

A BRIEF METHOD TO ASSESS THE ASSOCIATION OF SOCIO AND TECHNICAL DEPENDENCIES ON SOFTWARE QUALITY

A. J. Suali¹, S. S. M. Fauzi^{1,*}, M. H. N. M. Nasir² and W. A. W. M. Sobri¹

¹Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Arau, Perlis, Malaysia

²Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

Published online: 17 October 2017

ABSTRACT

Managing coordination among developers is very challenging when dependent task occurred since the quality of the software during development need to be maintained. Socio-Technical Congruence (STC) introduced to measure the coordination and help to identify the coordination breakdown from the root of development. However, since STC is a new area in software engineering, there are limited researches on this study. That provides opportunities to expand STC study by investigate the relationship of socio and technical dependencies on software quality in software engineering project. This paper outline related research, literature review and the method used to conduct the study.

Keywords: socio-technical congruence; actual coordination; coordination requirements; software engineering projects; software quality.

Author Correspondence, e-mail: shukorsanim@perlis.uitm.edu.my

doi: <http://dx.doi.org/10.4314/jfas.v9i5s.9>

1. INTRODUCTION

To develop software systems with complexity and challenges required high tolerance and coordination among developers. The more complex project is developed, the more frequent the coordination happen among developers because of the dependencies task activities. Moreover, difficulties might occur when organize team development distributed because of different cultures and backgrounds, but they need to coordinate when their task dependence takes place during development process [1]. Therefore, a measurement is needed to conceptualize coordination to get familiar with environment in software development.

Socio-Technical Congruence (STC) was introduced by [2] to conceptualize and measure coordination among developer by examine the alignment between actual coordination and coordination requirement. Actual coordination derived from two or more developers need to coordinate according to the task dependencies of the project [3]. Meanwhile, coordination requirement can be described as dependencies among task that require coordination among developer to complete the task. STC is introduced to identify the mismatch between dependencies and coordination activities [4]. STC using Conway's law to construct the model which any activities that carried out must be based on the architecture design. Conway's law explain the communication happen during development according to the structure of the design architecture demand [5]. The coordination among developers should be based on the design project to ensure the achievement of the goals.

Since STC is a new area, there are limited researches on STC in software engineering projects. This paper intends to discuss a brief method to assess the association of STC in software engineering projects. This paper is structured as follows. Firstly, this paper discuss on the literature of STC. Next section describes the methodology of this research. Finally, we concluded this paper.

2. LITERATURE REVIEW

One of the critical issues in software engineering projects is coordination among developers, a factor that is recognized to be important in software development project [6][7]. Coordination can be defined as *“a management in dependencies and focus on the relationship between coordination structure and project performance during development of project in software*

engineering” [8-9]. Furthermore, coordination can be stated as extra information engagement when several developers pursue the same goals as the developer in-charge does not perform the task given [10]. Frequent coordination among developers occurs because of the dependence of the task activities [8, 11-12]. To elaborate, coordination is emphasized when interdependence of tasks between developers takes place during development [13]. In software engineering, a complex project involves many lines of code, plenty of documentation, large quantities of design information, specifications, test cases and testing output as well as user manuals and maintenance manuals. All these need to be organized and assigned to eligible developers, usually called technical structure, in which developers handle their own work and also coordinate with other developers to exchange vital information in completing tasks [14-15].

Coordination among developers can be done in several ways. Coordination could happen in formal condition such as group meeting to discuss about the source codes, test cases and documentations [23]. Meanwhile, informal coordination happen outside the company where they can discuss freely with other developers about the project issues during development phase [24]. Coordination among developers not only happens in meeting room or informal situations but also via email, instant messenger (IM) and web-based application in order to monitor project progress [25]. As it is known, development of project in software engineering involves people from different backgrounds, skills, knowledge and expertise and requires them to mingle, work together as well as coordinate with each other [26-27]. To elaborate further as distributed developers work in a team during the project, they can combine their expertise and skills to reduce the cost of transferring information and decision-making time [28]. Teamwork helps by increasing the skills and collaboration level in development of each individual.

Collaboration between distributed teams requires a social network as a medium to follow-up updates of the project from other developers [29]. Hence, distributed teams need to follow-up on progress made even if it is not related to their task. This is to ensure that any difficulty related to tasks or task dependence can be solved based on necessary requirements. Worsen with the boundary such as distance, time zone, languages and cultural may disturb the coordination among developers and effects the communications, delayed and high risk of

development [30-33]. In this case, communication plays an important role in exchanging information to ensure the coordination in software development project is on the right track especially in area of global collaboration. This is because decisions regarding the project like design, documentation, report and changes can bring implications to development efforts by the other team [20].

In order to ensure the coordination among developers is sufficient during development, STC was introduced by [2] to measure the coordination by measuring the alignment between actual coordination and coordination requirement. Actual coordination is the relationship between two developers who communicate with one another on matters related to work, hence resulting in coordination [17]. Meanwhile, coordination requirement is about tasks that are dependent on each other and developers should work together to achieve coordination. The purpose of STC is to coordinate dependent tasks properly. STC uses the “fit” measure between structure of the software system and structure of organization that is responsible for the development [2]. Empirical study suggests that when applying STC in development, better management and organizational environment can be created in order to identify any conflict area in development thus promoting communication flow in affected areas [34]. In other findings, STC suggests that low congruence may cause developers to be unaware of important project dependencies, hence resulting in missed opportunities to collaborate with likeminded developers [35]. Empirical evidence affirms that STC can help in improving team productivity and coordination between developers [36]. The previous study added that when STC is used in development especially in decision-making, it assists in solving current issues plaguing a development project [35]. Furthermore, STC is not only used to help understand architecture design and organizational management but also assists in decision-making when changes occur in architectural design or in organization. This is done to minimize the negative impact of the changes and to ensure that congruence is intact [14].

Strengthening of STC through collaboration among developers in coordinating during development also helps the organization to explore the most effective way to achieve its objectives [37]. In addition, previous study conducted used STC to help developers create various strategies based on the budget provided by the organization. Developers then needed to pick the best strategy relevant to the project in hopes of improving team productivity and

coordination between developers [38].

Lack of congruence due to lack of awareness, unclear responsibilities, delayed problem-solving and mismatched changes may lead to delays in problem solving, contrasting modification and time taken to complete the given tasks [39]. Therefore, good congruence through communication among developers based on architecture design certainly helps explain the faulty conceptualization in some parts of the system [40]. However, little research has been done on how STC influence software quality when coordination requirement and actual coordination are congruent.

3. OVERVIEW OF THE METHOD

3.1. Setting and Datasets

This research intends to use open source project. Coordination needs is more significant in open source project since the developer team is distribute [40]. To conduct this research, Yetus project with 20 developers was chosen. Yetus project is selected as it provide libraries and tools that enable contribution and release processes for software projects. Table 1 summary for open source project used in this study:

Table 1.Project summaries

Project Name	No of Tasks	Total No of Developers Involved	Versions
Yetus	442	20	0.1.0
			0.2.0
			0.3.0
			0.2.1
			0.4.0

3.2. Descriptions of Measures

Fig. 1 illustrates the model applied in this research. This model involved two variables which are dependent and independent variables.

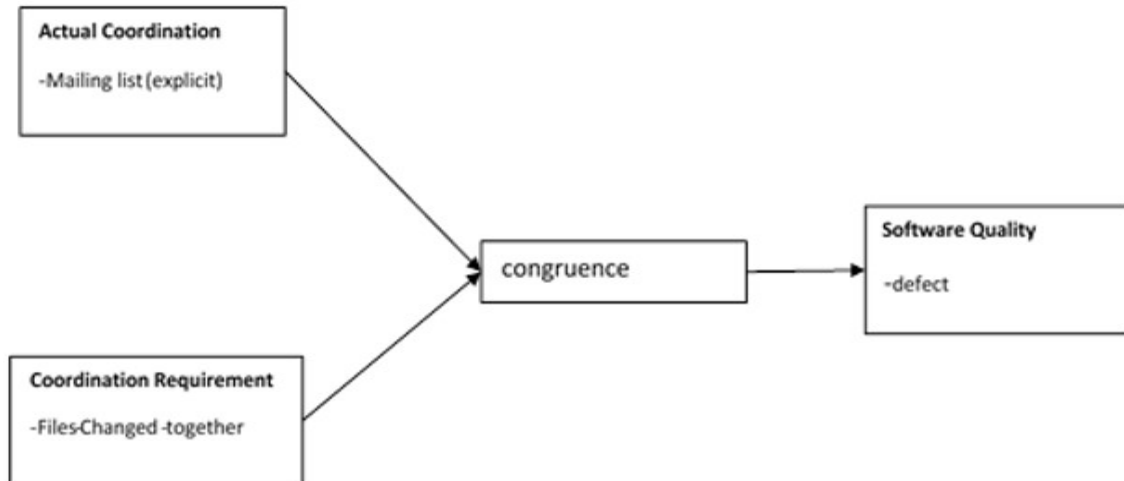


Fig.1.Research model

Software quality is the dependent variable in this research. For the purpose of the research, defect will be used as a measure for software quality [41]. Defect is found during tester which executes the test cases. The test result is referred as defect. The defect also can be called bug, issue and problem. It is expected that low defect is achieved when the congruence is high. It means that the defect will decrease in every version to make the comparison whether the quality impacted by defect in project. Defect is calculated by using the following equation [42].

$$Defects = \frac{Total\ defect}{Size\ of\ Project}(1)$$

where the total of defect calculated by a number of defects found in every version divided the size of the project. This is use for comparing the software versioning in order to identify whether the latest versioning of software is better than the previous version.

Meanwhile, first element for independent variable is coordination requirement which derived from a relationship that indicates from two or more developers should be coordinating based on their task performance [43]. Fig. 2 [44] illustrates the interdependencies task and the developer assigned the tasks.

Fig. 2 (a) explain the individual assigned the task where each of tasks are related with each other. Task 3 relates to the task 2. Task 2 and 1 are related with each other. This means that in Fig. 2 (b), it explains how developers will coordinate with each other based on the task given. If the tasks are interdependent, the developers need to communicate with each other. Thus, this defines how coordination requirement will lead to communication among developers.

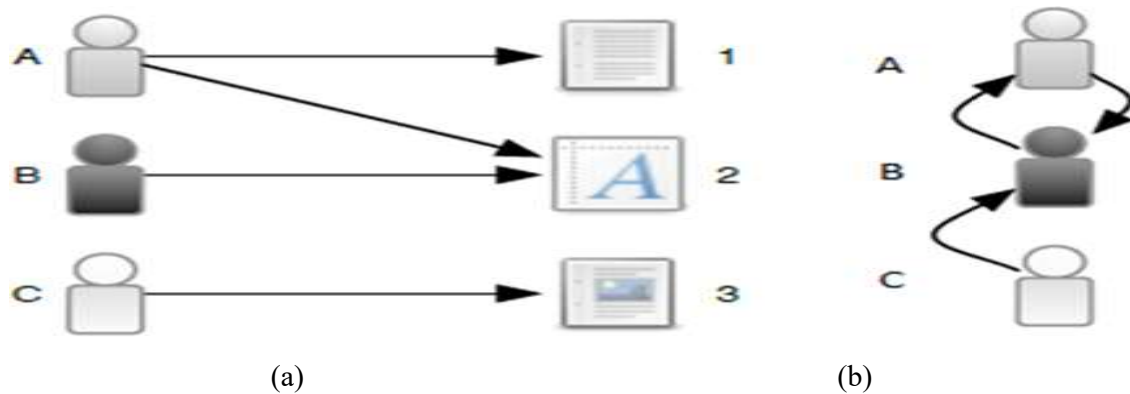


Fig.2. Examples of interdependence task and developers in development

Coordination Requirement is achieved by multiplying *task assignment* (T_A) and *task dependencies* (T_D) and *transpose of task assignment* (T_A^T). Task assignment derived from person (i) assigned to the task (j). Meanwhile, Task dependencies represent to the task (i) and task (j) are interdependent. The result for Coordination Requirement (C_R) will be in table matrix person (i) by person (j) where i works on the task that share work dependencies with the tasks worked on j .

The following equation will be used to calculate coordination requirement [2]:

$$C_R = T_A * T_D * T_A^T \quad (2)$$

Information for coordination requirement will be extracted using an Excel format. From the Excel format, table matrices that provide information on the tasks assigned for each developer and files changed in each task will be developed. Each table will be developed using R script. Actual coordination is the second element for independent variable which is to measure the communication activities among developers. Developers that communicate together stated that they coordinate related to their work. Actual coordination will be calculated based on communication constructed from a mailing list that occurs as “*sender-receiver*”.

Actual coordination is calculated by developing table matrices that indicate the communication which occurred between person i and person j . The table matrix shows the interactions between developers represent actual coordination (C_A). Information related to developer interactions will be extracted from the project’s mailing list. Mailing list is a place for developers to communicate to discuss the current topic or issues arise in their current project.

Congruence is measured by comparing the metric between coordination requirement and actual coordination. Congruence is the alignment between coordination requirement and actual coordination. The calculation is based on comparing coordination requirement matrix with actual coordination matrix. The value for congruence is from 0 to 1 [44]. This suggests that, when the match between coordination requirement matrix and actual coordination matrix is not enough, the congruence is less than 1. The following is the equation that will be used to compute congruence [4]:

$$\begin{aligned} \text{Diff}(C_R, C_A) &= \text{card} \{ \text{diff}_{ij} \mid C_{R_{ij}} > 0 \ \& \ C_{A_{ij}} > 0 \} \\ |C_R| &= \text{card} \{ C_{R_{ij}} > 0 \}, \\ \text{Congruence} &= \text{Diff}(C_R, C_A) / |C_R| \end{aligned} \quad (3)$$

These equations describe the calculation of the congruence. Since coordination requirement (CR) matrix constructed from tasks which are dependent, so this matrix can be compared with actual coordination (CA) matrix that represents the coordination activities of interaction among developers. Then, congruence is defined as the proportion of coordination activities that actually occurred (refer to CA) that need to compare the total of the number of coordination activities that should be taken place (refer to CR). For example, if the CR matrix shows that 8 persons should coordinate and 4 interactions among developers identified in coordination activities in CA matrix, the proportion of congruence is 0.5. Since the proportion of congruence is 0 to 1, the total of congruence between CR matrix and CA matrix is less. So, the coordination among developers is not sufficient. This measure of Socio-Technical Congruence provide a new way of perspective about coordination by providing a fine-grain level of analysis of different types of technical dependencies and explore how the coordination needs are impacted by the technical dependencies [4].

Next is to investigate the relationship between STC and software quality, which will be calculated using linear regression to identify the association in software engineering project. Probability-value (*p-value*) reflects to the probability which the statistical expectation for a given test is true. Acceptance significant levels are 0.1, 0.05 and 0.01 which indicates that the smaller the value, the higher the confidence level of the relationship exists. Following is the equation to calculate the relationship between STC and software quality using linear

regression model:

$$y = \beta_0 + \beta_1 x \quad (4)$$

where y is defect, x is congruence, β_0 is the intercept and β_1 is the coefficient which serve as the slope of the straight line. To perform this research, R script will be used to extract data, cleaning and statistical regression. Some unused data will be removed using R script. Before applying the script in this research, R script needs to be examined and tested before applied in the real project. Fig. 1 illustrates the extraction method used in this study.

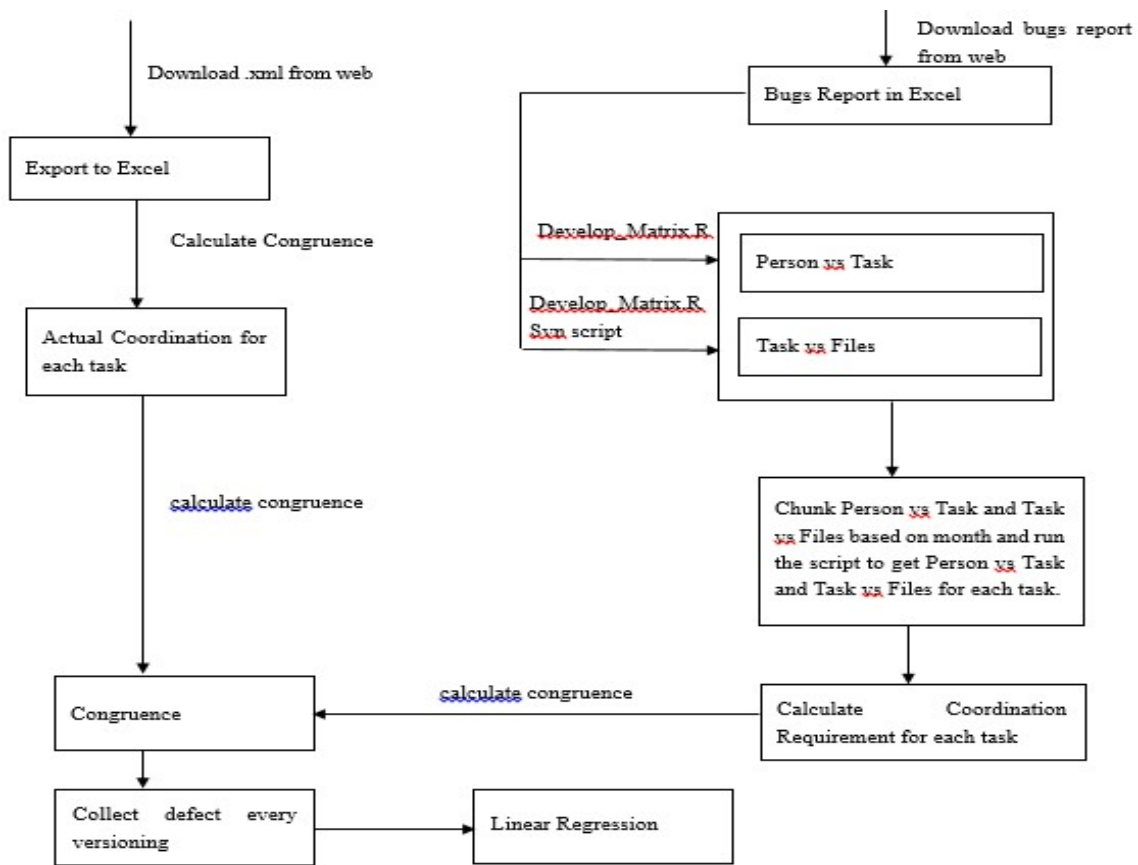


Fig.1.Data extraction and analysis method

4. CONCLUSION

This paper discusses a method to assess the association of STC in software engineering projects especially the impact on software quality. We expect this study would give significant impact to the software quality when we implement STC on the development where we

assume high congruence in development would decrease the number of the defects and increase software quality. Currently, we have completed gathering the required data which used to perform this study and run the R script to extract the data and clean the irrelevant data from modification request and mailing list. Next step is to analyze the relationship between STC and Software quality using linear regression.

5. ACKNOWLEDGEMENTS

This research work was supported by Ministry of Education (MOE), Malaysia and Universiti Teknologi MARA, Malaysia under Research Acculturation Grant Scheme (RAGS), Project code: “600-RMI/RAGS 5/3 (4/2014)”.

6. REFERENCES

- [1] Hirama K. Social requirements elicitation for socio-technical systems development. *IEEE Latin America Transactions*, 2013, 11(2):870-877
- [2] Cataldo M, Wagstrom P A, Herbsleb J D, Carley K M. Identification of coordination requirements: implications for the Design of collaboration and awareness tools. In 20th Anniversary Conference on Computer Supported Cooperative Work, 2006, pp. 353-362
- [3] Kwan I, Schröter A, Damian D. Does socio-technical congruence have an effect on software build success? A study of coordination in a software project. *IEEE Transactions on Software Engineering*, 2011, 37(3):307-324
- [4] Cataldo M, Herbsleb J D, Carley K M. Socio-technical congruence: A framework for assessing the impact of technical and work dependencies on software development productivity. In 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 2008, pp. 2-11
- [5] Conway M E. How do committees invent. *Datamation*, 1968, 14(4):28-31
- [6] Bailey S E, Godbole S S, Knutson C D, Krein J L. A decade of Conway’s law: A literature review from 2003-2012. In 3rd International Workshop on Replication in Empirical Software Engineering Research, 2013, pp. 1-14
- [7] Pirkkalainen H, Pawlowski J M. Global social knowledge management-Understanding barriers for global workers utilizing social software. *Computers in Human Behavior*, 2014,

30:637-647

- [8] Yuan M, Zhang X, Chen Z, Vogel D R, Chu X. Antecedents of coordination effectiveness of software developer dyads from interacting teams: An empirical investigation. *IEEE Transactions on Engineering Management*, 2009, 56(3):494-507
- [9] Scheerer A, Hildenbrand T, Kude T. Coordination in large-scale agile software development: A multiteam systems perspective. In *47th Hawaii International Conference on System Sciences*, 2014, pp. 4780-4788
- [10] Ciancarini P. Coordination models and languages as software integrators. *ACM Computing Surveys*, 1996, 28(2):300-302
- [11] Blincoe K. Timely detection of coordination requirements to support collaboration among software developers. In *34th International Conference on Software Engineering*, 2012, pp. 1601-1603
- [12] Dittrich Y, Randall W, Singer J. Software engineering as cooperative work. *Computer Supported Cooperative Work*, 2009, 18(5-6):393-399
- [13] Lee G, Espinosa J A, Delone W H. Task environment complexity, global team dispersion, process capabilities, and coordination in software development. *IEEE Transactions on Software Engineering*, 2013, 39(12):1753-1771
- [14] Betz S, Smite D, Fricker S, Moss A, Afzal W, Svahnberg M, Wohlin C, Borstler J, Gorschek T. An evolutionary perspective on socio-technical congruence: The rubber band effect. In *3rd International Workshop on Replication in Empirical Software Engineering Research*, 2013, pp. 15-24
- [15] Osterweil L J. System processes are software too. In *9th IEEE International Conference on Software Engineering*, 1987, pp. 2-13
- [16] Bourazeri A, Pitt J. collective awareness for collective action in socio-technical systems. In *8th IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops*, 2014, pp. 90-95
- [17] Kwan I, Damian D. Extending socio-technical congruence with awareness relationships. In *4th International Workshop on Social Software Engineering*, 2011, pp. 1-8
- [18] Smite D, Dingsøyr T. Fostering cross-site coordination through awareness: An investigation of state-of-the-practice through a focus group study. In *38th EUROMICRO*

-
- Conference on Software Engineering and Advanced Applications, 2012, pp. 337-344
- [19] Blincoe K, Valetto G, Damian D. Facilitating coordination between software developers: A study and techniques for timely and efficient recommendations. *IEEE Transactions on Software Engineering*, 2015, 41(10):969-985
- [20] Bass M, Herbsleb J D. On coordination mechanisms in global software development. In *2nd IEEE International Conference on Global Software Engineering*, 2007, pp. 71-80
- [21] Kasi B K. Minimizing software conflicts through proactive detection of conflicts and task scheduling. In *22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2014, pp. 807-810
- [22] Marczak S, Damian D. How interaction between roles shapes the communication structure in requirements-driven collaboration. In *19th IEEE International Requirements Engineering Conference*, 2011, pp. 47-56
- [23] Pikkarainen M, Haikara J, Salo O, Abrahamsson P, Still J. The impact of agile practices on communication in software development. *Empirical Software Engineering*, 2008, 13(3):303-337
- [24] Herbsleb J D, Grinter R E. Splitting the organization and integrating Conway's law revisited the code. In *21st International Conference on Software Engineering*, 1999, pp. 85-95
- [25] Whitehead J. Collaboration in software engineering: A roadmap. In *International Conference on Software Engineering*, 2007, pp. 214-225
- [26] Zager D. Collaboration as an activity coordinating with pseudo-collective objects. *Computer Supported Cooperative Work*, 2002, 11(1-2):181-204
- [27] Cabitza F, Simone C. Computational coordination mechanisms: A tale of a struggle for flexibility. *Computer Supported Cooperative Work*, 2013, 22(4-6):475-529
- [28] Cockburn A, Highsmith J. Agile software development: The people factor. *Computer*, 2001, 34(11):131-133
- [29] Damian D, Marczak S, Kwan I. Collaboration patterns and the impact of distance on awareness in requirements-centred social networks. In *15th IEEE International Requirements Engineering Conference*, 2007, pp. 59-68
- [30] Eckstein J. Agile software development with distributed teams: Staying agile in a global world. New York: Dorset House Publishing, 2013

-
- [31] Redmiles D, Van Der Hoek A, Al-Ani B, Hildenbrand T, Quirk S, Sarma A, Filho R S S, de Souza C, Trainer E. Continuous coordination: A new paradigm to support globally distributed software development projects. *Wirtschaftsinformatik*, 2007, 49(1):28-38
- [32] Herbsleb J D, Mockus A, Finholt T A, Grinter R E. Distance, dependencies, and delay in a global collaboration. In *ACM Conference on Computer Supported Cooperative Work*, 2008, pp. 319-328
- [33] Sengupta B, Chandra S. A research agenda for distributed software development. In *28th International Conference on Software Engineering*, 2006, pp. 731-740
- [34] Valetto G, Helander M, Ehrlich K, Chulani S, Wegman M, Williams C. Using software repositories to investigate socio-technical congruence in development projects. In *4th IEEE International Workshop on Mining Software Repositories*, 2007, pp. 1-4
- [35] Syeed M M M, Hansen K M. Socio-technical congruence in the ruby ecosystem. In *ACM International Symposium on Open Collaboration*, 2014, pp. 1-9
- [36] Georgas J C, Sarma A. STCML: An extensible XML-based language for sociotechnical modeling. In *4th International Workshop on Cooperative and Human Aspects of Software Engineering*, 2011, pp. 61-64
- [37] Valetto G, Chulani S, Williams C. Balancing the value and risk of socio-technical congruence. In *Workshop on Sociotechnical Congruence*, 2008, pp. 1-4
- [38] Sarma A, Herbsleb J, Van Der Hoek A. Challenges in measuring, understanding, and achieving social-technical congruence. In *Socio-Technical Congruence Workshop*, in conjunction with the *International Conference on Software Engineering*, 2008, pp. 1-12
- [39] Šmite D, Galviņa Z. Socio-technical congruence sabotaged by a hidden onshore outsourcing relationship: Lessons learned from an empirical study. In *International Conference on Product Focused Software Process Improvement*, 2012, pp. 190-202
- [40] Syeed M M M, Hammouda I. Socio-technical congruence in OSS projects: Exploring Conway's law in FreeBSD. In *IFIP International Conference on Open Source Systems*, 2013, pp. 109-126
- [41] Yu L, Ramaswamy S. Using bug reports as a software quality measure. *International Journal of Information Quality*, 2013, 3(2):164-183
- [42] Fenton N, Kitchenham B. Validating software measures. *Software Testing, Verification*

and Reliability, 1991, 1(2):27-42

[43] MacKellar B K. Analyzing coordination among students in a software engineering project course. In 26th IEEE Conference on Software Engineering Education and Training, 2013, pp. 279-283

[44] Hoegl M, Georg H. Teamwork quality and the success of innovative projects: A theoretical concept and empirical evidence. *Organization science*, 2001, 12(4):435-449

How to cite this article:

Suali A.J., Fauzi S.S.M., Nasir M.H.N.M., and Sobri W.A.W. M. A Brief Method to Assess the Association of Socio and Technical Dependencies on Software Quality. *J. Fundam. Appl. Sci.*, 2017, 9(5S), 101-114.