International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 23 (2015) pp 43669-43675s

The Role of Characteristics Behavior in Software Development Team to Improve Source Code Quality

Tien Fabrianti Kusumasari

Information System Department, FRI Telkom University Bandung, Indonesia
tienkusumasari@telkomuniversity.ac.id

Kridanto Surendro[1], Husni Sastramihardja[2], Iping Supriana[3]

Informatics, STEI Bandung Institute of Technology Bandung, Indonesia
endro@informatika.org[1], husni@informatika.org[2], iping@informatika.org[3]

Abstract
Human behavior in software construction is one of the primary factor of project success. This research would test the effect of human behavior on application development, especially in the construction phase. Project success in this research is measured with source code quality, which consists of reduction of source code complexity, % duplication, %technical debt, and the increase of SQALE rating. This research assess the effect of human behavior on source code quality of software industries in city of Bandung, Indonesia. Data processing and analysis are done using variance-based Structural Equation Modeling (SEM) methods specifically Partial Least Square technique. The contribution of human behavior properties variable (HB), Technological Usage variable (TU), experience of programmer (EXPC), and programmer ability of information processing (COQ) to source code quality (SCQ) are weak, which are 34.4%. Contribution of latent variable HB is 84% toward the decrease of SCQ value, meanwhile for technology usage (TU), information processing capability of programmer (COQ), and programmer experience (EXPC) each have contribution toward the increase of SCQ value respectively 53.9%, 43.8%, and 38.9%.

Keywords— human behavior, software development team, source code quality, collaboration, technical debt, code complexity, code duplication

Introduction

Collaboration is a process through which people who see different aspects of a problem can constructively explore their differences and search for solutions that go beyond their own limited vision of what is possible [1]. In [2] were stated a collaboration model known as Seven Layer Model of Collaboration (SLMC), consists of goals, products, activities, patterns, techniques, tools, and scripts.

Reference [3]classifies a collaboration model of software development according to management profile, communication, coordination, product integration, information flow and forms, typical lifetime of the collaboration, flexibility, risk management, intervention management, technology support, organization learning, organization trust and culture. The result of mapping collaboration process during software construction in [4] using axiomatic design (AD) principles were partnership, collaboration process, and software development methods. Reference [5] describes a collaboration framework for process classification and evaluation, which consists of six dimensions: organization distribution, temporal distribution, spatial distribution, process direction, process integrity, and discipline process.

The previous collaboration model research concluded that the collaboration model consist of process/method element, technology element, and human element. Reference [6] identifies human behavior properties during collaborative software development. The paper suggests that human behavior properties in software development team collaboration consists of technology tool usage, communication, interaction, potential capability, competency, physical and psychological condition, and human dynamics. The behavioral properties were improved using several case data into tool usage, symbol use, communication skill, potential capability, competency, physical state, physiological traits/state and experience [7]. The same research [7] suggests a collaboration model for software development construction phase (coding and testing) that were linked to product software quality. The model was changed to improve human behavior properties, make collaboration dimensions clearer and improve product quality measures in [8].

The objective of this research is to measure the contribution of human behavior on software development team collaboration during coding and testing phase. The human behavior model is based on Collaborative model in construction phase of software defined in [8] The measurement of human behavior properties were done using competency concept described in [9]. This collaboration model has been tested in software industries in Bandung, Indonesia which web based application project. The duration of IT (Information Technology) projects was three until six month and be built in PHP and Java programming language.

Literature Review

Human behavior was differentiated as internal behavior and external behavior in [10] Internal and external behaviors were components of human behavior [11]. Internal behavior is a behavior that exists within the person, while external behavior

is an action toward other person or objects in the environment. [10]-[11].
A competency is an underlying characteristic of an individual that is causally related to criterion-referenced effective and/or superior performance in a job or situation [12]. The characteristic is said to be a competency, if the characteristic predicts something valuable in the real world [12]. Thus human behavior in certain jobs can be measured using competency model [9].

A. Competence
Competency could be clustered based on the underlying intent, which analyzed social motives and superficial behaviors [12]. [12]classifies competency into six groups: achievement and action, helping and human service, impact and influence, managerial, cognitive, and personal effectiveness.
Using critical incident interview method, 38 essential competencies of software engineers were identified in [13]. The competencies were clustered into four categories: task accomplishment, personal attribute, situational skill, and interpersonal skill [13].

TABLE I. HUMAN BEHAVIOR MAPPING TO COMPETENCY

Human behavior property	Competency	Indicator	Measure
Interaction & Communication	• Teamwork, role in a group(TW1) • Teamwork, contribution in a group (TW3)	Spencer scale	Competency Level
Knowledge	• Software Construction Planning (PLN) • Managing Software Construction (MAN) • Detailed Design and Coding (DDC) • Debugging * Testing (DT) • Integrating and collaborating (IC)	SWECOM – IEEE 2014	Competency Level
Information Processing	• Analytical Thinking – Complexity of Analysis (AT1) • Analytical Thinking – Size of Problem (AT2/CT2) • Conceptual Thinking – Complexity and Originality of Concept (CT1) semua dimensi • Technical/Profesional Expertise – Acquisition of Expertise(EXP3) • Technical/Profesional Expertise – Distribution of Expertise (EXP4)	Spencer scale	Competency Level
Psychological Condition	• Achievement Orientation – Motivated Action (ACH1) • Achievement Orientation – Degree of Innovation(ACH3) • Information Seeking (INFO) • Concern for Order (CO) • Initiative – Self motivation and amount of discretionary effort (INT2)	Spencer scale	Competency Level

Reference [14] suggested that software development competence consist of business competence, technical/professional competence, and social competence. Business competency is required to understand relationship between business and organization. Technical/professional competence is a competency specific to a given work, assignment, occupation, activity, or profession. Social competence is a competency needed to be able to cooperate with others. Human behavior needed in collaborative team development in construction phase (coding & testing) could be measured using competency concept [9]. Mapping of human behavior using the competency concept are shown in Table I.**Error! Reference source not found.** [9]. Mapping of human behavior model based Human Behavior Representation Model (HBR Model), Spencer competency dictionary, and competency dictionary of SWECOM-IEEE 2014.

B. Teamwork Performance
According [11] human behavior is a purposive reaction of a human being to a meaningful situation. Human behavior have three interrelated components: cognitive, psycho-motor, and socio-affective [11]. Three major elements of human behavior dimension are person, environment, and time [15]. Human behavior in the environmental dimension consists of human behavior in physical environment and social environment. One kind of human behavior in social environment is human behavior inside group or team.
A team member must take part in individual assignments and teamwork process to achieve group purpose [16]. Individual task work is part of team member performance that have no dependence to other team members [17]. Teamwork is defined as group performance component that depends on each other, requires effective coordination between multiple individuals [17]. Individual performance consists of individual knowledge, trust, commitment, and flexibility [16]. Team process consist of coordination, communication, cohesion, decision making and conflict management [16].
Group collaboration processes that are done fully collocated (same geographical location) instead of fully distributed have positive impact to the software quality results as stated in Kusumasari, et al., 2013. The quality of team collaboration process in software team have positive correlation with team performance[18]. Collaboration inside the development team will give better performance than competition in the team [19]. Reference [20] creates a model that each development team member performance could be measured in source code quality, software development productivity, and competency distribution inside the team. Programming experience in previous projects has positive contribution to the source code quality [21].

Method
Human behavior mapping during development team collaboration, especially in construction phase were adopted from Collaborative model in construction phase of software described in [8]. Human behavior according to the adopted model consist of tool usage & programming languages,

communication and interaction, information processing, knowledge, motivation, and programming experience. Tool use capability and experience were measured using questionnaires, programming language capability were measured using programming quiz. Communication, interaction, motivation, and processing, knowledge, were measured by Spencer scale and Software Engineering Competency (SWECOM IEEE-2014) [22].

Programmer competency were measured using 360 degree appraisal. The 360 degree appraisal is a performance appraisal from multiple point of views: superiors, peers, and subordinates. Human behavior criteria mapping is done according construction-phase software collaboration behavior mapping done in reference [23]. Questionnaires, programming quiz, and 360 degree competency appraisal is done online by the respondents guided directly by this researcher.

The population for model testing is software house and IT division of companies located in Bandung, Indonesia. The amount of samples taken are 7 instances (of software company and IT division) with 17 project team constructing web-based application and 68 respondents. Data processing and analysis were done using Partial Least Squares (PLS-SEM), a variance-based Structural Equation Modeling method.

Result

Human behavior model during software development collaboration consist of tool usage, symbol mastering, communication capability, coordination capability, potential & competency, physical, psychological, and experience[6]. The human behavior properties could be related with team performance in the form of source code quality [8], as shown in. Based on these two factors, human behavior property could be evaluated according to the source code quality resulting from the efforts of the development team.

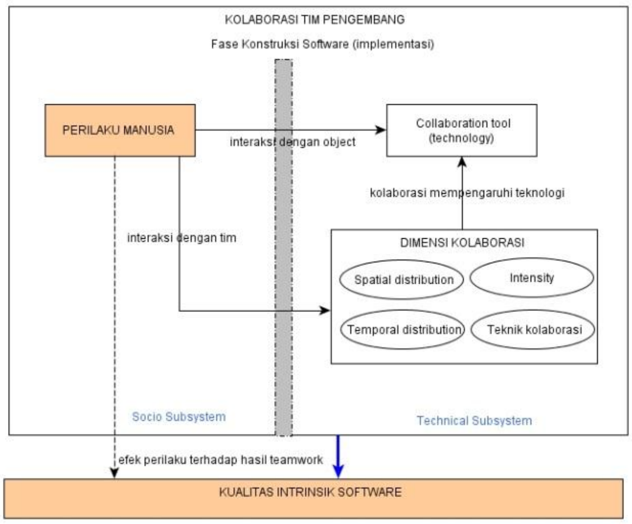

Figure 1. Collaborative Model in Construction Phase of Software

This paper proposes a human behavior model during team collaboration, especially coding and testing phase, as shown in.

In the structural model, human behavior in collaborative application development consists of: tool usage, communication and coordination capability, information processing, knowledge, physical, psychological state, and experience. The source code quality attributes that became indicators of team collaboration performance are: duplication, technical debt, cyclomatic complexity, and code size.

Measurements of human behavior properties (communication and coordination capability, information processing, knowledge, and psychological state) were done using 360 degree appraisals. Human behavior property mapping using competency concept were adopted from the research done in [9]. Physical properties, tool usage, and experience were measured using questionnaires and programming quiz.

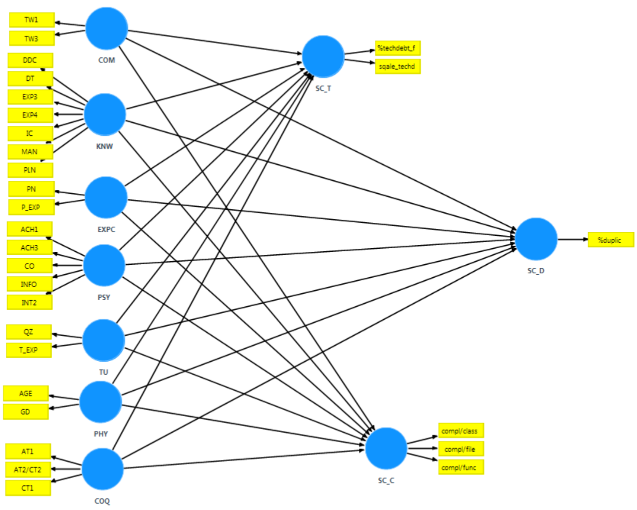

Figure 2. Structural Model of Human Behavior in Collaborative Software Development

Latent variable tool_usage (TU) was measured using time period of tool usage indicator (T_EXP) and programming quiz indicator (QZ). Latent variable communication and interaction (COM) was measured by role in group competency indicator (TW1) and contribution in group competency indicator (TW3). Latent variable knowledge (KNW) was measured using several indicators: professional expertise – expertise acquisition dimension (EXP3), professional expertise – expertise distribution (EXP4), software construction planning (PAN), managing software construction (MAN), detailed design and coding (DDC), debugging and testing (DT), and source code integration (IC). Latent variable expertise (EXPC) was measured using years of experience in software development (P_EXP) and amount of software projects that had been worked on (PN). Psychological state (PSY) was measured using motivation dimension of achievement competency (ACH1), innovation dimension of achievement competency (ACH3), concern for order competency (CO), information seeking competency (INFO), and initiative competency(INT2). Latent variable physical (PHY) was measured by gender and age indicators. Meanwhile latent variable information processing (COQ) was measured using complexity dimension of analytical thinking competency (AT1), size of problem dimension of analytical thinking and critical thinking

(AT2/CT2), and complexity and originality dimension of critical thinking competency (CT1).

Source code quality was measured using duplication (SC_D), cyclomatic complexity (SC-C), and technical debt (SC_T). SC_D was measured by %code duplication indicator, the higher SC_D means the worse the source code quality is. SC_C was measured using file complexity, per-class complexity, and per-function complexity. The higher the complexity score means more complexity in the source code, with upper limit of source code complexity about 20 per class. SC_T was measured using %technical debt indicator and SQALE rating. Higher the technical debt value corresponds to higher amount of working days required to fix source code problems. SQALE rating is a source code classification rating that are measured according to %technical debt, which we value as follows: A=1, B=2, C=3,D=4, and E=5. Smaller SQALE rating means better source code.

Structural Model was evaluated using PLS-SEM method based on data from software houses and company IT divisions situated in city of Bandung, Indonesia. The evaluation results for the outer model were described in. According to the model result, the outer model is not valid and reliable enough; therefore improvements were done by removing indicators that didn't qualify for the model. Validity and reliability test results of the refined model were shown in. According to the results, all indicators were valid and reliable. T-statistic value and probabilities are shown in.

TABLE II. VALIDITY & RELIABILITY OF REFLECTIVE INDICATOR

Latent Variable	Indicator	Loading	Composite Reliability	AVE
		>0.4	**>0.5**	**>0.5**
COM	TW1	0.986	0.512	0.487
	TW3	0.052		
KNW	DDC	139	0.266	0.161
	DT	-0.465		
	EXP3	0.443		
	EXP4	0.181		
	IC	0.291		
	MAN	0.119		
	PLN	0.751		
EXPC	PN	0.959	0.926	0.863
	P_EXP	0.898		
PSY	ACH1	0.722	0.803	0.467
	ACH3	0.795		
	CO	0.45		
	INFO	0.705		
	INT2	0.657		
TU	QZ	0.698	0.763	0.62
	T_EXP	0.867		
PHY	AGE	0.841	0.011	0.636
	GD	-0.751		
COQ	AT1	0.923	0.872	0.698
	AT2/CT2	0.691		
	CT1	0.874		
SC_D	%duplic	1	1	1
SC_T	sqale_techd	-0.974	0.154	0.688
	tech_debt	0.415		
	%techdebt_f	0.972		
SC_C	compl/class	0.156	0.782	0.608
	compl/file	0.947		
	compl/func	0.951		

TABLE III. VALIDITY AND RELIABILITY OF MEASUREMENT MODEL

Latent Variable	Indicator	Loading	Composite Reliability	AVE
		>0.4	**>0.5**	**>0.5**
COM	TW1	1	1	1
KNW	DT	-0.608	0.01	0.441
	PLN	0.716		
EXPC	PN	0.959	0.926	0.863
	P_EXP	0.898		
PSY	ACH1	0.72	0.803	0.456
	ACH3	0.799		
	CO	0.458		
	INFO	0.708		
	INT2	0.641		
TU	QZ	0.697	0.763	0.619
	T_EXP	0.868		
PHY	AGE	0.85	0.016	0.635
	GD	-0.741		
COQ	AT1	0.921	0.876	0.705
	AT2/CT2	0.71		
	CT1	0.873		
SC_D	%duplic	1	1	1
SC_T	sqale_techd	-0.981	0	0.966
	%techdebt_f	0.984		
SC_C	compl/file	0.953	0.948	0.901
	compl/func	0.945		

TABLE IV. SIGNIFICANT TESTING RESULT OF STRUCTURAL MODEL

Path			t-statistic	P	note
COM		SC_C	2.517	0.012	ok_5%
COM	->	SC_D	1.114	0.266	NS
COM	->	SC_T	0.358	0.721	NS
COQ	->	SC_C	0.293	0.77	NS
COQ	->	SC_D	1.114	0.266	NS
COQ	->	SC_T	1.143	0.254	NS
EXPC	->	SC_C	0.941	0.347	NS
EXPC	->	SC_D	0.409	0.682	NS
EXPC	->	SC_T	0.397	0.692	NS
KNW	->	SC_C	1.139	0.255	NS
KNW	->	SC_D	0.621	0.535	NS
KNW	->	SC_T	1.193	0.233	NS
PHY	->	SC_C	0.82	0.413	NS
PHY	->	SC_D	0.788	0.431	NS
PHY	->	SC_T	0.394	0.693	NS
PSY	->	SC_C	0.156	0.876	NS
PSY	->	SC_D	1.24	0.216	NS
PSY	->	SC_T	1.264	0.207	NS
TU	->	SC_C	0.303	0.762	NS
TU	->	SC_D	3.67	0	1%
TU		SC_T	2.305	0.022	ok_5%

Path diagram results of model refine-1 shows that paths in the model were not significant, except for path COM → SC_C, TU → SC_D, and TU → SC_T. Based on the results we could infer that most human behavior model do not directly influence source code quality. One of the way to improve the model so that the exogenous variables correlate with endogenous variables is by building a second order model [24]. The resulting improved model is shown in.

TABLE V. REFLECTIVE MEASUREMENT MODEL TEST RESULTS (REFINE-2)

Latent Variable	**Indicator**	**Convergent Validity**		**Reliability**		**Discriminant Validity**
		Loading	**AVE**	**Composite Reliability**	**Cronbach's alpha**	
		>0.4	**>0.5**	**>0.5**	**>0.6**	
KNW	DDC	0.850	0.545	0.855	0.786	Yes
	DT	0.673				
	EXP3					
	EXP4	0.683				
	IC	0.629				
	MAN	0.831				
	PLN					
EXPC	PN	0.938	0.868	0.893	0.848	yes
	P_EXP	0.926				
PSY	ACH1		0.534	0.819	0.707	yes
	ACH3	0.617				
	CO	0.704				
	INFO	0.736				
	INT2	0.848				
TU	QZ	0.788	0.624	0.769	0.398	yes
	T_EXP	0.792				
	COM_T					
PHY	AGE					
	GD					
COQ	AT1	0.820	0.735	0.893	0.820	yes
	AT2/CT2	0.864				
	CT1	0.886				
TW	TW1	0.728	0.609	0.756	0.361	Yes
	TW2	0.829				
HB			0.351	0.901	0.881	None
SQALE	sqale_techd	1.000	1.000	1.000	1.000	Yes

TABLE VI. FORMATIVE MEASUREMENT MODEL TEST RESULTS (REFINE-2)

Latent Variable	**Indicator**	**Weight**	**Loading**	**t-value**	**p-value**	**VIF**
			>0.5	**>1.96 or >1.65**		**0.1<VIF<10**
SCQ	compl	-0.362	0.709	0.928	0.353	3.779
	dupl	-0.505	0.537	1.568	0.117	3.348
	size	1.589	0.961	3.953	0.000	7.178

TABLE VII. SIGNIFICANT PATH COEFFICIENT MODEL TEST RESULTS (REFINE-2)

Correlation	**Path Coefficient**	**t statistic**	**p value**	f square
	>0.25	**>1.67**	**>0.1**	
COQ -> SCQ	-0.438	1.961	0.050	0.080
EXPC -> SCQ	-0.389	2.297	0.022	0.089
HB -> COQ	0.843	18.783	0.000	2.447
HB -> EXPC	0.768	10.331	0.000	1.435
HB -> KNW	0.853	19.947	0.000	2.672
HB -> PSY	0.763	14.881	0.000	1.393
HB -> SCQ	0.840	2.795	0.005	0.187
HB -> TU	0.483	3.934	0.000	0.304
HB -> TW	0.445	4.350	0.000	0.247
HB -> sqale	0.171	1.861	0.063	0.817
SCQ -> sqale	-0.655	7.135	0.000	0.817

TABLE VIII. MODEL R SQUARE DAN Q SQUARE (REFINE-2)

Latent Variable	**R Square**	**Q square**
COQ	0.710	0.502
EXPC	0.589	0.481
KNW	0.728	0.376
PSY	0.582	0.292
SCQ	0.344	0.152
TU	0.233	0.123
TW	0.198	0.091
Sqale	0.479	0.408

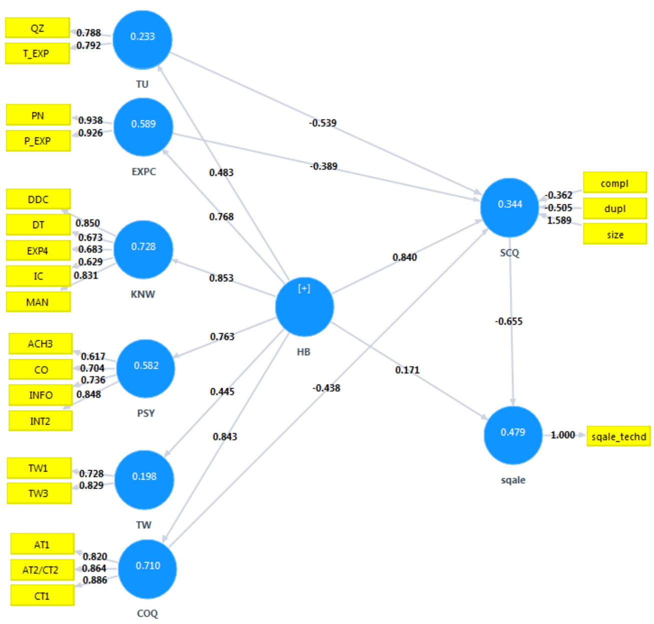

Figure 3. Structural Model of Human Behavior in Software Development Process (Refine-2)

The test results of all path in Refine-2 collaborative model (Table 7) shows that the paths are significant. Validity and reliability checks in Table 5 and Table 6 show that the Refine-2 collaborative model is valid and reliable.

Discussion

Measurement model evaluation results shows that all reflective indicators were valid and reliable in measuring latent variable HB and SQALE. But formative measurement model evaluation shows that complexity indicators were not significant. Complexity indicators were not removed from the model because basically source code complexity measurement were done using a mature common measuring tool that are stated as valid and reliable, taken from open source software. So the complexity indicators were assumed to be proven in validity and reliability.

Human behavior (HB) latent variable is second order variable HB was composed by first order latent variable, which are COQ (information processing), EXPC (programmer's experience), KNW (knowledge), PSY (psychological state), TU (technological usage), and TW (teamwork). The contribution of HB, TU, EXPC, and COQ to source code quality (SCQ) are weak, which are 34.4% (R^2 = 0.344). The rest factor of SCQ is influenced by other factors outside the scope of this study. Improvement of source code quality is measured by decrease of code complexity and duplication. If code size (function/method) per class increased to more than

14, the effect is reduction in quality [25].In the simulation data, the increase of SCQ are followed by the increase of code size (size). This is caused by the sample data mostly from the projects with high function-to-class ratio, causing incomplete value distribution statistics. Contribution of latent variable HB is 84% toward the decrease of SCQ value, meanwhile for technology usage (TU), information processing capability of programmer (COQ), and programmer experience (EXPC) each have contribution toward the increase of SCQ value respectively 53.9%, 43.8%, and 38.9%.

Largest human behavior property contributor to source code quality is technology usage (TU), which contribution is 53.9%. Technology usage consists of programming language mastery and expertise using collaboration tool in software construction. Expertise in collaborative technology for coding and testing such as SCM (software configuration management), bug tracking, automatic testing and communication technology will help programmers to build source code that are easier to maintain. Collaboration through SCM would provide programmers access to all source code, in which enables them to give suggestion and solutions to increase source code quality. The research done by [26] shows that the team of programmers trained in automated testing implementation would produce source code with low complexity. Meanwhile, agile software development concept states that tool usage is mandatory in application development [27].

Information processing (cognitive) capability of programmers also contributes to source code quality (43.8%). This indicator is measuring complex thinking capability, problem decomposition capability, originality, and innovation. In order to solve complex problems during coding and testing, high analysis capabilities, multiple viewpoints for a given problem, and problem simplification capability are required. Presence of a programmer with high cognitive capability in a development team would increase source code quality. The result of this research was inline with the requirement of development team when using extreme programming technic, that is the existence of programmer with advanced capability [27].

Programmer experience in application development (coding & testing) would positively influence source code complexity and duplication reduction, with 38.9% contribution. The most important success factor for software development team is that the team members consists of 2-8 person, working in the same room, existence of expert in the domain being developed, short delivery cycles, and experienced programmers [28]. Experienced programmers will have development completion speed between 2-10x of normal programmers [28].

Based on the survey results and discussions above, this research can contribute to the election of members of the development team. Programmers with the ability to mastery of programming language and programming environments(with high programming quiz scores), experience in the use of software tools (SCM, bug tracking tool, automatic testing unit, communication tool), a high ability to process information, and how much experience of software application projects will make a major contribution to the quality of the source code. Model (linear equations) produced in this study can be embedded in an application that is used to conduct an assessment of the ability of the programmer. In addition this model can also be used to search for methods of collaboration are appropriate for a particular programmer in order to produce a good source code (value SQALE=Aor1).

Hypotheses for further research is that not all abilities is the key to improve the quality of the source code must be owned by a programmer as well, but these abilities must exist on programmers who are members of a team of developers. Therefore, this study can be developed further, for example by testing the suitability of the existing members of the team with the method and process of collaboration so as to produce a good quality source code (the value SQALE = 1 / A). How can the source code of the application made by industry keep software quality, by optimizing the existing programmer (owned by the software industry).In this case a lot of empirical data needed to model the characteristics of the team with various combinations of programmer.

Conclusion

Human behavior properties in collaborative coding & testing phases contribute to the source code quality of the resulting team product. The order of human behavior property contributions from large to small are technology use, programmer's information processing and cognitive thinking capability, and programmer's experience in application development projects. The three variables directly influence source code quality in reduction of complexity and duplication. On the other hand, combined of human behavior properties contributes 84.7% towards source code quality reduction.

Acknowledgements

The authors thank to the Director General of Higher Education, Ministry of National Education of the Republic of Indonesia for a research grant which has funded this research. The author also would like to thank the Institute of Technology Bandung, which has helped facilitating laboratory space and access to literature. Special thanks to Telkom University that permit for author to conduct the doctoral study.

References

[1] Callahan, S., Schenk, M., & White, N. (2008). Building a Collaborative Workspace. Anecdot White Paper.

[2] Briggs, R. O., Kolfschoten, G., Vreede, G.-J. d., Albrecht, C., Dean, D. R., & Lukosch, S. (2009). A Seven-Layer Model of Collaboration: Separation of Concerns for Designers of Collaboration Systems. Thirtieth International Conference on Information Systems (pp. 1-14). Phoenix: Association for Information Systems.

[3] Mohtashami, M., Kirova, V. D., Marlowe, T. J., & Deek, F. P. (2009). A comparison of three modes of collaboration for software development. Proceedings of the Fifteenth Americas Conference on Information Systems. California.

[4] Arsenyan, & Büyüközkan. (2009). Modelling Collaborative Software Development Using

Axiomatic Design Principles. IAENG International Journal of Computer Science, 36(3)

[5] Hildenbrand, T., Rothlauf, R., Geisser, M., Heinzl, A., & Kude, T. (2008). Approach to Collaborative Software Development. International Conference on Complex, Intelligent and Software Intensive Systems, (pp. 523-528).

[6] Kusumasari, T., Sastramiharja, H., & Surendro, K. S. (2013). Human Behavior Factor of Collaborative Software Development (Faktor Perilaku Manusia dalam Kolaborasi Pengembangan Perangkat Lunak). Mataram: KNSI 2014.

[7] Kusumasari, T., Sastramiharja, H., & Surendro, K. S. (2013). Human behavior conceptual model in collaborative software development product quality. Bali: Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on.

[8] Kusumasari, T. F., Sastramihardja, H., Surendro, K., & Supriana, I. (2014). Collaborative Model in Construction Phase of Software. ICITSI. Bandung-Bali: IEEE.

[9] Kusumasari, T. F., Sastramihardja, H. S., & Supriana, I. (2015). Instruments Measurement Design of Human Behavior in Collaborative Software Construction. The 2015 International Conference on Advanced Informatics: Concepts, Theory and Application (ICAICTA2015). Chonburi, Thailand: in press.

[10] Popp, A. (2009, September 17). An understanding of influence on human behavior. Retrieved January 1, 2013, from MPRA Paper No. 17367: http://mpra.ub.uni-muenchen.de/17367/

[11] Dompke, U. (2001). Human Behavior Representation : Definition, Presentasi pada RTO SAS Lecture Series on Simulation of and for Military Decision Making, II-1 –II-12. USA.

[12] Spencer, L. M., & Spencer, S. M. (1993). Competence Work : Models for Superior Performance. United State of America: John Wiley & Sons, Inc.

[13] Turley, R. T., & Bieman, J. M. (1994). Identifying essential competencies of software engineers. New York: CSC '94 Proceedings of the 22nd annual ACM computer science conference.

[14] Duka, D., & Hribar, L. (2011). Implementation of software competence development process. Opatija : IEEE.

[15] Hutchison, E. (2007). Dimension of Human Behavior : Person and Environment, 4th ed, chapter 1, page 3-35. SAGE Publications.

[16] Lo, L. (2011). Teamwork and Communication in Healthcare a Literature Review. Ottawa: Canadian Patient Safety Institute.

[17] Salas, E., Cooke, N. J., & Rosen, M. A. (2008). On Teams, Teamwork, and Team Performance: Discoveries and Developments. Human Factors: The Journal of the Human Factors and Ergonomics Society, 50(3), 540-547.

[18] Hoegl, M., & Gemuenden, H. G. (2010). Teamwork Quality and the Success of Innovative Projects: A Theoretical Concept and Empirical Evidence. Organization Science, Vol.12, No.12, 435-449.

[19] Avital, M., & Singh, B. (2007). The Impact of Collaboration and Competition on Project Performance. ICIS 2007 Proceedings, Paper 89 (pp. 1-19). AIS Electronoc Library.

[20] Schwind, M., Schenk, A., & Schneider, M. (2010). A Tool for the Analysis of Social Networks in Collaborative Software Development. Proceedings of 43rd Hawaii International Conference on System Science. Hawaii.

[21] Kusumasari, T. F., Surendro, K., Sastramihardja, H., & Supriana, I. (2013). Faktor manusia dalam kolaborasi pengembangan perangkat lunak. KNSI. Lombok.

[22] IEEE. (2014, March). Software Engineering Competency. Retrieved from IEEE Computer Society: https://computer.centraldesktop.com/swecomreview/

[23] Kusumasari, T. F., Sastramihardja, H. S., & Supriana, I. (2015). Instruments Measurement Design of Human Behavior in Collaborative Software Construction. Submited. -: Submited.

[24] HairJr, J. F., Hult, G. M., Ringle, C. M., & Sarstedt, M. (2014). A primer on partial least squares structural equation modeling (PLS-SEM). Wasington DC: SAFE Publication.

[25] McCabe Software. (n.d.). McCabe Whitepaper. Retrieved April 1, 2014, from McCabe Software: http://www.mccabe.com/pdf/McCabeCodeQualityMetrics-OutsourcedDev.pdf

[26] Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development method (review and analysis). Retrieved 02 2011, from Portal of Agile Software Development methodologies at VTT Electronics: http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf

[27] Bannerman, S., & Martin, A. (2011). A multiple comparative study of test-with development produt qualityct changes and their effects on team speed and produc. Empirical Software Engineering, 16(2), 177-210.

[28] Abrahamsson, P. (2007). Innovation Report modeling. Retrieved October 16, 2010, from ITEA homepage on: www.itea2.org/project/result/download/result/5583