, SD=1.97; t(116)=0.13 , p=0.9]. This can be illustrated in Table 3.

Table 3. Group Differences for Decision Making Between Pair Programming and non-pair programming groups

	Pair Programming		Non-Pair Programming		
Decision	M	SD	M	SD	T
Making	17.82	2.28	17.76	1.97	0.13

*$p < 0.05$

Pair programming promotes better decision making when two heads better than one (Chong et al., 2005). However, in reality, this position not always true. Students in pair have to put more effort in creating mutual understanding between them in order to make better decision. This becomes more complex when the partners have different programming abilities and also personality types (Hannay et al., 2010; Hahn et al., 2009; Katira et al., 2004; Williams et al., 2006). Common characteristics of pair partners are important to drive consensus in the decision making. Nevertheless, programmers need to understand each pair partner differences to reach project goals successfully. Understanding others' differences yield more added values and better decision making process in programming tasks.

The results indicate that both pair programming and non-pair programming achieved a statistically significant result in thinking activities. Therefore null hypothesis, H_0 for thinking activities in internationalization knowledge sharing has been rejected. However, two null hypotheses, H_0 for learning and decision making in internationalization knowledge sharing has been accepted because there were no statistically significant different for both groups.

CONCLUSION

This study contributed for better understanding of important knowledge sharing activities to construct student's skills during Internalization process through pair programming. It is undisputed that the pair programming is one of the pedagogical approaches that can enhance students' abilities in the areas of programming. Knowledge sharing in pair programming can be improved with the guidance of lecturers and also increasing the frequency of programming activities between the pairs. Socialization factor is an important factor in ensuring the success of pair programming. Pair programmers need time to understand other's differences. This can lead them to share insights, leap their thought, make soundness decisions, and thus inducing knowledge sharing during programming activities. Further works will be considered rigorous theoretical framework for constructing tacit knowledge among the students in pair programming environment.

REFERENCES

Beck, K. (2005). *Extreme Programming Explained: Embrace Change.* Second edition Reading, Mass: Addison-Wesley.

Brereton, P., Turner, M. & Kaur, R. (2009). Pair programming as a teaching tool: a student review of empirical studies. *Proceedings of the Conference on Software Engineering Education and Training*,22, pp.240-247.

Brockmann, E. N. & Simmonds, P.G. (1997). Strategic decision making: the influence of ceo experience and use of tacit knowledge. *Journal of Managerial Issues, IX*(4), pp.454-467.

Canfora, G., Cimitile, A., & Visaggio. C.A. (2003). Lessons learned about distributed pair programming: what are the knowledge needs to address?," *Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Colloborative Enterprises.*

Chau, T. & Maurer, F. (2004). Knowledge sharing in agile software teams. *Lecture Notes in Computer Science,* vol. 3075/2004, 173-183.

Chong,J. et al. (2005). Pair Programming: When and Why it Works. *17th Workshop of the Psychology of Programming Interest Group*, Brighton, UK.

Cliburn, D.C. (2003). Experiences with pair programming at a small college. *Consortium for Computing Science in College,* vol.19, no.1, pp.20-29.

Fengjie, A., Fei,Q., & and Xin, C. (2004). Knowledge sharing and web-based knowledge-sharing platform. *Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business.*

Gallis, H., Arisholm, E., & Dyba, T. (2003). An initial framework for research on pair programming. *Proceedings of the International Symposium on Empirical Software Engineering.*

Gerard, J.G. (2003). Measuring knowledge source tacitness and explicitness: A comparison of paired items. *Proceedings: 5th Annual Organizational Learning and Knowledge Conference*.

Gerholm, T. (1990). On tacit knowledge in academia. *European Journal of Education*, *25*(3), pp.263-271.

Guthrie, S. (1995). The role of tacit knowledge in judgement and decision making. *Proceedings of the International Conference on Outdoor Recreation and Education*, pp.105-115.

Hahn, J. H. et al. (2009). Assessment Strategies for Pair Programming. *Journal of Information Technology,* vol. 8, pp. 273-284.

Hannay, J.E. et al. (2010). Effects of Personality on Pair Programming. *IEEE Transactions on*

Heywood, J. (1992). The training of student-teachers in discovery methods of instruction and learning and comparing guided discovery and expository method : teaching the water cycle in geography. Technical Report, *Research in Teacher Education Monograph,* 1/92. Dept. of Teacher Education, Dublin University.

Ho, C., Raha, S., Gehringer, E. & Williams, L.(2003). Sangam – A Distributed Pair Programming Plug-in for Eclips. Retrieved 1 August 2010 from http://collaborationngam.pdf.

Ho, C.W. (2003). Tacit knowledge management and pair programming. *CSC591m Term Paper 2003.* Retrieved 1 July 2010 from http://www4.ncsu.edu/~cho/articles/TCMandPP.pdf

Holzman, L. (2009). *Vygotsky at work and play*. NY: Routledge.

Katira,N. et al. (2004). On Understanding Compatibility of Student Pair Programmers. *ACM Technical Symposium on Computer Science Education (SIGCSE)* Norfolk, VA, pp. 7-11.

Mazida, A. (2010). *An investigation of knowledge creation processes in LMS-supported expository and PBL teaching methods*," Universiti Sains Malaysia: Unpublished doctoral dissertation.

Mazni,O. et al. (2009). Being Agile in Classroom: An Improvement to Learning Programming. *Seminar Kebangsaan ICT dalam Pendidikan*, Ipoh, Malaysia.

McDowell, C., Hanks, B. & Werner,L. (2003). Experimenting with pair programming in the classroom. *Proceedings of the 8th annual conference on innovation and technology in computer science education*, 35(3), pp.60-64.

McDowell, C., Werner, L., Bullock,H.E., & Fernald, J. (2003). The impact of pair programming on student performance, perception and persistence. *Proceedings of the 25th International Conference on Software Engineering*, pp.602-607.

Mendes, E., Al-Fakhri, L.B. & Luxton-Reilly, A. (1997). Investigating pair-programming in a 2nd year software development and design computer science course. *Proceedings of the ITiCSE'05*, pp.285-295.

Muller, M. & Tichy, W. (2001). Case study: extreme programming in a university environment. *Proceedings of the 23rd International Conference on Software Engineering,* pp. 537-544.

Natarajan, G. & Shekhar, S. (2001). *Knowledge management; enabling business growth*. Singapore: McGraw-Hill International Edition.

Nonaka, I. & Konno, N. (1988). The concept of ba: Building a foundation of knowledge creation. *California Management Review*, 40(3), pp.40-55.

Nonaka, I. & Takeuchi, H. (1995). *The knowledge creating company: How Japanese companies create the dynamics of innovation*. Oxford: Oxford University Press.

Nunnally, J. C. (1978). *Psychometric theory*. NY: McGraw-Hill.

Slaten, K.M., Droujkova, M.,Beenson, S.B., Williams,L. & Layman, L. (2005). Undergraduate student perceptions of pair programming and agile software methodologies: verifying a model of social interaction. *Proceedings of the Agile Development Conference. Software Engineering,* vol. 36, pp. 61-80.

Sommerville, J. & Craig, N. (2006). *Implementing IT in construction,* NY: Taylor and Francis Group.

Vanhanen, J. & Korpi, H. (2007). Experiences of Using Pair Programming in an Agile Project. *Proceedings of the 40th Hawaii International Conference on system Sciences.*

Vygotsky, L.S. (1978). *Mind in society.*MA: Harvard University Press.

Werner, L.L., Hanks, B. & McDowell, C. (2004). Pair-programming helps female Computer Science Students. *ACM Journal of Educational Resources in Computing,* vol. 4, 1(3).

Williams, L. et al. (2002). In Support of Pair Programming in the Introductory Computer Science Course. *Computer Science Education,* vol. 12, pp. 197-212.

Williams, L. et al. (2006). Examining the Compatibility of Student Pair Programmers. *Agile Conference 2006*, Minneapolis, MN, pp. 411-420.

Williams, L.A. & Kessler, R.R. (2000). The effects of "pair-pressure" and "pair-learning" on software engineering education. *Thirteenth Conference on Software Engineering Education and Training*, pp.59-65.

Yin, T.S. & Zhang, Q. (2005). Dynamic game analysis in worker's tacit knowledge Sharing process in enterprise. *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics,* Guangzhou.