Full paper

Sains Humanika

Assessing the Impact of Socio-Technical Dependencies on Software Defect

A.J. Suali[a], S.S.M. Fauzi[a*], M.H.N.M.Nasir[b], W. A. W. M. Sobri[a]

[a]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 02600 Arau, Perlis, Malaysia*
[b]*Faculty of Computer Science & Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia*

*Corresponding author: shukorsanim@perlis.uitm.edu.my

Abstract

One of the measures used to fit or align social and technical dependencies is known as Socio-Technical Congruence (STC). However, STC is a new and vastly unexplored area. To our best knowledge, there has been limited research done on the use of STC in software engineering projects. Hence, this provides an opportunity to extend the body of knowledge by investigating the impact of STC on other measures such as software quality. This paper outlines the motivation behind this said research, the methodology and expected findings of this research. This research aims to provide further information about the domain of STC in software engineering field, specifically its relationship with software quality. The expected result of this research is a higher level of congruence in a development project can yield a higher quality project.

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

■1.0 INTRODUCTION

Coordination among developers is an element of Socio-Technical Congruence (STC) called actual coordination (Kwan & Damian, 2011). Another element is coordination requirement, where tasks are dependent on each other and require developers to work together. This model on measuring the relationship between actual coordination and coordination requirement to ensure congruence of both these relationships was introduced by Marcelo (Cataldo, Wagstrom, Herbsleb, & Carley, 2006).

Prior research on STC found that higher coordination needs for a task leads to more reliance on communication and written artifacts for the explicit transfer of knowledge, and more congruence to reduce resolution time (Cataldo et al., 2006). Using the STC model, alignment between social relationship and technical relationship impacts the development productivity (Cataldo, Herbsleb, & Carley, 2008). Apart from that, previous studies revealed that STC model has an effect in software build success (Kwan, Schröter, & Damian, 2011). Other related studies on STD models discovered that it can be used in large projects that require high attention to be given to excellent execution of coordination congruence and implicit coordination, as it is connected with the task performance in all project types and is essential in increasing performance (S.S.M.Fauzi, 2014).

Little is known as to how STC model impacts software quality. This research intends to investigate the impact of STC on software quality in software engineering through an analysis of the relationship between STC and software quality. This is crucial in validating STC as an established theory in software engineering. Hence, the purpose of this research is to construct a model on the relationship between STC and software quality. The model will then be used to investigate the impact of STC on software quality in software engineering projects. This paper discusses the motivation behind this research and the proposed methodology that will be used throughout. The expected conclusion of this research is the higher the congruence, the better the product quality.

This paper is structured as follows: Firstly, the paper details the literature review of STC. The next section describes the methodology of this research. Subsequently, the expected result of the research is presented. Lastly, this paper concludes all the findings.

■2.0 LITERATURE REVIEW

Every isolated task is often dependent on other components, thus, leading to coordination among components called task dependencies. Once there are dependent tasks, developers may need to coordinate with each other. The type of approach and motivation to coordinate among developers needs to be inspected and "fit" between task dependencies and coordination among these developers. This research will use the STC model to calculate congruence in order to measure the "fit" (Cataldo et al., 2006). The term "fit" means task dependencies that occur during development correlate with developers who are individually carrying related tasks (Cataldo et al., 2008).

More specifically, STC model is used to compute the congruence between coordination requirement and actual coordination in software engineering projects (Cataldo et al., 2006). To study alignment between social and technical dependencies, STC model is introduced to understand the concept and calculate congruence between these factors (Cataldo et al., 2006). This model is composed of two elements which are technical and social relationship. Technical relationship or coordination requirement happens when two developers who have related tasks need to discuss and work together (Kwan & Damian, 2011). Meanwhile, social relationship or actual coordination is considered as communication that occurs among developers during development (Kwan & Damian, 2011).

STC model applies the concept of Conway's law to create a model that calculates coordination among developers in regards to task dependencies which is the communication among developers to relate to the task that they have to complete (Cataldo et al., 2008). Conway's law suggests that the design architecture must mirror the communication among developers who develop the software (Georgas & Sarma, 2011). STC model is interpreted as the alignment between coordination requirement established by technical dependencies, and actual coordination by developers who engage in the software engineering development (Kwan et al., 2011).

Using STC model, the alignment between social relationship and technical relationship would impact development productivity (Cataldo et al., 2008). Empirical evidence suggests that STC model helps to improve team productivity and coordination between developers (Georgas & Sarma, 2011). Any breakdown in coordination will impact software quality (Osterweil, 1997).

This research intends to illustrate whether congruence between social and technical factors will affect software quality when an STC model is in place.

3.0 METHODOLOGY

Figure 1 illustrate basic theoretical model used in this research.

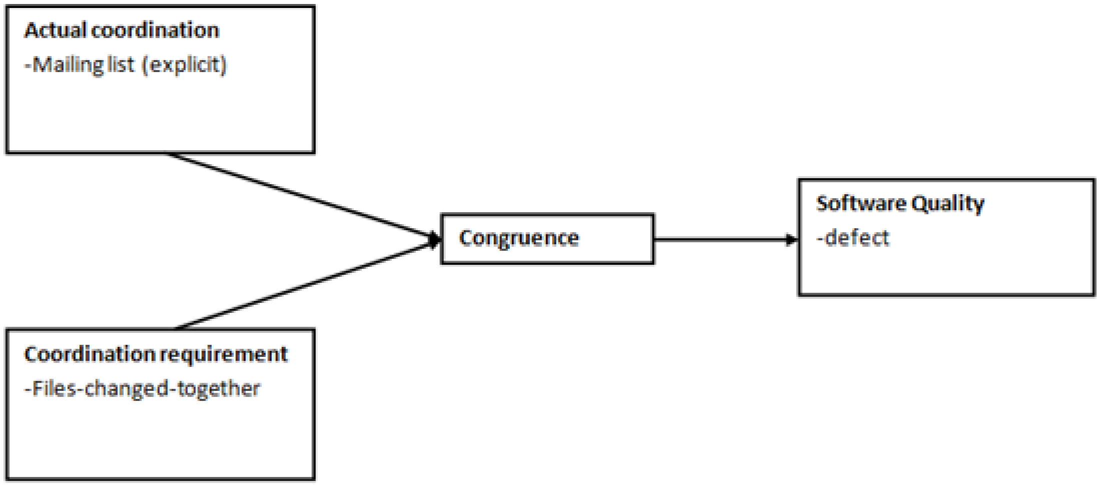

Figure 1 Proposed STC model

The STC model proposed in Figure 1 involves several elements, such as actual coordination, coordination requirement, and software quality. These elements include dependent variables, independent variables and control variables. Each variable is identified based on the present circumstance taking place in software engineering projects.

The first element, coordination requirement, is derived from a relationship in which two or more developers coordinate based on their task performance (Cataldo et al., 2008). This coordination dependency occurs in files-change-together. Every change in file is usually made by developers. The changes can be tracked in Modification Request (MR). Coordination happens when two or more developers change the same files in MR. Coordination Requirement is achieved by the multiplication of task assignment (TA) and task dependencies (TD) and transpose of task assignment (TAT).

The following equation is used to calculate coordination requirement:

$$\mathbf{C_R = T_A * T_D * T_A^T} \qquad \text{(Eq. 1)}$$

Actual coordination is the second element in measuring communication activities among developers. Developers who communicate together also intentionally work together on their tasks. Actual coordination is based on communication from the mailing list. It is calculated by developing table matrices that indicate communication that occurs between person i and person j. This represents interaction among developers.

The objective of this research is to compute congruence between social relationship and technical relationship. Congruence happens when there is an alignment between coordination requirement and actual coordination. Congruence is calculated by comparing the metric between coordination requirement and actual coordination. The developed equation below is used to compute congruence:

$$\begin{gathered}\mathbf{Diff\ (CR,\ AC = card\ \{\ diffij\ |\ CRij > 0\ \&\ ACij > 0\}} \\ \mathbf{|CR| = card\ \{CRij > 0\ \},} \\ \mathbf{Congruence = Diff\ (CR,AC)\ /\ CR} \qquad \text{(Eq. 2)}\end{gathered}$$

This research will use linear regression to identify the association between STC and software quality in software engineering projects.
In addition, to investigate the impact of STC on software quality, defect density will be used. As known, defects are found when testers carry out test cases. The test result is referred to as defect. The defect can also be called bug, issue or problem. It is measured by the overall number

of defects in a project, where total number of defects is divided by size of the project. The expected result is low number of defects and high congruence. Other variables include priority of what is important and the order in which tasks should be fixed.

To conduct this research, an open source project (OSS) will be selected. This research chooses Yetus project that comprises 20 developers. This is to ensure that communication among developers takes place during coordination activities. Yetus project is chosen as it provides libraries and tools that enable contribution, releases processes for software projects and helps community-driven software projects improve their contribution and release processes.

This research will fully utilize R script to perform data extraction, cleaning, and statistical regression. Thus, the script needs to be developed and tested before it is implemented in the actual project.

■4.0 EXPECTED RESULTS

The research is still at the preliminary stage. The expected outcome of this research is that STC has a significant and direct relationship with software quality. This research tries to provide empirical evidence of the impact of STC on software quality. By using defects as a measurement of software quality, this research expects a decreasing number of defects where congruence is high.

■5.0 CONCLUSION

Through the conducting studies on a broad spectrum of software engineering projects, this research hopes to demonstrate the important role of STC in these projects. Thus, this research will provide another piece of evidence that further signifies the impact of STC on software.

Acknowledgment

This research 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)".

References

Cataldo, M., Herbsleb, J. D., & Carley, K. M. (2008). Socio-Technical Congruence: A Framework for Assessing the Impact of Technical and Work Dependencies on Software Development Productivity. *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, 2 – 11.

Cataldo, M., Wagstrom, P. A., Herbsleb, J. D., & Carley, K. M. (2006). Identification of Coordination Requirements: Implications for the Design of Collaboration and Awareness Tools. *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work*, 353–362.

Georgas, J. C., & Sarma, A. (2011). STCML : An Extensible XML-based Language for Socio- Technical Modeling. *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering*, 61–64.

Kwan, I., & Damian, D. (2011). Extending Socio-Technical Congruence with Awareness Relationships. *Proceedings of the 4th International Workshop on Social Software Engineering*, 1 – 8.

Kwan, I., Schröter, A., & Damian, D. (2011). Does Socio-Technical Congruence Have an Effect on Software Build Success? A Study Of Coordination in a Software Project. *IEEE Transactions on Software Engineering*, 37(3), 307–324.

Osterweil, L. J. (1997). Improving the Quality of Software Quality Determination Processes. *Quality of Numerical Software*, 90–105.

S. S. M. Fauzi, (2014). Developer Coordination in Software Engineering Projects (Unpublished doctoral dissertation). The University of New South Wales.