Full paper

Sains Humanika

Measuring The Impact of Socio-Technical Congruence in a Different Types of Software Life Cycle

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

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

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

Abstract

Measuring developer coordination is a fundamental challenge and complex task in software development organizations. One way used to conceptualize and measure developer coordination is known as 'Socio-Technical Congruence (STC)', which is fit between the coordination requirements established by the dependencies among tasks and the actual coordination activities carried out by the developers. However, STC has not been widely accepted as a broad theory. This is for the reason that, STC is relatively new, and there are many fundamental questions that need to be addressed and understood. This research intend to construct a model of the relationship between STC and project performance in the different types of software development lifecycle. The model constructed can be used to provide additional evidence to the body of knowledge, which will further strengthen the STC theory. Therefore, we outline research questions, the proposed method used to conduct the research, potential contributions and the expected results of the research.

Keywords: Coordination; socio-technical congruence; software development; actual coordination; coordination requirement

■1.0 INTRODUCTION

Measuring developer coordination in project development is important to increase project performance (Valetto et al., 2007). Even so, measuring developer coordination is a complex task in software development organizations (Cataldo, Herbsleb, & Carley, 2008). Thus, ‘Socio-Technical Congruence’ (STC) is proposed by Cataldo et al as a technique to conceptualize and measure developer coordination. STC is a measure that assesses the fit between the coordination requirement imposed by the task dependency and the actual coordination activity carried out by the developers (Cataldo et al., 2008; L. Jiang, K. M Carley, 2010). Result from Cataldo et al research suggests that the higher the STC, the greater the chances to increase in task performance (Cataldo, Wagstrom, & Carley, 2006).

However, despite the acknowledged impact of STC, there are still many fundamental questions that need to be addressed. For instance, software engineering projects might employ different types of software development lifecycle in each project; for example, waterfall model, agile development model, rapid application development (RAD) model, V and incremental model and many others. This leads to the differences of the project characteristics and might also affect the organization of the developers.

This study intend to construct a model of the relationship between STC and project performance in the different types of software development lifecycle.

The research paper is structured as follows. First section, discussion on the literature reviews. Followed by, discussion on the research questions. After that, the research methodology, followed by the expected potential contributions. Then, discussion on expected result. The final chapter infers the conclusion of this study and future research.

■2.0 LITERATURE REVIEW

Software development lifecycle (SDLC) is a process of planning, development, deployment and until the maintenance of software (Ruparelia, 2010). There is an abundance of models in SDLC, for instance waterfall, agile, spiral, incremental, v-model, RAD model and others. SDLC is essential in software development project in order to develop software systematically with lowering cost, time and resource. Along with that, coordination between developers in SDLC is vital to ensure the achievement of development goals.

Coordination has become more challenging in line with the increase of distributed project worldwide. The distributed location shown negative impact on task resolution time and perception of developers team (Nguyen-duc, Cruzes, & Conradi, 2015). Hence, STC is introduced since coordination needs for a more powerful method to establish technical coordination (Herbsleb et al., 2008). STC derived from the conceptualization of Conway, which state that design system tend to copy structured organization that design it (Conway, 1968). Based on Conway’s Law, a system is developed effectively with smaller technical dependencies in order to prevent communication overhead (Conway, 1968).

Prior research use congruence as a method to fit measure between task dependencies and coordination activities (Cataldo et al., 2006). The same study also found that, congruence is advantageous in lessen development time (Cataldo et al., 2006). When coordination requirement fulfilled through formed of congruence, performance is beneficial in which shorter time required to fix the error (Helander, Valetto, & Williams, 2008). Another study discovered that , teams that worked closely together were able to build software successfully when socio-technical alignment was high, this is in contrast with integration build where increase in congruence lead to decrease in build success probability (Irwin, Adrian, & Daniela, 2011). Other research identify that effective software project achieved by gained domain knowledge through cross-functional communication among members (Damian, Helms, Kwan, Marczak, & Koelewijn, 2013). In addition, team with high interdependence and complexity involved can be effective only if they keep up with their communication (Mirani, 2007). Little is known on how STC impact in different from SDLC model.

■3.0 RESEARCH QUESTION

The complexity of identifying and managing product dependencies is a paramount importance in ensuring the success of coordination that later led to the success of the productivity and quality of project (Cataldo et al., 2008). Prior literature indicates congruence indeed affects the task performance in the distributed projects, where resolution time reduce when congruence increase (Cataldo et al., 2008). In this case, congruence is gained through coordination capabilities (Herbsleb et al., 2008). To develop STC as an established theory and measure of coordination in software engineering projects, more fundamental studies are required. This will help to provide additional knowledge and evidence on STC in software engineering, including the boundaries and its relevance. Opportunity exists, therefore it is vital to do further research on STC in the different types of software development lifecycle. This research will be guided by the following research question:

RQ1: ***How does socio-technical congruence impact task performance in the different types of software development lifecycle?***

■4.0 RESEARCH METHODOLOGY

A preliminary study will be conducted to understand the nature of developer coordination in the different SDLC model in software engineering project. Based on the preliminary study, a basic theoretical model will be proposed. Initial model for this research is illustrated in Figure 1.

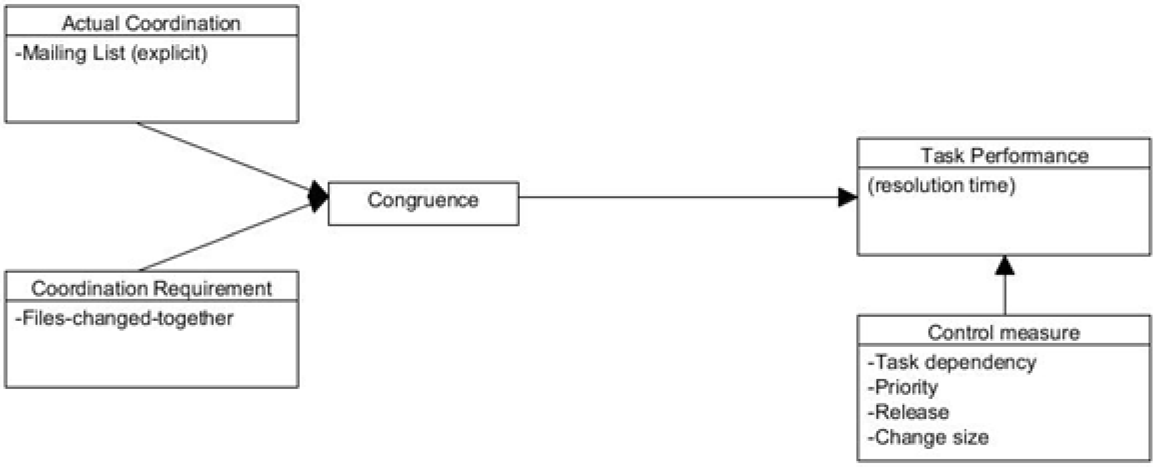

Figure 1 Proposed Socio-Technical Congruence (STC) model

The descriptions about each variable are discussed as follows:

Coordination requirement is a relationship that indicates that two developers should be coordinated based on their task assignment (Cataldo et al., 2008, 2006; Helander et al., 2008). Data for coordination requirement will be extracted from various sources, for instance, modification request report, version control system and also source code itself (Cataldo et al., 2008). When coordination requirement is achieved, through establish of congruence, the performance will gain advantage when lower time is required to solve the error (Cataldo et al., 2008).

Actual coordination is the interaction between developers that occurs in the project (Cataldo et al., 2008, 2006; Marczak, Kwan, & Damian, 2010). This is represented by explicit communication (as determined from mailing lists) (Kwan, 2011). Data for actual coordination will be collected in various communication channel like instant messenger, mailing list (Marczak et al., 2010). However, actual coordination among developers who do not have related coordination need will not affect the congruence (Marczak et al., 2010).

Congruence is the match between coordination requirements and actual coordination (Cataldo et al., 2006). Congruence will be computed by comparing coordination requirement and actual coordination.

Task performance is the time taken to resolve a particular task (Valetto, Jose, & Williams, 2008). It is measured as the overall time taken to complete the task (resolution time).

Control measure in this study covered four elements. First is priority, also define as the importance and order in which the task should be fixed. It will be measured as a value assigned to Modification Requests (MRs) to represent the level of criticality of the MRs. Second is change size, which is proxy for the actual amount of development work done. Change size is computed as the number of files and components that are modified as part of the change for the MR. Third is release, the ordinal number of the software version contributed to be the task. Lastly task dependency, is a measure of the affected MRs which need to be referred to in order to accomplish the task (Cataldo et al., 2008). It is measured as a number of MRs that are depended on in order to perform the task *(#of MRs)*.

Open source project will be selected to conduct the research. This research will fully utilize R script to perform data extraction, cleaning, and statistical regression. Thus, script needs to be developed and tested before being implemented.

The study will use archived data from a MR bugs repository and mailing list. Data for this study will be extracted using Mining Software Repository (MSR) techniques. The MR bugs repository consists of a bundle of information related to changes in each project. This information can be manipulated and analysed to understand coordination requirement among the developers.

Linear regression will be used to identify the association between STC and project performance in the different software development lifecycle in software engineering projects.

■5.0 POTENTIAL CONTRIBUTION

This research will potentially provide the following contributions: A model related to the relationship between STC and project performance in the different type of software development lifecycle. The model will also lead to the following knowledge: 1) Provides empirical evidence of the validity and limitations of applicability of STC in the different types of software development lifecycle 2) Update the STC literature on the impact of STC on the different types of software development lifecycle 3) Provide a new understanding on measuring coordination and congruence.

■6.0 EXPECTED RESULT

Therefore, we expect that by using STC measure between coordination requirement and actual coordination, task performance (resolution time) on various SDLC models will be identified. This study potentially provides a model related to the relationship between STC and project performance in the different types of software development lifecycle.

The advantage of this study in software industry is it helps to reduce overhead cost of coordination through aligining social and technical approach in software development project. In reality, project nowadays have become more complex, coupled with worse condition like when project is over deadline. Hence, in order to resolve this problem, socio-technical congruence is introduced to measure congruence that will result in the level of performance between team members. However, socio-technical congruence is a relatively new area of research in software engineering. Thus, this study benefits by providing informations about the impact of task performance in software development lifecycle models. Then, it will be a future reference for researcher who study related to this topic.

■7.0 CONCLUSION

The intention of this research is to investigate the impact of STC on task performance in the different SDLC models. In this paper, an outline of the research questions, research methods, potential contributions and expected results of the research are presented. Yet, to our knowledge there are limited research on used of socio-technical congruence method to conceptualize and measure developer coordination. As the software industry has become more complex, organizations sometimes are forced to add manpower when projects fall behind schedule. Findings from the research may help the software industry to pay more attention to aligning and accounting for social and technical factors in software development. This is because the alignment of social and technical factors can assist in substantially decreasing the cost of the unavoidable overhead of coordination in software development activities.

Acknowledgment

We gratefully acknowledge support by Ministry of Education (MOE), Malaysia and Universiti Teknologi MARA, Malaysia under Research Acculturation Collaborative Effort under Grants No 600-RMI/RACE 16/6/2 (10/2014).

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., & 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.

Conway, M. E. (1968). How Do Committees Invent. *Datamation Magazine*, April Issue, 28–31.

Damian, D., Helms, R., Kwan, I., Marczak, S., & Koelewijn, B. (2013). The Role of Domain Knowledge and Cross-Functional Communication in Socio-Technical Coordination. *35th International Conference on Software Engineering (ICSE)*, 442–451.

Helander, M., Valetto, G., & Williams, C. (2008). An Analysis of Congruence Gaps and Their Effect on Distributed Software Development. *Proceedings of the 1st International Workshop on Socio-Technical Congruence*, 1–10.

Herbsleb, J., Cataldo, M., Damian, D., Devenbu, P., Easterbrook, S., & Mockus, A. (2008). Socio-technical congruence (STC 2008). *Companion of the 13th International Conference on Software Engineering - ICSE Companion '08*, 1027.

Irwin, K., Adrian, S., & Daniela, and 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(7), 307–324.

Kwan, I. H.-B. (2011). The Study of Socio-technical Coordination Using A Socio-technical Congruence Model. *ProQuest Dissertations and Theses*. Dissertation, University of Victoria.

L. Jiang, K. M Carley, J. O. (2010). Measuring Team Performance Using Socio- Technical Congruence. *Computational Analysis of Congruence of Complex System Group Technical Report*, TR-2010-01.

Marczak, S., Kwan, I., & Damian, D. (2010). Investigating Collaboration Driven by Requirements in Cross-Functional Software Teams. Proceeding of the *2009 Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills*, 15-22.

Mirani, R. (2007). Procedural Coordination and Offshored Software Tasks: Lessons from Two Case Studies. *Information and Management*, 44(2), 216–230.

Nguyen-duc, A., Cruzes, D. S., & Conradi, R. (2015). The Impact of Global Dispersion on Coordination, Team Performance and Software Quality – A Systematic Literature Review. *Information and Software Technology*, 57, 277–294.

Ruparelia, N. B. (2010). Software Development Lifecycle Models. *ACM SIGSOFT Software Engineering Notes*, 35(3), 8.

Valetto, G., Helander, M., Ehrlich, K., Chulani, S., Wegman, M., & Williams, C. (2007). Using Software Repositories to Investigate Socio-Technical Congruence in Development Projects. *Proceedings of the Fourth International Workshop on Mining Software Repositories*, 0–3.

Valetto, G., Jose, S., & Williams, C. (2008). Balancing the Value and Risk of Socio-Technical Congruence. *Workshop on Sociotechnical Congruence*, 4.