Synthesizing the Literature on the Issues of Coordination and its Impact on Software Quality

A. J. Suali[1], S. S. M. Fauzi[1], M. H. N. M. Nasir[2] and W. A. W. M. Sobri[1]
[1]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perlis, Malaysia.*
[2]*Faculty of Computer Science & Information Technology, University of Malaya, Kuala Lumpur, Malaysia.*
shukorsanim@perlis.uitm.edu.my

***Abstract*—High quality of products is one of the main priorities in development. Delivering products of good quality demands a high level of coordination among developers and maintaining coordination is challenging. This paper intends to investigate the issues of coordination and its impact on software quality in software engineering projects. The researchers applied Systematic Literature Review (SLR) to perform this study. Among the coordination issues discovered are language barriers, intercultural, inefficient communication, trust, lack of project flow understanding, different time zones, dependency issues, strategic issues, knowledge management, geographical distance, awareness, and organizational boundaries. All these obstacles then significantly impact software quality.**

***Index Terms*—Coordination; Systematic Literature Review; Software Quality.**

I. INTRODUCTION

Software engineering projects in nature are teamwork oriented [1]. Generally, developers need to collaborate to ensure the project development is delivered on time, is low cost and meets the project schedule. Occasionally, developers do co-operate with outsiders in order to cut-cost, share expertise and exchange technology [2]. Coordination with other developers especially in terms of distributed sites is not an easy task because they need to consider many aspects such as culture, language, time working and schedule [3].

Apart from that, complex project development happens when certain software development projects require many developers to coordinate with each other [4]. Therefore, this leads the organization to distribute work in order to complete the project based on scheduled dates [5].

In the development process, high level of attention among developers is needed to ensure coordination is efficient and the project can be completed in time. As known, coordination is defined as “body of principles about how activities can be coordinated, that is, about how actors can work together harmoniously” [6]. Moreover, coordination takes place between developers in any situations, ranging from small tasks that need consideration even to large-scale modifications. As long as the project is related to the development, developers must carry out actions together [7]. Coordination occurs when they work on achieving the same goals and interdependent tasks are conducted by different group members [8].

Managing coordination especially in terms of distributed work is difficult as it not only impacts the team performance but also affects the quality of software during development [9] [10]. Quality is defined as the degree of requirement satisfaction of clients when their demands are met [11]. Besides, quality software is defined as “the efficient, effective, and comfortable use by a given set of users for a set of purposes under specified conditions” [12]. Software quality can be measured through time, cost or productivity [13]. Moreover, software quality attributes such as reliability, maintainability, correctness and flexibility are used to describe characteristics of the product to ensure it fulfills the requirement needs [14]. Therefore, developing high-quality products requires a lot of effort including the challenges developers have to face such as issues of coordination during development.

There are various empirical studies about issues of coordination and its impact on software quality in software engineering. This study intends to synthesize the issues of coordination and its influence on software quality in software engineering projects. Furthermore, this study is about to update the issues of coordination during development in the current literature.

This paper is structured as follows. The next section explains on the methodology used in this study which is Systematic Literature Review (SLR). Next, the findings from Systematic Literature Review on the issues of coordination and its impact on software quality in software engineering projects are presented. This section is followed by a discussion of the findings of this study. Finally, the conclusion summarizes the issues of coordination and its impact on software quality in software engineering projects.

II. METHODOLOGY

Systematic Literature Review (SLR) was applied as a guideline to conduct this proposed study [15]. SLR assisted this study through “identifying, evaluating, and interpreting all available research relevant to the particular research question, topic area or phenomenon of interest” [16]. SLR involves several phases including research question, search process using certain keywords, inclusion and exclusion, result, and discussion. Every phase that was used to perform this study will be discussed in detail.

A. Research Question

This research question acted as a guide to carry out this study. The selection of research question was very important in directing this study to ensure the objective was achieved. Thus, the research question derived from the objective of this study is:

RQ1: What are the issues of coordination and its impact on software quality in software engineering projects?

B. *Search and keywords criteria*

Search processes were from digital databases derived from the research question by using certain keywords. The keywords are divided into three categories which are "coordination issues", "software quality", and "software engineering project" as shown in Table 1. Search terms were built from the keyword criteria using "OR" and "AND" Boolean operator. The following are the lists of online databases used:

i. IEEE Xplore
ii. ACM Digital Library
iii. Springer Online Journal Collection
iv. Science Direct
v. Google Scholar

Table 1
Keywords searches for this study

Criteria	Keywords
Coordination issue	Coordination issue(s) Coordination challenge(s) Coordination problem(s) Coordination constraint(s)
Software quality	Software quality Quality of product
Software engineering project	Software engineering Software engineering project(s) Software project(s) Software development Open Source Project(s) Open source development

C. *Inclusion and exclusion criteria*

The next step was to screen the paper that was obtained from digital databases. The screening was done using inclusion and exclusion criteria to answer the research question. The criteria for inclusion (I) and exclusion (E) are as follows:

I1. Papers published in workshops, conferences and journals that directly mention issues of coordination and its impact on software quality.

I2. Papers that cover coordination issues and state its impact on software quality in software engineering projects.

E1. The paper which discusses coordination but does not engineering projects.

E2. Posters, abstracts, article summaries, and slide presentations.

There are plenty of literature discuss issues of coordination in software engineering in the digital databases. We apply inclusion and exclusion to gather the related study to provide significant evidence that discusses about the research question. The total number of papers gathered from the digital databases after screening using exclusion and inclusion was 104. At the first stage, screening of papers started from the title and keywords used in the study. Next, a total of 63 papers remained after screening through the abstract and the other papers were removed. The remaining papers will potentially be used for this study. The last stage comprised of reading all the contents of the papers and results indicate only 20 papers were discussed critically according to the objective, and 4 duplicated papers related to this study were removed.

III. Result and Discussion

A. *What are the issues of coordination and its impact on software quality in software engineering projects? (RQ1)*

The previous section described the research question, search keyword criteria and the selection criteria used to search relevant studies. This section summarizes the findings after completing the screening stage. The final round of screening showed that 20 papers discussed about coordination challenges and its impact on software quality in software engineering projects. The full list of papers that has been gathered can be seen in Appendix A.

Table 2 illustrates the journal publication by years to identify the issues of coordination during development. The result shows that the study about issues of coordination still become a concern because of the issues of coordination has been identified and discussed in every year.

Table 2
Journal publication (by years)

Years	Percentage (%)	Frequency
2000	5	1
2001	5	1
2002	0	0
2003	0	0
2004	10	2
2005	5	1
2006	0	0
2007	5	1
2008	15	3
2009	5	1
2010	5	1
2011	10	2
2012	5	1
2013	10	2
2014	10	2
2015	5	1
2016	5	1
Total	100	20

The investigators of this study managed to collect information regarding issues of coordination among developers. The intention is to identify issues that disrupt coordination and its impact on software quality in software engineering projects. This study is important to other researchers in finding evidence on how coordination issues affect software quality in these projects. There are several issues of coordination that were identified during the course of this study.

Figure 1 shows the finding about issues of coordination in software engineering project. Issues in software development that identified distract the coordination among developers, where we expect its impact on software quality. The finding suggests that geographical distance had the highest frequency of interrupting coordination during development, at twenty-one percent. Software engineering teams tend to work with outsiders to exchange expertise and skills that helps in fulfilling customer needs. Furthermore, inefficient communication had the second highest frequency of disrupting coordination among developers, at fifteen percent. When developers decide to distribute work, they need to be prepared in dealing with different cultures among developers. From the findings, it is known that intercultural issues were also a factor in hampering coordination during developments.

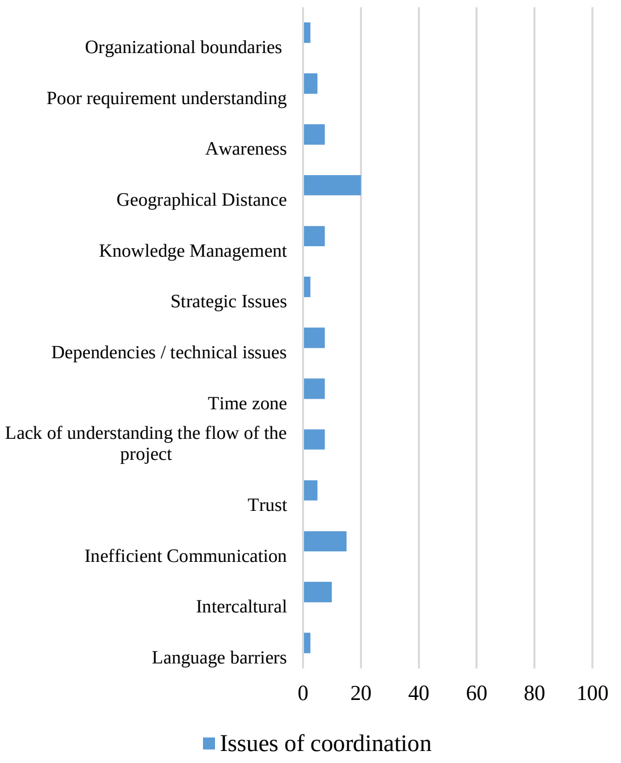

Figure 1: Issues of coordination

Other issues such as lack of project flow understanding, different time zones, dependency issues, knowledge management and awareness represent seven percent for each finding. Trust issues, at five percent, and poor requirement understanding also interrupted the coordination during development. Language barriers, strategic issues and organizational boundaries make up three percent of findings that contribute to issues of coordination during development. Any mistakes or errors in managing issues of coordination among developers can impact the software quality in software engineering projects.

B. *What are the issues of coordination and its impact on software quality in software engineering projects? (RQ1)*

Table 3 illustrates the coordination challenges and its impact on software quality. This empirical evidence exhibits the challenges that take place during development. From the findings on coordination challenges, researchers tried to classify the impact of software quality using certain criteria of software quality based on coordination impact. In order to construct a quality software product, the software process plays an important role in order to satisfy the requirements product. The quality process might in turn affecting the software quality in software engineering project other than inspecting the product quality by examining through software quality assurance activity. Examples of software processes that can be implemented in any software development project are CMM, CMMi, ISO 15504 and McCall [11] [37] [38]. Meanwhile for the software product quality implement standardization such as ISO 9126, ISO 14598, ISO 25051, ISO 15026 and ISO 15910 [39]. Prior study suggests that issues in coordination impact on software quality but we did not see any paper that discusses in deep about specific quality attributed affected by the issues of coordination [19]. This study intends to use ISO9126 to classify the impact of the quality product because of the coordination issues [40]. Table 3 illustrates the result of our study.

Table 3
Coordination challenges and its impact on software quality

Issues	Paper Id	Impact of the Software Quality
Language barriers	[1]	Satisfaction; fault on product's specification; accuracy; correctness
Intercultural	[2] [3] [4] [5]	Correctness
Inefficient Communication	[3] [6] [5] [7] [8] [9]	Product's specification; accuracy; correctness; consistency
Trust	[3] [10]	Increase time and cost; productivity; integrity; satisfaction; accuracy
Lack of understanding the flow of the project (Unaware of the progress)	[3] [11] [12]	Efficiency; effectiveness; completeness
Time zone	[13] [12] [12]	Integrity; accuracy; satisfaction
Dependencies / technical issues	[14] [4] [15]	Productivity;
Strategic Issues	[4]	Efficiency
Knowledge Management	[16] [4] [10]	Integrity; accuracy
Geographical Distance	[2] [17] [18] [6] [7] [19] [20] [10]	Completeness; satisfaction
Awareness	[6] [15] [19]	Productivity
Poor requirement understanding	[3] [7]	Correctness; understandability; decreasing software quality assurance
Organizational boundaries	[14]	Productivity; correctness

IV. CONCLUSION

This study found substantial empirical evidence that discusses issues of coordination and its impact on software quality in software engineering projects. By applying Systematic Literature Review (SLR), several issues of coordination were discovered including language barriers, intercultural, inefficient communication, trust, lack of project flow understanding, different time zones, dependency issues, strategic issues, knowledge management, geographical distance, awareness, and organisational boundaries that have a significant impact on software quality. The researchers have also identified several software quality attributes that are significantly affected by coordination issues, which are satisfaction, fault on product's specification, accuracy, correctness, consistency, increase in time and cost, productivity, integrity, efficiency, effectiveness, completeness, integrity, understandability and decrease in software quality assurance. It is then concluded that many aspects can impact software quality and it is the responsibility of the organization to monitor the development progress to ensure quality products can be manufactured and customer satisfaction is met.

V. FUTURE WORK

Based on the empirical evidence gathered from this study, the researchers are of the opinion that coordination issues do impact software quality in software engineering projects. For instance, these issues affect productivity in terms of dragging the completion time, not meeting the satisfied requirements, increasing the cost and time spent. Future work will focus on measuring congruence through the alignment between developer coordination and dependent tasks in order to investigate its relationship with software quality.

APPENDIX A

Table 4
List of Finalized Papers on Coordination Challenges and Its Impact on Software Quality in Software Engineering Projects

Id	Authors	Years	Title of Paper	Issue(s) identified
[17]	Yi Wang	2015	Language matters	Language barriers
[18]	Bernard Wong & Sazzad Hasan	2008	Cultural Influences and Differences in Software Process Improvement Programs	Geographical Distance; Intercultural
[19]	Sanjay Misra & Luis Fernández-Sanz	2011	Quality Issues in Global Software Development	Inefficient communication; Intercultural; Trust; Lack of project flow understanding (unaware of the progress); Poor requirement understanding
[20]	J. Alberto Espinosa & Erran Carmel	2004	The Impact of Time Separation on Coordination in Global Software Teams: a Conceptual Foundation	Time zone
[21]	James D. Herbsleb, Daniel J. Paulish & Matthew Bass	2005	Global Software Development at Siemens: Experience from Nine Projects	Knowledge management
[22]	Beth Yost, Michael Coblenz, Brad Myers, Joshua Sunshine, Jonathan Aldrich, Sam Weber, Matthew Patron, Melissa Heeren, Shelley Krueger and Mark Pfaff	2016	Software Development Practices, Barriers in the Field and the Relationship to Software Quality	Lack of project flow understanding (unaware of the progress)
[23]	Rafael Prikladnicki & Erran Carmel	2013	Is Time-Zone Proximity an Advantage for Software Development? The Case of the Brazilian IT Industry	Time zone
[24]	Anh Nguyen-Duc & Daniela S. Cruzes	2013	Coordination of software development teams across organizational boundary – An exploratory study	Dependencies/ technical issues; Organizational boundaries
[25]	Rafael Prikladnicki, Jorge Luis Nicolas Audy & Roberto Evaristo	2004	Global Software Development in Practice Lessons Learned	Intercultural; Dependencies/ technical issues; Knowledge management
[26]	Narayan Ramasubbu, Marcelo Cataldo, Rajesh Krishna Balan, & James D. Herbsleb	2011	Configuring Global Software Teams: A Multi-Company Analysis of Project Productivity, Quality, and Profits	Geographical distance
[27]	Nachiappan Nagappan, Brendan Murphy & Victor R. Basili	2008	The Influence of Organizational Structure on Software Quality: An Empirical Case Study	Geographical distance
[28]	Christian Bird, Nachiappan Nagappan, Premkumar Devanbu, Harald Gall & Brendan Murphy	2009	Does Distributed Development Affect Software Quality? An Empirical Case Study of Windows Vista	Inefficient communication; Geographical distance
[29]	Stina Matthiesen, Pernille Bjørn & Lise Møller Petersen	2014	Figure Out How to Code with the Hands of Others: Recognizing Cultural Blind Spots in Global Software Development	Intercultural; Inefficient Communication

[30]	Anum Tariq & Aliya Ashraf Khan	2012	Framework supporting team and project activities in Global Software Development (GSD)	Inefficient Communication;
				Geographical distance;
				Poor requirement understanding

[31]	James D. Herbsleb, Audris Mockus, Thomas A. Finholt & Rebecca E. Grinter	2000	Distance, Dependencies, and Delay in a Global Collaboration	Inefficient communication
[32]	J. Alberto Espinosa, Sandra A. Slaughter, Robert E. Kraut, & James D. Herbsleb	2007	Team Knowledge and Coordination in Geographically Distributed Software Development	Dependencies / technical issues; Awareness
[33]	Andrew Begel	2008	Effecting Change: Coordination in Large-Scale Software Development	Lack of project flow understanding (unaware of the progress); Time zone; Awareness; Geographical distance
[34]	Christof Ebert & Philip De Neve	2001	Surviving Global Software Development	Geographical distance
[35]	Sanjay Mohan & Jude Fernandez	2010	Distributed Software Development Projects: Work Breakdown Approaches to Overcome Key Coordination Challenges	Inefficient communication
[36]		2014	Software Development Outsourcing: Challenges and Opportunities in Nigeria	Trust; Geographical distance

ACKNOWLEDGEMENT

We are grateful to the Ministry of Higher Education for supporting this research through the Research Acculturation Grant Scheme (RAGS) grant (600-RMI/RAGS 5/3 (4/2014).

REFERENCES

[1] J. Grundy, I. Mistrik, A. van der Hoek, and J. Whitehead, “What we know (and do not know) about collaborative software engineering,” *Collab. Softw. Eng.*, pp. 237–239, 2010.

[2] A. Baird and F. J. Riggins, “Planning and Sprinting: Use of a hybrid project management methodology within a cis capstone course,” *J. Inf. Syst. Educ.*, vol. 23, no. 3, pp. 243–257, 2012.

[3] J. D. Herbsleb and D. Moitra, “Global Software Development,” *IEEE Softw.*, vol. 18, no.2, pp. 16–20, 2001.

[4] I. Omoronyia, J. Ferguson, M. Roper, and M. Wood, “A review of awareness in distributed collaborative software engineering,” *Softw. Pract. Exp.*, vol. 40, no. 20, pp. 1107–1133, 2010.

[5] A. Taweel, B. Delaney, T. N. Arvanitis, and L. Zhao, “Communication, knowledge and co-ordination management in globally distributed software development : informed by a scientific software engineering case study,” in *Fourth IEEE International Conference on Global Software Engineering, 2009. ICGSE 2009.*, 2009, pp. 370–375.

[6] T. W. Malone, “What is coordination theory ?,” in *Nasional Science Foundation Coordination Theory Workshop*, 1988, pp. 1–32.

[7] Gutwin, Carl, and Greenberg, “A descriptive framework of workspace awareness for real-time groupware,” in *Computer Supported Cooperative Work (CSCW)*, 2002, pp. 411–446.

[8] S. Whittaker and S. Heinrich, “Meetings of the board : the impact of scheduling medium on long term group coordination in software development,” in *Computer Supported Cooperative Work (CSCW)*, 1999, vol. 8, pp. 175–205.

[9] M. Cataldo and J. D. Herbsleb, “Coordination breakdowns and their impact on development productivity and software failures coordination breakdowns and their impact on development productivity and software failures,” *IEEE Trans. Softw. Eng.*, vol. 39, no. 3, pp. 343–360, 2013.

[10] A. Nguyen-duc, D. S. Cruzes, and R. Conradi, “The impact of global dispersion on coordination , team performance and software quality – a systematic literature review,” *Inf. Softw. Technol.*, vol. 57, pp. 277–294, 2015.

[11] R. Fitzpatrick, “Software quality : definitions and strategic issues software quality : definitions and strategic issues,” pp. 0–34, 1996.

[12] D. Wallace and L. Reeker, “Software quality,” *Guide. to Softw. Eng. Body Knowl.*, 2001, p. 165.

[13] L. S. Nekkanti, *Impact of Coordination Challenges on Quality of Global Software Development Projects*. Blekinge Institute of Technology, Karlskrona, Sweden, 2016.

[14] R. E. Al-qutaish, “Quality models in software engineering literature : an analytical and comparative study,” *J. Am. Sci.*, vol. 6, no. 3, pp. 166–175, 2010.

[15] B. Kitchenhama, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, “Systematic literature reviews in software engineering – a systematic literature review,” *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, 2009.

[16] B. Kitchenham, “Procedures for performing systematic reviews,” 2004.

[17] P. Berander, L. Damm, J. Eriksson, T. Gorschek, K. Henningsson, P. Jönsson, S. Kågström, D. Milicic, F. Mårtensson, K. Rönkkö, P. Tomaszewski, L. Lundberg, M. Mattsson, and C. Wohlin, “Software quality attributes and trade-offs,” *Blekinge Inst. Technol.*, pp. 100, 2005.

[18] C. P. Team, “CMMI ® for Development , Version 1 . 2,” *Carnegie Mellon University*, 2006.

[19] R. E. Al-Qutaish and K. Al-Sarayreh, “Software process and product ISO standards: a comprehensive survey,” *Eur. J. Sci. Res.*, vol. 19, no. 2, pp. 289–303, 2008.

[20] S. Misra and L. Fernández-sanz, “Quality issues in global software development,” in *ICSEA 2011 : The Sixth International Conference on Software Engineering Advances*, 2011, pp. 325–330.

[21] Y. Wang, “Language matters,” in *2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2015, pp. 58–67.

[22] B. Wong and Sazzad Hasan, “Cultural influences and differences in software process improvement programs,” in *Proceedings of the 6th International Workshop on Software Quality*, 2008, pp. 3–10.

[23] R. Prikladnicki, J. Luis, N. Audy, and R. Evaristo, “Global software development in practice lessons learned,” *Softw. Process Improv. Pract.*, vol. 8, no. 4, pp. 267–281, 2004.

[24] S. Matthiesen, P. Bjørn, and L. M. Petersen, “Figure out how to code with the hands of others: recognizing cultural blind spots in global software development,” in *Proceedings of the 17th ACM Conference. on Computer Supported Cooperative Work & Social Computing*, 2014, pp. 1107–1119.

[25] B. C. Bird, N. Nagappan, P. Devanbu, H. Gall, and B. Murphy, “Does distributed development affect software quality? an empirical case study of Windows Vista,” *Commun. ACM*, vol. 52, no. 2, pp. 85–93, 2009.

[26] A. Tariq and A. A. Khan, “Framework supporting team and project activities in Global Software Development (GSD),” in *Emerging Technologies (ICET), 2012 International Conference on*, 2012, pp. 1–6.

[27] J. D. Herbsleb, A. Mockus, T. A. Finholt, R. E. Grinter, B. Labs, and L. Technologies, “Distance, dependencies, and delay in a global collaboration,” in *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, 2000, pp. 319–328.

[28] S. Mohan and J. Fernandez, “Distributed software development projects : work breakdown approaches to overcome key coordination challenges,” in *Proceedings of the 3rd India Software Engineering Conference*, 2010, pp. 173–182.

[29] C. Casado-lumbreras, R. Colomo-palacios, N. Francisca, and S. Misra, “Software development outsourcing : challenges and opportunities in Nigeria,” *J. Glob. Inf. Technol. Manag.*, vol. 17, no. 4, pp. 267–282, 2014.

[30] B. Yost, M. Coblenz, B. Myers, J. Sunshine, J. Aldrich, S. Weber, M. Patron, M. Heeren, S. Krueger, and M. Pfaff, “Software development practices , barriers in the field and the relationship to software quality,” in *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2016, p. 16.

[31] A. Begel, “Effecting change : coordination in large-scale software development,” in *Proceedings of the 2008 International Workshop on Cooperative and Human Aspects of Software Engineering*, 2008, pp. 17–20.

[32] J. A. Espinosa and E. Carmel, “The impact of time separation on coordination in global software teams : a conceptual foundation,” *Softw. Process Improv. Pract.*, vol. 8, no. 4, pp. 249–266, 2003.

[33] R. Prikladnicki and E. Carmel, “Is time-zone proximity an advantage for software development ? the case of the Brazilian IT industry,” in *Software Engineering (ICSE), 2013 35th International Conference on*, 2013, pp. 973–981.

[34] A. Nguyen-duc and D. S. Cruzes, “Coordination of software development teams across organizational boundary – an exploratory study,” in *Global Software Engineering (ICGSE), 2013 IEEE 8th International Conference on*, 2013, pp. 216–225.

[35] J. A. Espinosa, S. a Slaughter, R. E. Kraut, and J. D. Herbsleb, “Team knowledge and coordination in geographically distributed software development,” *J. Manag. Inf. Syst.*, vol. 24, no. 1, pp. 135–169, 2007.

[36] J. D. Herbsleb, D. J. Paulish, and M. Bass, “Global software development at siemens : experience from nine projects,” in *Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference on*, 2005, pp. 524–533.

[37] N. Ramasubbu, M. Cataldo, R. K. Balan, and J. D. Herbsleb, “Configuring global software teams : a multi-company analysis of project productivity, quality , and profits,” in *Proceedings of the 33rd International Conference on Software Engineering*, 2011, pp. 261–270.

[38] N. Nagappan, B. Murphy, and V. R. Basili, “The Influence of organizational structure on software quality: an empirical case study,” in *Software Engineering, 2008. ICSE’08. ACM/IEEE 30th International Conference on*, 2008, pp. 521–530.

[39] C. Ebert and P. De Neve, “Surviving global software development,” *IEEE Softw.*, vol. 18, no. 2, pp. 62–69, 2001.

[40] B. Wong and S. Hasan, “Cultural influences and differences in software process improvement programs,” in *Proceedings of the 6th International Workshop on Software Quality*, 2008, pp. 3–10.

[41] J. A. Espinosa and E. Carmel, “The impact of time separation on coordination in global software teams : a conceptual foundation,” *Softw. Process Improv. Pract.*, vol. 8, no. 4, pp. 249–266, 2003.