Journal of Engineering and Applied Sciences 13 (6): 1310-1315, 2018
ISSN: 1816-949X

Understanding Knowledge Sharing in Pair Programming Practice

[1]Mazida Ahmad, [1]Mazni Omar, [1]Azman Yasin, [1]Rohaida Ramli and [2]Ariffin Abdul Mutalib
[1]School of Computer Computing, College of Arts and Sciences,
[2]School of Multimedia Technology and Communication, College of Arts and Sciences,
Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia

Abstract: Promoting knowledge sharing among students using Pair Programming (PP) is the main focus of this study. PP that pairs a driver and a navigator is expected to improve student's social skills to work creatively and innovatively in resolving difficulties in programming. Based on that this study employs PP in advancing the programming skills among students of higher learning institutions. Beyond PP, this study integrates the solution with Socialization, Externalization, Combination and Internalization (SECI) Model. The main objective is to investigate the relationships between each of the four stages in SECI Model and code quality. A group of students were paired and assigned a task. Several benefits of PP have been discovered based on performance statistic. Obviously, the code quality is improved as a result of knowledge sharing, transfer process and communication between the pair. In the end, it is found that externalization has no significant relationship with code quality as opposed to the others. Meanwhile, internalization is the most influential stage.

Key words: Pair programming, code quality, tacit knowledge, SECI Model, stage

INTRODUCTION

The facilitation ofteaching and learning practice has become a serious issue in order to improve student's programming skills particularly in Computer Science (CS) and Software Engineering (SE) courses (Omar *et al*., 2012). One of the possible techniques is Pair Programming (PP). PP is a collaborative programming manner of eXtreme Programming practices of agile software development family. The "driver" and the "navigator", who sit on one workstation with only one set of screen, mouse and keyboard and the technique they adapt in processing a task, distinguishes PP from other collaborative programming styles. This motivates practitioners in pedagogical context to rely on PP in reducing student's failure rate in programming subject.

Essentially, SE community admits PP as one of the unique approaches in overcomingprogramming issues in CS/SE courses (Sharifah-Lailee *et al*., 2009; Omar *et al*., 2009). In the late 1990's, PP has been embedded in the teaching technique in CS. Appropriate code quality is an indicator of expected programming skills. To come up with accurate code quality, it is necessary to polish student's personal knowledge. Generally, taking on PP in overcoming programming problem is extremely correlated with the main concern of this study which is to increase programming skills among students of higher learning institutions. It is represented in the form of a conceptual model. Hence, other than PP, constructing a model also requires a well-known model that deals with knowledge management and impact on tacit knowledge.

Literature review: Knowledge management has been defined by several scholars in various ways. It is comprehended as planning, controlling, organizing and inspiring individuals, systems as well as process development to improve knowledge asset and apply it effectively (King, 2009). To some extent, it is apparent as a procedure in establishing knowledge assets in order to undertake learning in the organization (Aggestam, 20015; Alipour *et al*., 2011).

As the foundation, the concept of knowledge as well as the conversion of tacit knowledge into explicit knowledge is documented as the basic component of knowledge management (Jabar *et al*., 2010). In order to convert tacit knowledge into explicit which involves experts and novices, four processes as outlined in SECI Model which are socialization (experts tacit knowledge to novices tacit knowledge), externalization (novices tacit knowledge to other individual's explicit knowledge), combination (individual's explicit knowledge-group explicit knowledge) and internalization (group explicit knowledge-organizational and individual's tacit knowledge) are normally involved.

Accordingly, knowledge sharing is a vital component in organizations. It is the answer to challenges and a popular complication problem in software development as maintained by agilest. That is the reason it is seen as a

Corresponding Author: Mazida Ahmad, School of Computer Computing, College of Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia

critical part of knowledge management and is an important mission in Agile. Its process involves two main components, the contributor and the receiver (Fengjie *et al*., 2004). The contributor begins by transmitting part of his/her knowledge to the receiver in the process. Then, the receiver will obtain the knowledge and adds their understanding and formulates it into their knowledge. This scenario is related with the PP practice, where the navigator plays the role of the contributor and the driver is the receiver.

In developing PP, knowledge sharing consists of social interaction, sharing and constructing knowledge between the partners (Kavitha and Ahmed, 2015). In this scenario, the SECI Model is applicable to promote sharing and constructing tacit knowledge between partners in generating codes with high quality (Omar *et al*., 2008). Code quality is an indicator for less number of imperfections in syntax and it measures the approval level of a program around users in terms of reliability, usability, maintainability and portability.

Furthermore, SECI Model recommends the facilitation and consideration of the association of interaction and transaction between tacit and explicit knowledge. Precisely, socialization mentions to a state in which tacit knowledge is created as the outcome of sharing mental thinking and practical experience during social interaction like informal session, debate and co-existence. With that, PP has been recommended in knowledge sharing for several courses.

MATERIALS AND METHODS

This study begins with investigating the relationship between code quality and PP in knowledge sharing. Particularly, code quality is the indicator for programs, that measures the number of successful test cases, effectiveness and academic performance (Salleh *et al*., 2011). Meanwhile in PP, the driver and the navigator uses a computer in their study by applying SECI Model. In such context, the independent and dependent variables are formulated as in Fig. 1.

Figure 1 shows that the code quality is evaluated with SECI Model. Particularly, the independent variables undergo the experimentation process. Then, the dependent variables are code quality and elements in the SECI Model. Generally, the role of PP and the proposed method as the research model are illustrated in Fig. 2.

In Fig. 2, the research model illustrates four common steps of software development life cycle. Specifically, PP practices are commonly applied in designing, coding and testing phases. In this study, the conceptual model examines the interaction between the driver and the navigator and its impact on tacit knowledge transferred from one student to another based on SECI Model. Then, it tracks the performance of students in terms of the quality of the end-program, based on marks given by the instructor. For better implementation of SECI processes, the participants were instructed with a set of guidelines, listed in Table 1.

Based on Table 1 and 4 components in SECI guidelines are formulated into 4 different hypotheses which are H_1 until H_4 as follow:

- H_1: the socialization process contributes to code quality with employing SECI process
- H_2: the externalization process contributes to code quality with employing SECI process

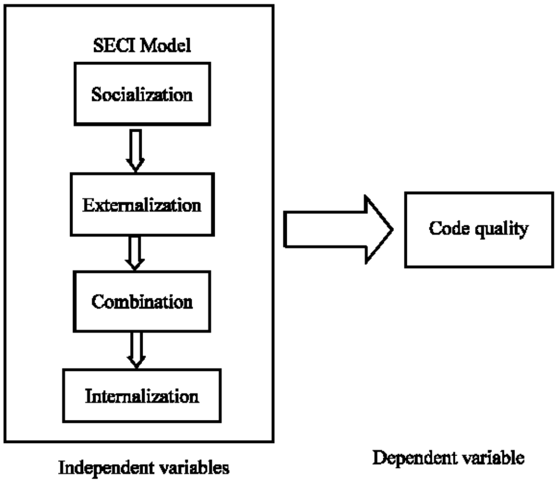

Fig. 1: Independent and dependent variables

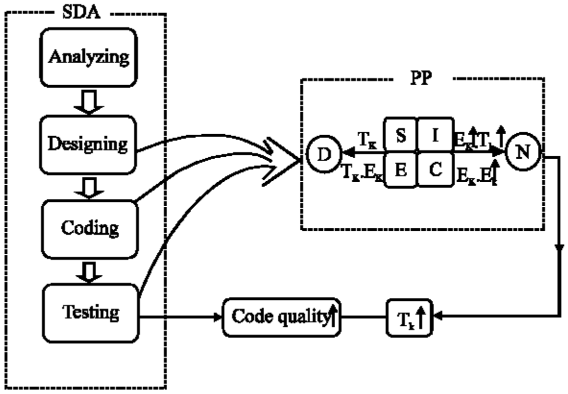

Fig. 2: Research model: SDA : Software Development; PP: Pair Programming activity; D: Driver; N: Navigator; TK: Tacit Knowledge; EK: Explicit Knowledge; S: Socialization; C: Combination; E: Externalization; I: Internalization

Table 1: SECI Guidelines

Hypothesis	SECI stage	Guidelines
H_1	Socialization	Each participant has to think for the solutions deeply in code quality
H_2	Externalization	The members of the pairs need to share by writing a draft code of the program and achieving quality code
H_3	Combination	The participants can refer to the internet, software book or any source to support their programming and code quality
H_4	Internalization	Once participants are satisfied with the output code, they can write and run it using the provided computer

- H_3: the combination process contributes to code quality with employing SECI process
- H_4: the internalization process contributes to code quality with employing SECI process

This study was carried out by using questionnaires which is adapted from (Omar *et al*., 2012). The instrument contains 31 items, asking participant's perception towards teaching materials, students, lecturers and the skills which refer to independent of learning, independent of thinking and independent of decision making. It was distributed to students who enrolled in PP lab experiment and gained experience in fundamentals of Java programming language.

The aim of the experiment was to measure the quality of the programs that the subjects produce (in pairs) with the full mark for the program is 40. Hence, the best program will be marked 40. Thus, the closer the mark to 40, the better the program is. Having collected the data, the correlation was determined between knowledge sharing and PP in code quality in which SECI Model has been adopted into PP Model.

RESULTS AND DISCUSSION

Based on the analyzed data using the International Business Management (IBM) Statistical Package for Social Sciences (SPSS) Version 20 and the Smart PLS 2.0 tools, the findings are discussed in the following subsections.

Respondent descriptive statistics: The statistical frequency distribution of variables in the questionnaire was classified and presented in Table 2 to reflect the originality of this study as.

Measurement and structural models: Structural Equation Modelling (SEM) is a methodological technique to ease the analytically complex model. Further it is a statistical technique for addressing a confirmatory approach of a structural theory that generates observations on multiple variables. Data are analysed using Partial Least Square SEM (PLS-SEM) due to its capability to analysis both normal and non-normal dataset. Within PLS, Composite Reliability (CR) is used to measure the internal consistency. For this study, the CR for each construct is shown in Table 3 which are >0.7. This indicates that the internal consistency is satisfactory.

Table 2: Demographic statistics

Factors/Variables	Frequency	Percentage
Gender		
Male	8	34.8
Female	15	65.2
Age		
18-20	6	26.1
21-23	14	60.9
24-26	3	13.0
Program		
Bsc information technology	18	78.3
Bsc multimedia	5	21.7
Course		
Database	9	39.1
Introduction to programming Java	11	47.8
Programming enhancement program	1	4.3
Expert system	1	4.3
Basic networking	1	4.3
Semester		
Semester 2	4	17.4
Semester 3	5	21.7
Semester 4	10	43.5
Semester 6	3	13.0
Semester 9	1	4.3

The construct reliability was evaluated independently. Indicator loadings must be significant at minimum 0.05 and the loading should be greater than 0.7 (Urbach and Ahlemann, 2010). This is because with the loading value at 0.7 a Latent Variable (LV) is considered to be able to explain at least 50% of its indicator's variance. On the other hand, Bootstrapping is a resampling method that can be used to examine the significance of the indicator loadings. In general, the decision of eliminating an indicator should be taken carefully when considering PLS characteristics of consistency. In case of low value of an indicator, it is logic to take the decision of eliminating that indicator and that elimination is linked with the significant increase of CR value. Therefore, the indicator reliability model ranges from 0.7682-0.923 as shown in Table 3.

Convergent validity indicates the extent to which individual items reflect a construct converging as compared with items that measure various constructs. With the aid of PLS, the value of Average Variance Extracted (AVE) is used to calculate the convergent validity. Convergent validity is considered sufficient when AVE value of a construct amount is not <0.5.

In regards to that, CR for the model shown in Table 3 reveals that the entire construct AVE values are greater than the threshold value (0.5). In the context of this research, the AVE ranges from 0.5182-0.6982. This also satisfies the AVE rule.

Discriminant validity is used to distinguish one measure from another of a construct measures. It examines

Table 3: Descriptive and reliability statistics

Construct/Item	Mean	SD	Loading	t-statistics	Composite reality	Ave. variance extracted
Socialization						
F1	4.13	0.920	0.5775	1.8629	0.9186	0.6982
SF2	4.30	0.703	0.9478	3.9258		
SF3	4.35	0.714	0.9201	3.3777		
SF4	4.30	0.703	0.8562	3.0366		
SF5	3.91	1.083	0.8243	3.7991		
Externalization						
E2	4.17	0.778	0.5145	1.207	0.7682	0.5182
E4	3.70	1.105	0.8784	2.297		
Combination						
C4	3.52	1.238	0.1697	0.3843	0.7805	0.5182
C5	3.43	1.199	0.9952	4.5707		
Internalization						
IIODM1	3.78	0.902	0.8559	3.2322	0.923	0.5245
IIODM2	4.09	0.733	0.6017	2.1824		
IIODM3	3.43	1.161	0.7347	2.6849		
IIODM5	4.00	0.739	0.6149	1.982		
IIOT3	4.04	0.767	0.7181	2.1625		
IIOT4	3.91	0.900	0.753	2.202		
IIOT5	3.96	0.767	0.7438	2.6714		
IIOL2	3.17	0.885	0.6689	2.2139		
IIOL3	4.04	1.054	0.6232	2.2286		
IIOL5	3.74	1.114	0.8432	3.1003		
IIOL7	3.65	0.878	0.7581	2.4053		
Code quality	-	3.57	2.233	1.0000	0.0000	1.0000

Table 4: Discriminant validity

Variables	Code quality	S	E	C	I
Code quality	1.0000				
S	-0.2548	0.8356			
E	-0.2896	0.4254	0.7199		
C	-0.4662	0.3401	0.5192	0.7139	
I	0.2918	0.6092	0.3571	0.4140	0.7242

whether the items intentionally measure another issue. Within PLS, cross loading and standard of Fornell-Larcke are two commonly used measures of discriminant validity. The first measurement analysis was conducted by examining the AVE in which results are represented in Table 4.

Validation of the structural model can assist this study to systematically estimate whether the data support the hypotheses characterized by the structural model. It is not proper to establish the analysis of the structural model unless the measurement model has been achieved successfully. Within PLS, a coefficient of determination (R^2) and path coefficients are used to evaluate the structural model.

The variance explanation of R^2 measures the relationship of LV to its total variance. Based on the benchmark (Chin, 1998) R^2 is considered weak if it is 0.19 and below. R^2 of 0.333 is accepted as the average while R^2 of 0.67 is considered as substantial. Figure 3 represents the results of structural model obtained in this study.It reveals that socialization, externalization, combination, and internalization are able to explain 72.4% of the variance on code quality.

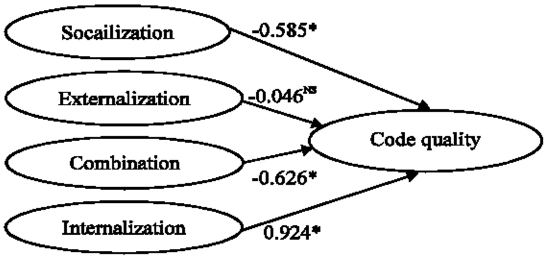

Fig. 3: Result of structural model; *p<0.05; NS: Not Significant

The model explains a significant amount of variance in the dependent variable (i.e., code quality) (R^2 = 0.724) which is strongly influenced by Internalization (β = 0.924, t = 2.417, $p<0.05$), followed by socialization (β = -0.585, t = 3.5097, $p<0.05$) and Combination (β = -0.626, t = 2.0671, $p<0.05$). Externalization (β = -0.046, t = 0.3025, NS) was found to be not significant in influencing code quality. Further, Table 5 shows the supported hypotheses for this study based on the results in Fig. 3.

The objective of this study concerns on investigating the relationships between each of the four processes of SECI Model and code quality. The literatures revealed that there is a relationship between the sharing of knowledge in the form of tacit to tacit between two people or groups towards achieving a code quality. This is confirmed by the results of the experiment with SECI Model. In the context of PP laboratory assignment, the results show that it would be easier for the participants to

Table 5: Hypotheses of the study

H	Code	Description	Results
H_1	S→CQ	The socialization process contributes to Code Quality with employing SECI process	Supported
H_2	E→CQ	The externalization process contributes to Code Quality with employing SECI process	Not supported
H_3	C→CQ	The combination process contributes to Code Quality with employing SECI process	Supported
H_4	I→CQ	The internalization process contributes to Code Quality with employing SECI process	Supported

achieve code quality due to their exposure to the knowledge of Java programming language. These explains that socialization is significantly related to code quality. The socialization process crucial in knowledge sharing due to positive communication and interaction between pairs as they chose partner that they are most comfortable with Lievre and Tang (2015).

The results reveal that, there is no significant relationship between the driver and the navigator in the effect of externalization on code quality. This is based on the obtained results from the analysis of the collected data. The obtained result affirms that achieving a project's completion (the transfer of knowledge from abstract to documented form) does not bring any improvement on the code quality (Ahmad *et al*., 2012).

The effect of combination on code quality is one of the knowledge management model which focuses on sharing or transferring of knowledge between the pair from explicit format to implicit format. The obtained results support the statement that the relationship between combination and code quality is significant. This means that it is mandatory to document the references that guide the code quality could be achieved through the combination form knowledge transfer.

Meanwhile, the effect of internalization on code quality in the SECI Model is described as a systematic explicit knowledge which can be converted into a richer consistent and more complicated tacit knowledge such as saved in human memory (memorization). This study confirms that there are significant relationships between Internalization and code quality of Java programming assignment. This implies that the exchange of knowledge from explicit form to tacit form while addressing Java programming language helps in achieving code quality.

CONCLUSION

As a conclusion, the significant findings among the four research hypotheses show that only one construct not supported, the externalization. In contrast, Internalization is found as the most influential factor among the SECI processes.

This study has contributed in providing a road map for the educators to achieve code quality using effective teaching methods through determining the impact factors for determining PP knowledge-based sharing for improving programming skills. Nevertheless, this study provides empirical evidences on the impact of each socialization, combination and internalization on code quality.

ACKNOWLEDGEMENTS

The 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

Aggestam, L., 2015. Learning organization or knowledge management-which came first the chicken or the egg?. Inf. Technol. Control, 35: 295-302.

Ahmad, M., M.M. Rejab, A.M. Syazwan, M. Omar, A.K. Bariah *et al*., 2012. Measuring tacit knowledge acquired during problem based learning teaching method in learning management system environment. AWER. Proc. Inf. Technol. Comput. Sci., 1: 775-781.

Alipour, F., K. Idris and R. Karimi, 2011. Knowledge creation and transfer: Role of learning organization. Int. J. Bus. Administration, 2: 61-61.

Chin, W.W., 1998. The Partial Least Squares Approach to Structural Equation Modeling. In: Modern Methods for Business Research, Gerorage, A.M. (Ed). Psychology Press, London, England, UK., ISBN:0-8058-3093-6, pp: 295-336.

Fengjie, A., Q. Fei and C. Xin, 2004. Knowledge sharing and web-based knowledge-sharing platform. Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business, Sept. 13-15, IEEE Computer Society Washington, DC, USA., pp: 278-281.

Jabar, M.A., F. Sidi and M.H. Selamat, 2010. Tacit knowledge codification. J. Comput. Sci., 6: 1161-1167.

King, W.R., 2009. Knowledge Management and Organizational Learning. In: Knowledge Management and Organizational Learning, William, R.K. (Ed.). Springer, Berlin, Germany, ISBN:978-1-4419-0007-4, pp: 3-13.

Kavitha, R.K. and M.I. Ahmed, 2015. Knowledge sharing through pair programming in learning environments: An empirical stud 2015. Knowledge sharing through pair programming in learning environments: An empirical study. Educ. Inf. Technol., 20: 319-333.

Lievre, P. and J. Tang, 2015. SECI and inter-organizational and intercultural knowledge transfer: A case-study of controversies around a project of co-operation between France and China in the health sector. J. Knowl. Manage., 19: 1069-1086.

Omar, M., M. Ahmad and A.K. Bariah, 2012. Knowledge internalization in pair programming practices. J. Inf. Commun. Technol., 11: 163-177.

Omar, M., R. Romli and A. Hussain, 2008. Automated tool to assess pair programming program quality. Proceedings of the 2008 International Conference Knowledge Management (KMICe2008), June 10-12, 2008, Universiti Utara Malaysia, Langkawi, Malaysia, ISBN:978-983-3827-73-2, pp: 516-521.

Salleh, N., E. Mendes and J. Grundy, 2011. Empirical studies of pair programming for CS-SE teaching in higher education: A systematic literature review. IEEE. Trans. Software Eng., 37: 509-525.

Sharifah-Lailee, S.A., O. Mazni, A.H. Mohd-Nasir, I. Che-Latifah and J. Kamaruzaman, 2009. Positive affects inducer on software quality. Comput. Inf. Syst., 2: 142-147.

Urbach, N. and F. Ahlemann, 2010. Structural equation modeling in information systems research using partial least squares. J. Inform. Technol. Theory Applic., 11: 5-40.