

Knowledge Sharing and Pair Programming: Are They Related?

Mazida Ahmad¹, Mazni Omar¹, Azman Yasin¹, Rohaida Ramli¹, Massudi Mahmuddin¹, Ariffin Abdul Mutalib²
¹*School of Computing, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia.*
²*School of Multimedia Technology and Communication, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia.*
 mazida@uum.edu.my

Abstract—Pair programming (PP) software development technique is recommended for improving programming skills using concept of knowledge sharing. This paper aims to elaborate knowledge sharing concepts based on review of several previous studies. The relationship of knowledge sharing concepts in PP also has been discussed in details. With the elaboration, knowledge sharing concept can be anticipated to demonstrate the relationship of knowledge sharing in PP.

Index Terms—Pair Programming; Tacit Knowledge; Agile.

I. INTRODUCTION

Knowledge has become necessary for human life in order to construct information and arrange data [1]. Therefore, in order to deliver the knowledge, it is important to facilitate process of transferring and sharing especially to academicians in all knowledge area. This is crucial due to academicians must established their knowledge as required in the highest institution. Thus, they need to deliver their knowledge to students by teaching and learning processes.

Computer science (CS) and software engineering (SE) courses is the highest in dropout rates among students compared to the other courses [2]. This finding motivates the practitioners to urgently employ a technique or practice that can facilitate teaching and learning practices in CS/SE courses.

The critical point of knowledge development in learning and teaching on CS/SE course is programming skill. Enhancing the programming skills of the students in higher learning institutions is the main concern in this study. An appropriate code quality is an indicator to a code quality there was essential to foster the personal knowledge of the students. On that origin the idea has been investigated to create a conceptual model that can enhance the programming skills among students of higher learning institutions which is the main concern of this study. Developing the conceptual model came up with 2 necessary components which are:

- Employing a well-known model that arranges knowledge management with influences on personal knowledge of the individuals which is tacit knowledge,
- Using a practice that is reliable in CS and SE community that deals with knowledge management and fosters tacit knowledge.

For the first component, this study employed the model of Nonaka and Takeuchi [3] which is Socialization, Externalization, Combination and Internalization (SECI). Then, for the second component, this study employed pair

programming (PP). Pedagogically, employing PP to solve programming problems is highly related to the main concern of this paper as shown in Figure 1.

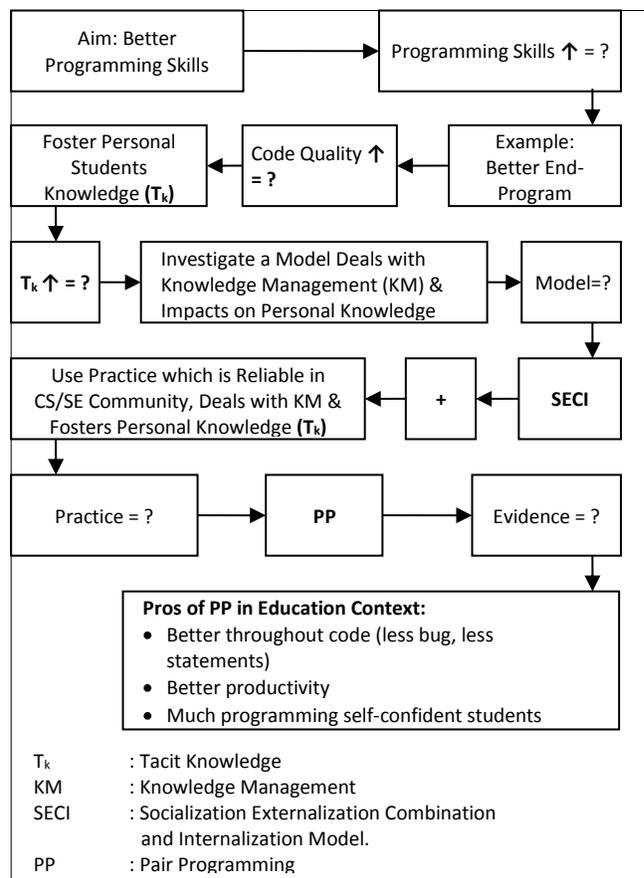


Figure 1: Conceptual relationship of knowledge sharing in pair programming

In Figure 1 clearly illustrated the conceptual relationship of knowledge sharing in PP for enhancing programming skills. Further, the diagram also highlighted the keywords operationally uses as the main focus in this paper which are PP, tacit knowledge, and code quality.

Generally, relationship of PP, tacit knowledge and code quality is part of knowledge that has been used in extreme programming practices of Agile software development. While PP is focusing on the process, but simultaneously it is able to cooperate in computer solving problem. Tacit knowledge are

focusing in learning innovation, skills, and memory of personal knowledge, while code quality focusing in number and measurement usability of level program.

The relationship of this knowledge illustrated in the Figure 2. This figure provides the relationship among keywords on the basis of understanding and the way by which they may be subordinate and interact. Knowledge consists of Explicit and Tacit knowledge while Agile consist of Pair Programming (PP) and Code Quality (CQ). This paper focuses on the exploration of knowledge sharing keyword and relationship of PP towards the knowledge sharing.

II. REVIEWS

This section focuses on concepts related to the relationships keywords that had been illustrated in Figure 2.

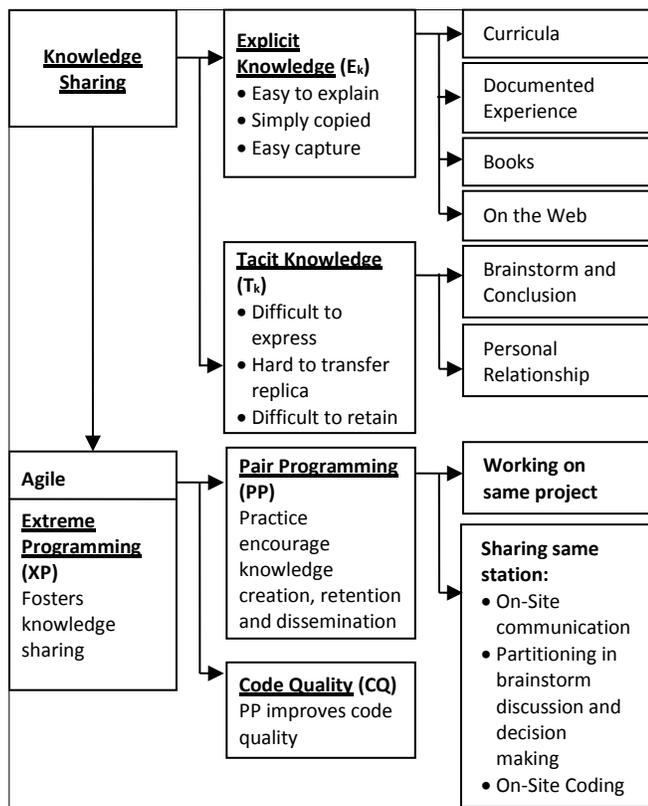


Figure 2: Relationship of knowledge sharing keywords

A. Knowledge Sharing

In 1958, Michael Polanyi [4] has ignited the classifications of knowledge as tacit and explicit. Tacit knowledge is characterized as the individual experience and expertise of a person that is hard to be described and understood by others [5]. Then, it is categorized as the ability of intention and decision making. It has also been defined as an applied acquisition of knowledge in a constitution that the information is not stated which makes it very challenging to carry on because it is unclearly taught or expressed. Applied knowledge is gained by a person through daily basis experience instead of through official instruction. Factors influencing tacit knowledge (TK) include what the human has mentally ratified during learning phase [6].

Besides, difficulty to be expressed, tacit knowledge also complicated to be transferred due to the variation in formulation of speech and understanding [7]. This resulted difficulties to retain the transferred knowledge. As compared to tacit, explicit knowledge can be transformed in the form of words and can be communicating through email and data that related to tangible resources [8], [9]. It is supported by archived information such as curricula, documented experiences [10], and books and web pages that also can be sources of tacit knowledge [11]. Thus, explicit knowledge is easier to be explained, copied, captured and can be divulged easily [12], [13].

Persistent revolution of new knowledge in knowledge process, such as knowledge acquisitions as well as knowledge sharing that will be continuously studied all the time. Knowledge are created and utilized through explicit and tacit knowledge intermingling, which is referred to as knowledge conversation techniques. Researchers also found that almost all the knowledge is tacit while, only a small portion of knowledge is explicit.

Based on knowledge category, this paper only focuses on tacit knowledge. Tacit knowledge defined as personal knowledge deposited in people's brain as a consequence of expertise, imagination, learning, innovation, skills, and memory. In teaching and learning process, tacit knowledge transfers development between lecturers and students is very crucial especially in retaining the prior and acquiring the new knowledge. It is worth to be mentioned that robust Socialization of expert lecturer, strong Externalization and Combination based on various references will be positively impact the Internalization of tacit knowledge in learning, thinking, and making decision skills [14]. Socialization, Externalization, Combination, and Internalization also known as SECI model which has become the foundation of knowledge creation and transfer theory [15].

Gerholm [16] has appealed that tacit knowledge consists of two types of knowledge; (1) knowledge as a reflection of daily life within educational institution regarding teaching and learning process taught by lecturers. The second category is knowledge gained by students directly or indirectly as it is gained from what students have learned from the lecturer within the educational organization or knowledge that is created from discussions and interpretation among students.

In addition, there is correlation between tacit and explicit knowledge in education between lecturers' and students' knowledge [17]. Tacit knowledge refers to lecturer's ideas and expertise, whereas explicit knowledge is signified by lectures through computer, over the Internet, and saved in databases [18]. According to Leonard and Insch [19] there are 3 categories of tacit knowledge which are; (1) self-organization and self-motivation (referred to as cognitive), (2) individual and institutional tasks (denoted as technical), and (3) social (reflects social interaction).

B. Agile

There are several studies of eXtreme Programming implementation techniques including interpretation and enhancement team communication [20]-[22]. eXtreme Programming applied in order to enhance the capability of Agile project, increase qualitative of commitment, and interaction of team members and compliance of experts during

project development. The interaction among human, coordination and cooperation in self-motivated team is very important [23].

In response to the concerns on ideational techniques, which comprise waterfall technique, Agile has been a frivolous model for software development [23]-[25]. Therefore, Agile meets the volatile desires of stakeholders that distribution of knowledge is a substantial way to overcome challenges in establishing the systems. To distribute a creative work, respond to volatility, deal with frequently changing demands of stakeholders, Agile team must be shielded with cross-functional members through collaboration of customers with developers and their interaction as well as meetings frequently.

According to Crawford *et al.* [26], agile has achieved a universal recognition for its potentials in enhancing software team efficiency in numerous degrees by way of encouraging team's focus atmosphere, inspiring individuals, and more focusing on the clarity and outcomes of a project.

Moreover, Agile teams based on Dorairaj *et al.* [4] overlap efficient teams that improve knowledge sharing about certain project by means of physical and effective discussion as well as partnership with stakeholders. However, normally the Agile teams are struggling to be well-informed and shared knowledge among team members towards the accomplishment of software projects.

As the popularity of eXtreme Programming has improved, the training of PP has continued to attract more attention in the eXtreme Programming, granting PP opportunity to attract programmers' attention. This is because it inspires the creation of tacit knowledge among team members and encourages knowledge sharing specifically through release and iteration planning, PP, and on-site customers.

C. Pair Programming

Pair Programming (PP) is a cooperative programming mode of eXtreme Programming practices of Agile software development family. Feature that distinguishes PP from further collaborative programming styles is the terms: "driver", "navigator" and the technique they adjust to tasks process, who sit at the same workstation with only one set of screen, mouse, and keyboard. The two persons are imposed to design, code, diagnose, and develop a project [27]-[29]. In the practice, both programmers enthusiastically interact among them utilizing role-base procedure [30].

Knowledge sharing has been observed as the main part of Agile because of its' basic expectation towards high quality and valuable software in brief statement on tacit knowledge, which are built among the teams in charge of a project through physical relations in order to enhance competitive benefits towards the customers, as well as traverse efficient teams. Due to the fact, knowledge sharing is vital in this paradigm because of brief and repetitive view and minimal records [23].

PP contains two individual programmers (the drivers and the navigators), acting as a team through similar algorithms, design, code, and test using the same computer. The driver is responsible inputting device in order to produce the code. Meanwhile, the navigator frequently and enthusiastically assessing the work of the driver to see if there is any flaw, consider other substitute, think through strategic inferences, as well as asking questions. Then, recognizing the strategic

paucity in the process of coding is also the role of the navigator. However, the roles as the navigator and the driver often swap, to improve their work in one way or another by practicing and learning appropriate skills when there are changes in their work routine, which occurs at the instance of natural transition during the coding pursuit [29].

One of the outcomes of PP is shorter length of code. Shorter length of code is introduced as an indication to improvement in underlying design. Besides, PP reduces rate of defect [27], [31] thus, a large number of successful test cases were achieved when utilizing PP.

D. Code Quality

Code quality is an indicator for less number of defects in syntax and it measures the receipt level of a program among users in terms of reliability, usability, maintainability, and portability [32].

Then, literature explains that expert opinion, effectiveness, academic performance, and number of successful test cases also measures code quality [33]. However, researchers also found that agile concept is a crucial factor towards achieving better software quality. Consequently, this relies on expert opinion to measure quality in terms of correctness criteria.

III. RELATIONSHIP OF AGILE SOFTWARE DEVELOPMENTS WITH KNOWLEDGE SHARING

The sharing of knowledge has been viewed as the main part of Agile because of its basic anticipation towards high quality and valuable software in brief statement on tacit knowledge, which are built among the teams in charge of a project through physical interactions in order to enhance competitive benefits towards customers, as well as traverse efficient teams.

Based on brief and repetitive feature and minimal records, knowledge sharing as an essential undertaking in Agile [23]. However, Agile methods including eXtreme Programming promote collaborations and stress physical tacit sharing of knowledge within the teams and their clients or stakeholders [34].

IV. PAIR PROGRAMMING AND KNOWLEDGE SHARING

All Agile practices concern on the management of tacit Knowledge. PP tactics highly sustenance knowledge sharing towards knowledge achievement and sharing, in which the individual relationship and association is very paramount in the CS context [35]. Though with high financial consequence, PP has been acknowledged as a critical tool for knowledge sharing among members of a project team. In fact, Kashif and Kelly [36] said that formal knowledge sharing in a workshop or project assessment meeting has also been valuable in improving skill and member's capabilities.

Knowledge sharing is an important technique in knowledge management that progressively improves the performance of an organization [23]. Long-established organizational effectiveness and accomplishment resulted from knowledge sharing abilities to overcoming problems, carrying out policies and techniques, acquiring new initiative through expertise collaboration, and supplying relevant information about the task to be accomplished. Nevertheless, the knowledge seekers

and the knowledge provider must be in a settlement in order to share the knowledge. Table 1 illustrates the comparison of PP with knowledge sharing.

Based on Table 1, when conducting a PP session, the driver and the navigator change idea in common tacit knowledge and explicit knowledge. And also as mentioned in previous sections concerning ease of utilizing the explicit knowledge in terms of ease of understand, explicit knowledge can be seen in the form of numerals and words [3]. However, tacit knowledge is fed by opinion, expert, and think of a human as well as strategy of decision making. Tacit knowledge also is a reflection of learning experience.

Table 1
Pair programming and knowledge sharing

Pair programming	Knowledge Sharing
Two partners: navigator and driver	Two partners: contributor and receiver
Navigator: participates with the driver by addressing alternative ideas in attempts to solve the problem. The best alternative will be selected.	Contributor: subscribes some of his expertise and sends it to the receiver, who will add it to his own knowledge base.
Driver: writes codes brainstormed with the navigator.	Receiver: guided by the contributor.
Partners share place and computer to solve problems unless they use distributed PP (no time and place limitations).	Contributor and receiver communicate over knowledge sharing space (e.g., .net meeting, team viewer)

Therefore, tacit knowledge can be acquired along with PP practices between the driver and the navigator to produce skills of learning, thinking, and decision making [35].

V. PAIR PROGRAMMING AND CODE QUALITY

One of the consequences of PP is the shorter length of code that will be an indication of improvement in underlying design. This is due to communication improvement between team members. Besides that, PP also is able to improve team spirit, and sharing knowledge. Therefore, PP has the ability to reduce rate of defect. Researchers also agree that a large number of successful test cases were achieved when operating PP [27], [31]. As results, PP implementation will increase software development productivity by reducing costs of defects and product risk.

VI. CONCLUSION

This paper presented the theoretical study for knowledge sharing towards CS and SE courses. The elaboration and discussion of multiple reviewed categories in knowledge sharing, Agile, PP, and code quality also has been justified. The relationship of knowledge sharing towards agile software development also has been clarified in this study. In general, this paper highlighted the conceptual PP and knowledge sharing by comparing the knowledge transfer process in software development.

ACKNOWLEDGMENT

Our utmost gratitude goes to the Ministry of Education Malaysia for supporting us by funding the Fundamental

Research Grant Scheme (FRGS/SO CODE: 12814). We would also like to thank the Universiti Utara Malaysia for the support and facilities provided in order to complete this research.

REFERENCES

- [1] Yi, J. 2006. Externalization of tacit knowledge in online environments. *International Journal on E-learning*. Vol. 5(4), pp. 663-674, 2006.
- [2] Md Rejab, M., Omar, M., & Ahmad, M. "Knowledge internalization in pair programming practices". *Journal of Information and Communication Technology (JICT)*. Vol. 11, pp. 163-177, 2012.
- [3] Nonaka, I., & Konno, N. "The Concept of "Ba": Building A Foundation For Knowledge Creation". *California Management Review*. Vol. 40(3), 1998.
- [4] Dorairaj, S., Noble, J., & Malik, P. "Knowledge Management in Distributed Agile Software Development". *Proceedings of the Agile Conference (AGILE)*, 2012.
- [5] Guthrie, S. 1996. The Role of Tacit Knowledge, Judgement and Decision Making.
- [6] Billett, S., & Choy, S. "Learning through work: emerging perspectives and new challenges. *Journal of Workplace*". *Learning*. Vol. 25(4), pp. 264-276, 2013.
- [7] Dudley, P. "Teacher learning in Lesson Study: What interaction-level discourse analysis revealed about how teachers utilised imagination, tacit knowledge of teaching and fresh evidence of pupils learning, to develop practice knowledge and so enhance their pupils' learning". *Teaching and Teacher Education*. vol. 34, pp. 107-121, 2013.
- [8] Marques-Lucena, C., Agostinho, C., Koussouris, S., Sarraipa, J., & Jardim-Goncalves, R. Collaborative Knowledge Management Using Wiki Front-End Modules. *17th International Conference*, 2015.
- [9] Wan, J., Wan, D., Luo, W., & Wan, X. "Research on Explicit and Tacit Knowledge Interaction in Software Process Improvement Project". *JSEA*. Vol. 4(6), pp. 335-344, 2011.
- [10] Murphy, G., & Salomone, S. "Using social media to facilitate knowledge transfer in complex engineering environments: a primer for educators". *European Journal of Engineering Education*. vol. 38(1), pp. 70-84, 2013.
- [11] Foss, N. J., Lyngsie, J., & Zahra, S. A. 2013. The role of external knowledge sources and organizational design in the process of opportunity exploitation. *Strategic Management Journal*.
- [12] He, W., Qi, C., Xu, X., & Guiyang, P. 2013. Linking Knowledge Sharing and Employee Creativity: Decomposing Knowledge Mode and Improving the Measure of Tacit Knowledge Sharing.
- [13] Ahmad, M., Md Rejab, M., Syazwan Abdulah, M., Omar, M., Bariah Ahmad, K., & Abbas, M. 2012. Measuring tacit knowledge acquired during problem based learning teaching method in learning management system environment. *AWER Procedia Information Technology and Computer Science*, 1.
- [14] Allal-Chérif, O., & Makhlof, M. "Using serious games to manage knowledge: The SECI model perspective". *Journal of Business Research*. Vol. 69(5), pp. 1539-1543, 2016.
- [15] Singh, A., Singh, K., & Sharma, N. 2013. Knowledge Management: the agile way. *Proceedings of the Information and Knowledge Management*.
- [16] Gerholm, T. "On tacit knowledge in academia". *European Journal of Education*. pp. 263-271, 1990.
- [17] Dalkir, K. 2013. *Knowledge management in theory and practice*. Routledge.
- [18] Liaw, S.-S., Huang, H.-M., & Chen, G.-D. "Surveying instructor and learner attitudes toward e-learning". *Computers & Education*. vol. 49(4), pp. 1066-1080, 2007.
- [19] Leonard, N., & Insch, G. S. "Tacit knowledge in academia: a proposed model and measurement scale". *The Journal of Psychology: Interdisciplinary and Applied*. Vol. 139(6), pp. 495-512, 2005.
- [20] Ellis, G. 2016. *Chapter 8 - Agile Project Management: Scrum, eXtreme Programming, and Scrumban Project Management in Product Development* (pp. 223-260). Boston: Butterworth-Heinemann.
- [21] Fojtik, R. "Extreme Programming in development of specific software". *Procedia Computer Science*. 3:1464-1468, 2011.
- [22] Haryono, K. 2015, 4-6 Nov. 2015). The extreme programming approach for financial management system on local government. *International Conference on Science and Technology (TICST)*, 2015.
- [23] Souza, C. R. D. 2012. Fostering Inter-Team Knowledge Sharing Effectiveness In Agile Software Development.

- [24] Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. "A decade of agile methodologies: Towards explaining agile software development". *Journal of Systems and Software*. 85(6):1213-1221, 2012.
- [25] Santos, V., Goldman, A., & Roriz Filho, H. "The influence of practices adopted by agile coaching and training to foster interaction and knowledge sharing in organizational practices". *Proceedings of the 46th Hawaii International Conference on System Sciences (HICSS)*, 2013.
- [26] Crawford, B., de la Barra, C. L., Soto, R., Dorochesi, M., & Monfroy, E. 2013. *The Role of Knowledge Management in Agile Software Development*. HCI International 2013-Posters' Extended Abstracts. Springer. 17-21.
- [27] di Bella, E., Fronza, I., Phaphoom, N., Sillitti, A., Succi, G., & Vlasenko, J. 2013. Pair Programming and Software Defects-A Large, *Industrial Case Study*.
- [28] Hannay, J. E., Arisholm, E., Engvik, H., & Sjöberg, D. I. K. "Effects of Personality on Pair Programming". *IEEE Transactions on Software Engineering*. Vol. 36(1), pp. 61-80, 2010.
- [29] Porter, L., Guzdial, M., McDowell, C., & Simon, B. "Success in introductory programming: what works?" *Communications of the ACM*. Vol. 56(8), pp. 34-36, 2013.
- [30] Williams, L. "Lessons learned from seven years of pair programming at North Carolina State University". *ACM SIGCSE Bulletin*. 39(4), pp. 79-83, 2007.
- [31] Jensen, R. "A pair programming experience". *The journal of defensive software engineering*. Vol. 16(3), pp. 22-24, 2003.
- [32] Omar, M., Romli, R., & Hussain, A. 2007. *Automated Tool to Assess Pair Programming Code quality*.
- [33] Salleh, N., Mendes, E., & Grundy, J. "Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review". *Software Engineering, IEEE Transactions*. 37(4), pp. 509-525, 2011.
- [34] Beck, K. 2001. *Planning extreme programming*: Addison-Wesley Professional.
- [35] Rejab, M. M., Omar, M., & Mazida Ahmad, K. B. A. 2011. *Pair Programming in Inducing Knowledge Sharing*.
- [36] Kashif, M., & Kelly, K. 2013. *Knowledge Management and Sharing Within Project Teams: A qualitative Study of Ericsson*. Mälardalen University.