KNOWLEDGE SHARING PROCESS FOR PAIR PROGRAMMING PRACTICE AMONG UNIVERSITY STUDENTS

Syamsul Bahrin Zaibon[1], Mazida Ahmad[2], Mazni Omar[3], and Azida Zainol[4]

College of Arts and Sciences, Universiti Utara Malaysia, Malaysia
[1]syamsulbahrin@uum.edu.my; [2]mazida@uum.edu.my; [3]mazni@uum.edu.my; [4]azida@uum.edu.my

ABSTRACT. Pair programming practice has been widely used as a pedagogical approach in educational setting specifically in the programming course. Most pair programming studies agree that this practice can foster knowledge sharing among students. However, the studies do not highlight knowledge sharing during pair programming practices and towards their performance. Therefore, the main aim of this study is to test the suitability of knowledge sharing activities by using Socialization, Externalization, Combination, and Internalization (SECI) model in pair programming practices (PPP) amongst students in higher learning institution. In achieving the main aim, this study involved experimentation in order to test the effectiveness of the SECI model. 202 participants were actively engaged in pair programming practices, and involved in the experimentation of the study. The findings reveal that there are significant direct effect of Socialization towards Externalization, Externalization towards Combination and Combination towards Internalization. However, the direct effect of Internalization towards Performance are not significant.In addition, the direct effect of SECI model for pair programming practices are not significant towards the performance of the students.

Keywords: pair programming, knowledge sharing, knowledge management process, SECI model, university student

INTRODUCTION

Pair programming changes the programming learning from individual activity into a collaborative learning process as stated by McDowell, Werner, Bullock, and Fernald (2003). They describe that pair programming in learning involves two students who act as a driver and a navigator working on the same problem from the design to the testing phase. A driver is a person who creates and implements code, whereas a navigator is will check errors and suggest the implementation technique. Navigator also provides alternative solution to a particular problem and assists the drivers to solve that problem. Meanwhile, driver has fully control all input devices and gives solution based on his/her idea or navigator's suggestions (Williams & Kessler, 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 (Beck, 2005; Chau & Maurer, 2004). This leads to exchange or spread information and knowledge throughout the whole team of software development (Muller & Tichy, 2001). In pair programming practice for learning, it can foster knowledge sharing among students. Many studies have been done with pair program-

ming in education settings, however few studies focus on the correlation between knowledge sharing and pair programming (Vanhanen & Korpi, 2007; Venkatesan & Sankar, 2010).

Generally, knowledge transfer involves knowledge sharing from the starting point of knowledge creation to the point of knowledge application, which is also obtained by adapting several cyclic activities namely mobilizing knowledge, knowledge searching, knowledge distributing, knowledge sharing, and knowledge pulling and pushing (Gover & Davenport, 2001). According to Fengjie, Fei, and Xin (2004), the knowledge-sharing process involves two main parties namely the contributor and the receiver. This scenario is similar to the pair-programming practices where the navigator plays the role of a contributor and the driver the role of a receiver. Knowledge sharing in pair-programming practices involves communication, updates, advice, problem-solving, decision-making, discussion over project data and information (Komchaliaw & Wongthongtham, 2010).

SECI model (Nonaka & Takeuchi, 1995) was chosen in this study because it is suited to be applied in programming environment. Several studies (Kutay & Aurum, 2007; Allan, 2015; Siadat, Matinvafa, Saeednia, & Moghadasi, 2015) discuss the processes in SECI model in educational settings without emphasizing on the tacit knowledge obtained at the end of the process. SECI model is adopted in this study to simplify knowledge transformation between tacit and explicit knowledge and to promote knowledge sharing between students (partner) during pair programming practice in learning environment.

Four processes in the model that would be the variables for this study are Socialization, Externalization, Combination, and Internalization. Socialization is a process of sharing experiences between partner and then creating tacit knowledge. While, Externalization means the process of expressing tacit knowledge into written form or explicit knowledge but still in an inconsistent condition. This knowledge can be shared by others and become the basis of the new knowledge. Next, Combination refers to the process of transforming explicit knowledge that is inconsistent into a systematic sets of explicit knowledge. After that, during Internalization process, the experiences will be converted into individual knowledge (Nonaka & 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 partners. Thus SECI model is applicable to promote sharing and constructing knowledge between partners in generating learning, thinking and decision making skills.

METHODOLOGY

The main aim of this study is test the suitability of SECI model in programming practices amongst students in higher learning institution. Therefore, to achieve the main aim, this study involved experimentation in order to test the effectiveness of the SECI model. The SECI model is based on Nonaka and Takeuchi (1995), whilst the experimentation process was adapted from Wohlin et al. (2000).

In Figure 1, independent variables refer to the causes of the study which is the pair programming practice (PPP). The independent variable was undergoing the experimentation process in order to evaluate the SECI model and performance. The effect was measured as dependent variables, which are constructs in the SECI model; socialization, externalization, combination, and internalization and performance. Non probability sampling which is purposive sampling was used in this study because as the consideration of the researcher in obtaining a representative sample of the population (Lavrakas, 2008). The sample of the study consisted of 202 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.

Experiment for pair programming practice was performed in Week 7 during the lab session. Participants were briefed on the PPP before the experimentation. The students were also emphasized on the role of each participant in order to ensure that they are conformed to the PPP applied. During pair formation, students were given the freedom to choose their partner. This is because conflict can be reduced when members are comfortable with each other, and thus provides better learning experiences. Past empirical evidence suggested that self-selection of team members was more effective than random formation (Scott & Pollock, 2006). Next, the instructor distributed programming test question to them. The instructor collected the programming test answer from the participants after 2 hours of the experiment. As mentioned earlier, two hours were allocated because the assignment is suitable to be solved in the time given. Then, the knowledge sharing questionnaires were distributed just after the participants submitted their programming test answer.

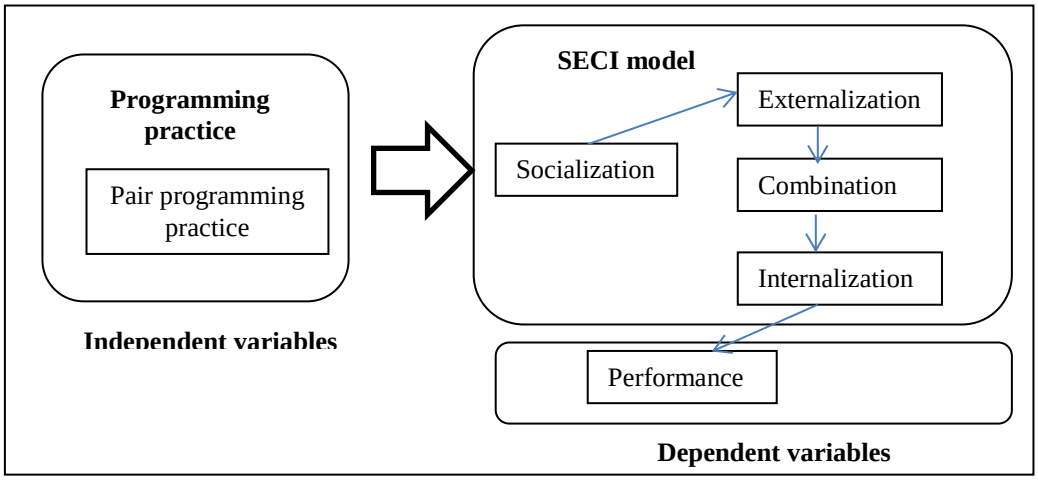

Figure 1. Overview of Experimentation for Model Evaluation

The instrumentation provides means to perform and monitor the experiment. In this study, a questionnaire was used. The questionnaire was adapted from SECI model in educational context which includes Socialization, Externalization, Combination and Internationalization (Mazida, 2010). The first dimension of the instrument contains 31 items on system infrastructure and architecture which is tailored to software engineering education environment. Meanwhile, dimension two contains 41 items asking about the learners, lecturers, teaching materials, and the outcome which are tailored towards independence of learning, independence of thinking, and independence of decision making. A five (5) point Likert scale from 1 to 5 was used in the instrument which defined as 1 refers to strongly disagree and 5 refers to strongly agree.

In assessing empirically the effect of knowledge sharing amongst students using pair programming practice, four hypotheses have been formulated (Table 1). Hypotheses testing were carried out using Structural Equation Modeling which was used to test fit of the derived model with the experimental data. It is a technique of analysis that includes measurement error to comprehend other influencing indicators (Conley, Muncey, & You, 2005; Tabri & Elliott, 2012). In this study, PLS based is employed, utilizing SmartPLS 2.0 as the tool. This is because PLS can be used to avert the limitations of co-variance-based SEM with regards to dis-

tributional properties, measurement level, sample size, model complexity, identification and factor interdependencies (Chin, 1998). Covariance based SEM like AMOS is used to test or confirm on the existing theory or model, but PLS can be used for theory confirmation or theory development, which include using to develop prepositions by exploring the relationship between variables (Chin, 1998). Since the model in this study is conceptualized based on literature review, then PLS is applicable.

Table 1. Hyphotheses

No.	Hyphotheses
H1	Learners' interaction in socialization process contribute positively significantly to the manipulation of tacit knowledge into more understandable forms in the externalization process in the pair programming practice.
H2	The externalization process contributes positively significantly to the complex and structured knowledge in the combination process in the pair programming practice.
H3	The combination process contributes positively significantly to the improvement of learners' tacit knowledge in the internalization process in the pair programming practice.
H4	The internalization process contributes positively significantly to students' performance in the pair programming practice.

FINDINGS

202 undergraduate students at Universiti Utara Malaysia (UUM) enrolled in Basic Programming course were sampled for this study. Figure 2 summarizes the demographic profiles of the respondents.

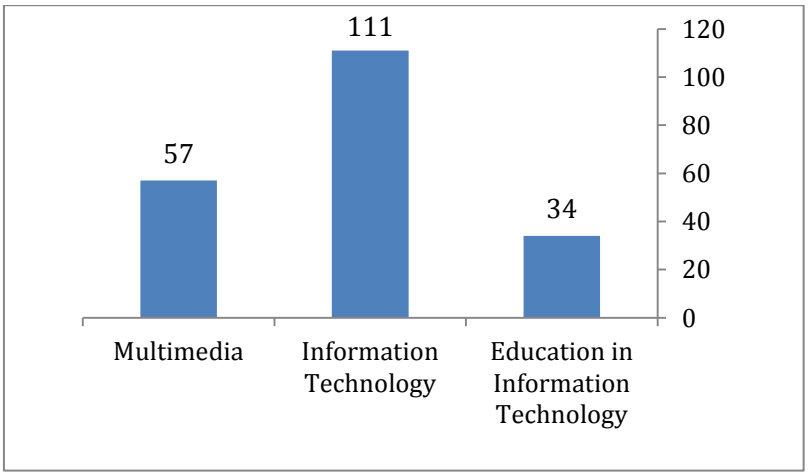

Figure 2. Respondents' demographic background: Programme

Construct Validity, Convergent Validity, and Reliability Tests

From the construct validity tests, all the items measuring a particular construct loaded were high on that construct and loaded lower on the other constructs. Thus, confirming the construct validity for PPP Model. In terms of convergent validity, it is conducted to test the degree to which multiple items to measure the same concept are in agreement. The results of the convergent validity depict the degree to which the construct indicators indicate the latent variable, and the construct ranged from 0.5078 to 1. The AVE measures the variance captured by the indicators relative to measurement error. Fornell and Bookstein (1982) stated that AVE should be greater than 0.5 to justify using the construct. In PPP model, the values for AVE for each constructs are above 0.5 and ranged from 0.5034 to 1. The results showed that all the

constructs are all valid measures based on their parameter estimates and statistical significant (Chow & Chan, 2008). To assess the inter item consistency of the measurement items for each model; Cronbach's alpha coefficient is used. From the test, all Cronbach's alpha values are above 0.7 as suggested by Nunnally (1978). The composite reliability values are ranged from 0.8076 to 1.0000 for PPP Model. Interpreted like a Cronbach's alpha for internal consistency reliability estimate, a composite reliability of 0.70 or greater is considered acceptable (Fornell & Bookstein, 1982). Therefore, it can be concluded that the measurements are reliable.

Analysis and Interpretation

Table 2 shows the result of the structural model for PPP model. There are significant direct effects of Socialization towards Externalization, Externalization towards Combination and Combination towards Internalization. The coefficient values are ranging from (β = 0.3221-0.5655, t= 4.6552-11.9378, p <0.01). However the direct effect of Internalization towards Performance is not significant (β = 0.0465, t= 0.6175, p >0.01). Thus, H1, H2, and H3 were supported and H4 is not supported.

Table 2. Path Coefficient and hypothesis testing for PPP Model

Hypothesis	Relationship	Coefficient	*t* value	Supported
H1	Soc -> Ext	0.5655	11.9378	SUPPORTED
H2	Ext -> Com	0.3221	4.6552	SUPPORTED
H3	Com -> Int	0.452	9.2774	SUPPORTED
H4	Int -> Perform	0.0465	0.6175	NOT SUPPORTED

Note: Soc=Socialization, Ext=Externalization, Com=Combination, Int=Internalization, Perform=Performance

The findings reveals that the significant direct impacts of Socialization towards Externalization, Externalization towards Combination and Combination towards Internalization for SPP Model is greater than PPP model. It also indicates the direct impacts of Internalization towards Performance are not significant for both models.

DISCUSSION

The study aimed to test the suitability of SECI model in programming practices among students in higher learning institution. At the end of this study, the main aim has been achieved. Variables were determined where itrefer to the causes of the study which is type of programming practices, PPP. The independent variable was undergoing the experimentation process in order to identify the relationship or each SECI model constructs. The relationship was measured which are constructs in the SECI model; *socialization, externalization, combination,* and *internalization.*

In PPP, there are significant direct effects of Socialization towards Externalization, Externalization towards Combination, and Combination towards Internalization. The coefficient values are ranging from (β = 0.3221-0.5655, t= 4.6552-11.9378, p <0.01). Therefore, H_1, H_1, and H_3 were supported.The relationship was measured between *internalization* towards *performance* for each programming practices. In PPP the direct effect of Internalization towards Performance is not significant (β = 0.0465, t= 0.6175, p >0.01). Therefore, hypothesis of H_4 is not supported.

To summarize, the research findings reveal that there are significant direct impacts of Socialization towards Externalization, Externalization towards Combination, and Combination

towards Internalization for PPP. However, the findings also indicate that the direct impact of Internalization towards Performance is not significant for PPP.

CONCLUSION

This study adapted SECI model by adding a new construct which is a performance. Educationist and software practitioners would get benefits from this model, in which they can refer to for applying the model in their programming environment. The uniqueness of the model lies in the educational settings which focus on the KM processes (Socialization, Externalization, Combination, and Internalization). This study contributed to better understanding of important knowledge sharing activities to construct students' skills during the Socialization, Externalization, Combination and Internalization (SECI) process through pair programming. It is undoubted that pair programming is one of the instructional approaches that can enrich student's capabilities in the areas of programming.

REFERENCES

Allan, D. (2015). I think, therefore I share: Incorporating Lesson Study to enhance pedagogical knowledge exchange. *Educate~, 15*(1), 2-5.

Beck, K. (2005). *Extreme programming explained: Embrace change*. (2nd ed.). Reading, Mass: Addison-Wesley.

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

Chin, W. W. (1998). Issues and Opinion on Structural Equation Modelling. *MIS Quarterly, 22*(1). 7-16.

Chow, W. S. & Chan, L. S. (2008). Social Network and Shared Goals in Organizational Knowledge Sharing. *Information and Management, 45*(7), 458 – 465.

Conley, S., Muncey, D. E. & You, S. (2005). Standards-based evaluation and teacher career satisfaction: a structural equation modeling analysis. *J Pers Eval Educ*, 18, 39-65.

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

Fornell, C., & Bookstein, F.L. (1982). Two Structural Equation Models: LISREL and PLS applied to Consumer Exit-Voice Theory. *Journal of Marketing Research*, 19(4), 440 – 452.

Gover, V., & Davenport, T. H. (2001). General perspectives on knowledge management: Fostering a research agenda. *Journal of Management Information Systems, 18*(1), 5-21.

Tabri, N., & Elliott, C. M. (2012). Principles and practice of structural equation modeling. *Canadian Graduate Journal of Sociology and Criminology*, *1*(1), 59-60.

Komchaliaw, S., & Wongthongtham, P. (2010). *A state of the art review on software project performance management*. In Procs. of the 4th IEEE International Conference on Digital Ecosystem and Technologies (IEEE DEST2010), pp. 653-655.

Kutay, C., & Aurum, A. (2007). Knowledge transformation for education in software engineering. *International Journal of Mobile Learning and Organisation*, *1*, 58-80.

Lavrakas, P. J. (2008). *Encyclopedia of survey research methods.* LA: SAGE Publications, Inc.

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

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

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

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

Siadat, S. H., Matinvafa, A., Saeednia, A., & Moghadasi, F. (2015). Effective Factors on Successful Implementation of Knowledge Management in Higher Education. *Management and Administrative Sciences Review*, *4*(1), 166-181.

Scott, E., & Pollock, M. (2006). Effectiveness of self-selected teams: A systems development project experience. *Issues in Informing Science and Information Technology, 3*, 601-617.

Vanhanen, J., & Korpi, H. (2007). Experiences of using pair programming in an agile project. In *Procs. of the 40th Hawaii International Conference on System Sciences.*

Venkatesan, V., & Sankar, A. (2010). Adoption of Pair Programming in the Academic Environment with Different Degree of Complexity in Students Perspective–An Empirical Study. *Int J Eng Sci Technol, 2*, 4791-4800.

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

Wohlin, C., Runeson, P., Host, M., Ohlsson, M. C., Regnell, B., & Wesslen, A. (2000). *Experimentation in software engineering: An introduction*. USA: Kluwer Academic Publishers.