Evaluation Model to Assess the Effectiveness of Coordination Processes in Global Software Development Projects: A Roadmap

Anusuyah Subbarao[1,2] and Mohd Naz'ri Mahrin[1]
[1]*Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia.*
[2]*Multimedia University, Cyberjaya, Selangor, Malaysia.*
anusya_r@yahoo.com

***Abstract*—Research shows that software organizations are facing many challenges related to coordinate issues by adopting Global Software Development (GSD) approach. Coordination is a primary mechanism used in between collocated and distributed software development teams in GSD environment. A lack of coordination in GSD can decrease the productivity, complicate the process and delay the completion of tasks. Effective coordination is a crucial aspect in successful software projects. In order to coordinate the processes effectively, it need to be assessed. Research shows that there were less studies on assessing the effectiveness of the coordination processes. Hence, this study intends to identify the coordination processes, coordination strategies, indicators related to the identified coordination processes and coordination strategies used in GSD-base software development organization. This paper presents the roadmap to formulate the evaluation model for GSD coordination processes; made up of indicators for every coordination processes components. In general, project managers can utilise this model as it will serve as a guideline to assess the coordination processes effectively between collocated and distributed team in GSD environment.**

***Index Terms*—Assess; Coordination Processes; Coordination Strategies; Global Software Development.**

I. INTRODUCTION

Globalization is fast turning into a prevalent trend in this present age and causing remarkable changes to take place within software development industries throughout the enterprise world. When a software is being distributed across the countries, this strategy is called Global Software Development [1]. Many software organizations are shifting their strategies towards GSD approach [2-3] due to many benefits such as access to large pool of competent developers, less time taken for software development, reduce software development cost, less time taken to market the software product and to produce better quality software [4-7].

Through implementing the GSD strategy, software organizations are decreasing their costs by substituting expensive collocated workforce with distributed resources. Some software organizations are replacing 65% of their collocated resources with distributed resources to cut down the development cost [8]. Despite its benefits and promises, a number of challenges has hindered the growth of GSD. These challenges have emerged from various factors, spanning from economical, technical, political and even cultural dimensions [7] due to contrasts in time zones, languages and geographical locations [9].

In reaction to these challenges, GSD projects are facing difficulties in communicating and coordinating the projects as these projects are geographically distributed [10, 7]. Darja Smite [27]. claims that coordination in distributed environment remains as a great challenge and it is not being very widely explored Research done by Nguyen et al. [11] shows that studies on team coordination in GSD is lacking and the geographical distribution has impacted the coordination in GSD environment. Poor coordination between the collocated and distributed team is effecting the scope of the contract in GSD projects stated by Khan [23].

It is often presumed that a well-coordinated development will not only produce software faster, but is also expected to collectively produce software of higher quality and at lower cost [32]. Therefore, in order to achieve successful software projects, effective coordination remains a crucial issue [31]. In comparison, projects that had better coordination effectiveness performed much better and achieved greater performance than those projects that lacked coordination [12]. Thus, proper coordination effectiveness is crucial in determining the software project successfulness.

The beginning of this paper discusses the research in relation to coordination effectiveness in software development projects and GSD projects. Section III then continues by highlighting well-ordered guidelines on the formulation of Evaluation Model to assess the Effectiveness of Coordination Processes in Global Software Development Projects. Section IV follows by providing the initial results of this research. The conclusion and future work of this research is included in the final section of this paper.

II. RELATED WORK

A software development project need to be well coordinated in order to produce software of higher quality and at lower cost [32]. A critical factor in successful software projects is effective coordination [31]. Compared to projects that has poor performance, projects that had better coordination effectiveness also generated better performance [12]. In determining the software project to be successful, appropriate coordination effectiveness is crucial.

According to Zhang and Galletta [22], coordination effectiveness alludes to the degree in which dependencies or reliance among task activities are very much overseen and well accomplished Hence, coordination effectiveness can be distinctively evaluated in software development teams using three main facets which are technical, temporal, and process [14]. Technical aspects in terms of checking whether all software parts are linked without any errors, temporal aspects in terms of all the software development is completed according to the schedule and process aspects in terms of fixed or accepted guidelines and priorities that are clearly agreed upon and followed [13-18].

According to Malone and Crowston [19], coordination strategy is a specific arrangement of organized activities to oversee the dependencies, while in coordination theory the actions are known as coordination mechanism. There are three types of coordination mechanisms namely mechanistic, organic and cognitive [13-15]. Mechanistic coordination mechanisms are identified as the most effective in managing routine aspects of tasks and dependencies with directed and proper plans, procedures, programs or different practices such as schedules, user guidelines and manuals [20].

Organic coordination mechanisms are most suitable in situations where routines alter or when tasks have few or completely absent routine aspects. They manage dependencies through communication such as giving feedback and mutual adjustment [20]. Mechanistic and organic coordination mechanisms are explicit coordination mechanisms that involves focused and practical execution. Cognitive coordination is accomplished implicitly when collaborators have knowledge about each other and about each other's tasks because it helps them make a forecast or prediction on what others are probably to do without having to communicate with each other [13]. For example, the knowledge that architects may have about a business user's or IT staff's work or the common grounding resulting from mutual understanding of key terminology can help achieve higher levels of coordination effectiveness.

Thus, in a situation where a particular action is required to support teams in directing the dependencies, the action is distinctly defined as a coordination mechanism. For example, uncomplicated matters in a person's normal existence such as monthly salaries can be regarded as coordination mechanisms that assist us to control and direct our reliance or dependencies with other expenses such as groceries, paying loans, paying utility bills and others respectively. Consequently, those mechanisms (or processes) explicitly employed by a team to help manage task dependencies can be defined as explicit coordination mechanisms (or processes). Explicit coordination mechanisms and processes have been studied in the classical organizational research literature for several years.

Teams need to decide which combination of coordination mechanisms should be applied in order to achieve a desired degree of coordination effectiveness. The evidence from empirical studies have shown that effectiveness of coordination mechanisms can be significantly vary due to different situational factors [13, 17] such as certain attributes of tasks (e.g., routineness), of teams (e.g., size, longevity, geographical, temporal, socio-cultural distances, experience), technology (e.g., available ICT, richness) or of organizations (e.g., organizational culture, power distribution) [13, 17]. Therefore, teams have to skillfully modify a combination or mix of coordination mechanisms that would fit into the given situational factors to achieve desirable coordination effectiveness.

According to Espinosa, et. al [16], work that is completed according to the schedule and within the cost meets the customer requirement, this indirectly shows that all the three aspects namely technical aspect, temporal aspect and process aspect of coordination effectiveness is playing a role to ensure the product meets the customer requirement. J. A. Espinosa et al. [16] also highlighted that team performance is one of the vital element in the software development teams. Though, this does not take place all the time as a high level of coordination effectiveness does not necessarily lead to better performance of the team. There are two other perspectives that need to be considered in determining the coordination effectiveness which are other antecedents influencing performance and several dependencies among the task activities that could bring larger influence on team performance compared to others [13].

According to Li and Maedche [18], "coordination effectiveness has greater predictive power on team performance in agile GSD compared to conventional". According to Chang and Shen [12], successful and well performing projects had better coordination effectiveness compared to projects that had poor performance ratings. Yuan, Zhang, Chen, Vogel, and Chu [21] emphasized that assessing the coordination effectiveness via technical aspect does not give any impact in conventional software development but it gives an impact in global software development.

In summary, appropriate coordination effectiveness is an essential element in GSD projects. It need to be assessed. Furthermore, no general framework, system, model or methodology is currently available to assess the effectiveness of coordination processes in GSD ventures or projects.

III. Formulation

The formulation of Evaluation Model to assess the Effectiveness of Coordination Processes in Global Software Development Projects is basically our initial idea of our research. Our aim is to present this entire roadmap to gain feedback of our model formulation. This model encompasses of three important phases based on our research questions which are Phase 1: Identification of Coordination Process, Coordination Strategies and related Indicators in GSD, Phase 2: Formulation of Evaluation Model to assess the Coordination Effectiveness in GSD and Phase 3: To evaluate the effectiveness of the proposed model. Each phase carries activities. Figure 1 indicates the series of steps and its coordinating activity towards the model formulation. Each phase and its activities are shown in Figure 1 and explained in detail as below.

A. Phase 1: Identification of Coordination Process, Coordination Strategies and related Indicators in GSD Projects

Phase 1 consist of 2 main activities namely Systematic Review and Semi structured interview. Activity 1 which is Systematic Review (SR) is well-known and highly established method analyzing the current study in the software engineering field. Kitchenham [24] stated that "SR is an action of evaluation and interpretation of all accessible causes that is related to the specific study request". As such, the goal of SR is to primarily mete out an assessment of research extent by consuming constant, demanding and auditable procedure.

In order for this SR to be conducted, the software engineering procedures proposed by Kitchenham [25] for Systematic Literature Review is used. Researchers usually plan to select the SR approach as it is a very systematic method and is conducted by following the steps of well-established guidelines. There are three main phases in conducting SR. Each phase of SR involves various tasks and each task is performed based on the research area. There are 3 core SR tasks consisting of: Review Planning, Conducting the Review, and Documenting the Review. Figure 1 displays the activities that are performed in the SR. Generally, the resulting output based on the SR would provide details of coordination processes, coordination strategies and indicators related to each of the identified coordination processes and strategies in GSD.

Activity 2 starts with conducting semi structured interviews sessions. The rationale of selecting semi structured interview is to establish the list of indicators that can be utilize to assess the coordination processes in GSD projects. The target population are mainly project managers who are involved in GSD projects and several of them will be identified as respondents to participate for this interview sessions. To assure that all research directions are explored appropriately, a semi-structured interview guideline comprising of open-ended questions will be conducted amongst the participants. Moreover, telephone interviews will also be held for the participants from diverse countries such as Norway, India, Malaysia, United States of America, Vietnam while face-to-face interviews will be conducted for participants from Malaysia.

Each session of the interview is intended to be between 1 to 3 hours. The recorded audio and the written data from interviews will be collected, organized, recorded and analyzed accordingly. The output will be as same as the activity 1.

The results from Activity 1 and Activity 2 will be the input for the next phase which is phase 2.

B. Phase 2: Formulation of Evaluation Model to assess the Coordination Effectiveness in GSD Projects

Phase 2 consist of 4 main activities. Activity 3 starts with integration of Systematic Review (SR) output and Semi Structured Interview output together. Here Grounded Theory will be used. Grounded theory “is a detailed grounding by systematically" and intensively "analyzing data, often sentence by sentence, or phrase by phrase of the field note, interview, or other document; by 'constant comparison,' data are extensively collected and coded," using the operations touched on in the previous section, thus producing a well-constructed theory [26]. The grounded theory approach is selected by researchers since it is able to considerably produce a significant means of analyzed data from data that have been collected from multiple sources [26]. In software engineering field, grounded theory is one of the well-established method to analyze qualitative data.

Constant comparison and memoing method will be used to finalize the output. Constant Comparison is a process of constantly comparing occurrence of data that labelled in a category with other same category to see they are fit and workable or not [43]. The output of this activity will be a finalized list of coordination processes, coordination strategies and indicators related to each of the identified coordination processes and strategies in GSD.

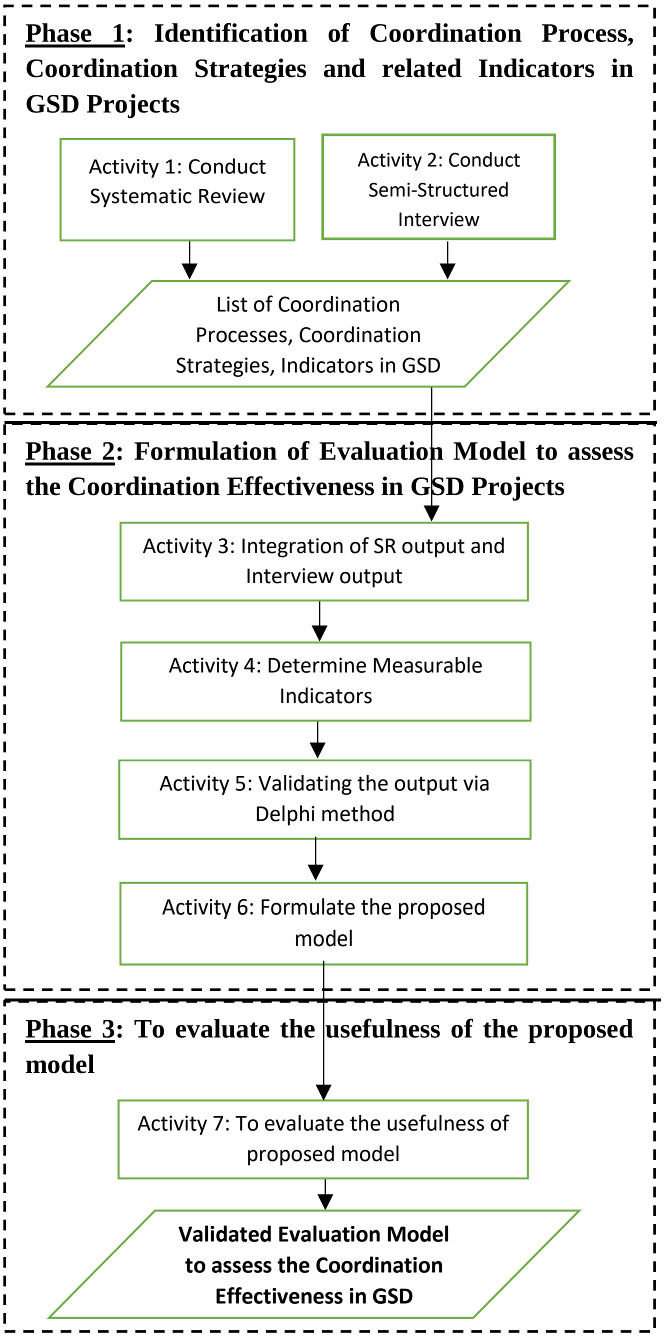

Figure 1: Formulation Process of Evaluation Model to assess the Coordination Effectiveness in GSD Projects

Next is Activity 4 which is to determine measurable indicators. There are two types of indicators namely objective indicators and subjective indicators. Only objective indicators are measurable, therefore it can be used to assess the effectiveness where by subjective indicators are non-measurable. In this activity, only objective indicators will be narrowed down.

This is followed by Activity 5 which is Delphi method. The procedures followed by researchers for implementing the Delphi approach is outlined by Schmidt (1997). Basically, it would serve the dual purpose of having experts provide their feedback and opinions as well as ranking them according to their significance [28]. Mainly, the purpose of the Delphi research method is to acquire the experts’ utmost consistent consensus concerning specific issues and due to its reliability and usability, it has been applied in numerous fields.

Last activity in this phase is Activity 6 namely to formulate the proposed model. The proposed model will consist of list of indicators which will be used to assess the coordination effectiveness. The list of indicators belongs to identified coordination processes and coordination strategies.

C. *Phase 3: To evaluate the usefulness of the proposed model*

In order to evaluate the usefulness of the proposed model, the concluding activity in this phase which is Activity 7 is performed by using case study. The rationale behind the selection of this approach is due to its suitability in investigating an existing phenomenon in a real-life situation [29] as it could offer a central theme or subject of understanding the phenomenon from a various point of perspectives. Also, in qualitative methods, case study is commonly used in software engineering and particularly by researchers to develop and test new theory in the area of global software development [30].

IV. Initial Results

As for now, the research progress is in phase 1. 50% of the data is already been collected by following Systematic Review (Activity 1). The main output of the activity 1 is a list of coordination processes, a list of coordination strategies for the identified coordination processes and a list of indicators of the identified coordination processes or coordination strategies. All these output is achieved by following each and every step of Systematic Review thoroughly. Each steps are followed one by one to retrieve this. Given below are the sample output of our research.

For example, list of coordination processes in GSD that are identified are bridging [33], managing vendor and client relationship [34], team management [35], cultural differences [36] and others.

The identified coordination strategies are Training [35], Tool Selection [35], Team Cognition [37] and Team Motivation [37 and 38], these are for team management [35] (coordination process). Another example is outsourcing relationship management [39], Technology [40], Staff Turnover [38], these are from managing vendor and client relationship [38] (coordination process).

Example of extracted list of indicators for task allocation (coordination process) are number of multi-site requests [41], number of multi-site modification requests which packages had to be done before other packages [41], number of core members per location [42] and others.

More findings are shown in the appendix given. The findings in the appendix is divided into three columns namely coordination processes, coordination strategies for the coordination processes and the last column indicates indicators related to the coordination strategies in global software development environment.

V. Conclusion

This research indicates the well-ordered formulation process of Evaluation Model to assess the Effectiveness of Coordination Processes in Global Software Development Projects in detail and a partial part of our initial findings. This proposed model will be used to assess the effectiveness of the coordination processes in GSD ventures or projects. Basically, in order to assess the coordination effectiveness in GSD projects worldwide, these identified indicators can be executed in dashboard systems that operates in any open source environment.

Future research will include the development of the proposed model. Moreover, an empirically validated research on this model can also be carried out in future.

Appendix

Table 1
Details Findings

Coordination Process (CP)	Coordination Strategy (CS)	Indicators	Paper ID
Team Setup	Team Members Selection	Professional Skills	44,38
		Technical Ability/Knowledge	44,38
		gender	44
		Area/Domain of expertise	42,38
		Capable of working with others to solve any problems(Social capital)	38
		Experience	38
		Pride	38
		Trust	38
		Corporate spirit	38
		competent and committed developers	38
	Team Structure	Flexible Communication Structure	37
		Clarifying Work Structure	37
		Using Boundary Spanning Roles	37
		Forming Virtual Communities	37
		Number of source-code files dependencies	45
		Dependencies a task has with another – Hours that a worker is supposed to spend in a task	45
		Dependencies a task has with another – Expertise a person has about a task	45
Team Development	Team Performance	team size	16,42
		project length	16
		team members' average number of years with the company	16
		project resources	16
		project priority	16
		role description	42
		role distribution	42
		task uncertainty	16
		task type	16
		team experience	16
		on-time completion of the project	16
		on-budget completion of the project,	16
		user participation	16
		team member satisfaction among other things	16
	NA	allocated task matches the capacities of that location	42
		number of dependencies between remote members escalate task complexity	42

Coordination Process (CP)	Coordination Strategy (CS)	Indicators	Paper ID
Team Management	Training	Teamwork	35
		Soft skills	35
		Team role distribution	35
		Team emotion information	35
	Tool Selection	collaborative tools to support the team	46
	Team Cognition	Fostering Transactive Memory (TM)	37
		Identifying gaps and Verifying Understanding	37
		Improving Team Qualification and Expertise	37
	Team Motivation	Temporary collocation	37
		Incentives motivations	37
		Budget for travels between sites	37
		Cost of Virtual communication	37
	NA	time needed to prepare and launch the teams	46
		delays in submission of deliverables	46

ACKNOWLEDGMENT

We are grateful to Universiti Teknologi Malaysia, Advance Informatics School (AIS) and other individuals for their kind support. This work is supported by UTM grant 16H64.

REFERENCES

[1] R. Jain, and U. Suman, "A systematic literature review on global software development life cycle," *ACM SIGSOFT Software Engineering Notes*, vol. 40, no. 2, pp. 1-14, 2015.

[2] W. Aspray, F. Mayadas, and M. Y. Vardi. *Globalization and Offshoring of Software: A Report of the ACM Job Migration Task Force*. New York, USA: ACM, 2006.

[3] D. Sˇmite, C. Wohlin, R. Feldt, and T. Gorschek, T. "Reporting empirical research in global software engineering: A classification scheme," in *Proceedings of the 3rd International Conference on Global Software Engineering*, 2008, pp. 173–181.

[4] P. Kaur, and S. Sharma, "Agile Software Development in Global Software Applications," *International Journal of Computer Applications*, vol. 97, no. 4, pp. 39-43, 2014.

[5] P. J. Ågerfalk, B. Fitzgerald, H. H. Olsson, and E. Ó. Conchúir, "Benefits of global software development: The known and unknown," in *Proceedings of International Conference on Software Process*, 2008, pp. 1-9.

[6] P. J. Ågerfalk, and B. Fitzgerald, "Flexible and distributed software processes: Old petunias in new bowls?" *Communication of the ACM*, vol. 49, no. 10, pp. 26-34, 2006.

[7] J. D. Herbsleb, and D. Moitra, "Global software development," *IEEE Software*, vol. 18, no. 2, pp.16-20, 2001.

[8] A. L. Chua, and S. L. Pan, "Knowledge transfer and organizational learning in is offshore sourcing," *Omega*, vol. 36, no. 2, pp. 267–281, 2008.

[9] D. Damian and D. Moitra, "Guest editors' introduction: Global software development: how far have we come?," *IEEE Software*, vol. 23, no. 5, 2006, pp. 17–19.

[10] E. Ó Conchúir, P. J. Ågerfalk, H. H. Olsson, and B. Fitzgerald, "Global software development: Where are the benefits?," *Communication of the ACM*, vol. 52, no. 8, pp.127-131, 2009.

[11] 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," *Information and Software Technology*, vol. 57, pp. 277-294, 2015.

[12] A. S. Chang, and F. Y. Shen, "Effectiveness of coordination methods in construction projects," *Journal of Management in Engineering*, vol. 30, no. 3, pp. 771-778, 2013.

[13] A. Espinosa, R. Kraut, S. Slaughter, J. Lerch, J. Herbsleb, and A. Mockus, "Shared mental models, familiarity, and coordination: A multi-method study of distributed software teams," in *ICIS Proceedings*, 2002, pp. 1-22.

[14] A. Espinosa, F. J. Lerch, R. E. Kraut, E. Salas, and S. M. Fiore, "Explicit vs. implicit coordination mechanisms and task dependencies: One size does not fit all," in *Team cognition: Understanding the factors that drive process and performance*," E. Salas and S. M. Fiore, Eds. Washington, DC: American Psychological Association, 2004, pp.107-129.

[15] J. A. Espinosa, F. Armour, and W. F. Boh, "Coordination in enterprise architecting: An interview study," in *2010 43rd Hawaii International Conference on System Sciences*, 2010, pp. 1-10.

[16] J. A. Espinosa, J. N. Cummings, and C. Pickering, "Time separation, coordination, and performance in technical teams," *IEEE Transactions on Engineering Management*, vol. 59, no.1, pp. 91-103, 2012.

[17] J. A. Espinosa, and C. Pickering, "The effect of time separation on coordination processes and outcomes: A case study," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, 2006, pp. 25b-25b.

[18] Y. Li, and A. Maedche, "Formulating effective coordination strategies in agile global software development teams," in *Proceedings of the International Conference on Information Systems (ICIS)*, 2012, pp. 1-6.

[19] T. W. Malone, and K. Crowston, "The interdisciplinary study of coordination," *ACM Computing Surveys (CSUR)*, vol. 26, no. 1, pp. 87-119, 1994.

[20] A. H. Van de Ven, A. L. Delbecq, and R. Koenig Jr, "Determinants of coordination modes within organizations," in *American Sociological Review*, vol. 41, no. 2, pp. 322-338, 1976.

[21] M. Yuan, X. Zhang, Z. Chen, D. R. Vogel, and X. Chu, "Antecedents of coordination effectiveness of software developer dyads from interacting teams: an empirical investigation," *IEEE Transactions on Engineering Management*, vol. 56, no. 3, pp. 494-507, 2009.

[22] P. Zhang, and D. F. Galletta, Human-Computer Interaction and Management Information Systems: Foundations. Routledge, 2006.

[23] A. W. Khan, and S. U. Khan, "Critical challenges in execution of offshore software outsourcing contract from vendors' perspective: A systematic literature review," in *2014 5th International Conference on Information and Communication Systems (ICICS)*, 2014, pp. 1-6.

[24] B. Kitchenham, D. Budgen, and P. Brereton, *Evidence-Based Software Engineering and Systematic Reviews*. CRC Press, Taylor & Francis Group, 2016.

[25] B. Kitchenham, "Guidelines for performing systematic literature reviews in software engineering (version 2.3)," Software Engineering Group, School of Computer Science and Mathemathics, Keele University and Department of Computer Science, University of Durham, 2007. Available at https://userpages.uni-koblenz.de/~laemmel/esecourse/slides/slr.pdf

[26] A. Strauss, and J. Corbin, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, Inc, 1998.

[27] D. Šmite, "A case study: Coordination practices in global software development," *in Product Focused Software Process Improvement*, F. Bomarius, and S. Komi-Sirviö, Eds. Berlin, Heidelberg: Springer, 2005, pp. 234-244.

[28] R.C. Schmidt, "Managing Delphi surveys using nonparametric statistical techniques," *Decision Sciences*, vol. 28, no.3, pp. 763–774, 1997.

[29] R. K. Yin, *Case Study Research: Design And Methods*. Sage, 2009.

[30] I. Benbasat, D. K. Goldstein, and M. Mead, "The case research strategy in studies of information systems," *MIS Quarterly*, vol. 11, no. 3, pp. 369, 1987.

[31] D. E. Strode, B. G. Hope, S. L. Huff, and S. Link, "Coordination effectiveness in an agile software development context," in *PACIS 2011 Proceedings*, 2011.

[32] J. A. Espinosa, S. A. Slaughter, R. E. Kraut, and J. D. Herbsleb, "Team knowledge and coordination in geographically distributed software development," *Journal of management information systems*, vol. 24, no. 1, pp. 135-169, 2007.

[33] T. Anand, C. Reddy, and V.S. Mani, "Managing customer involvement in globally distributed agile projects," in *2016 IEEE 11th International Conference on Global Software Engineering Workshops (ICGSEW)*, 2016, pp. 7-12.

[34] Y. Y. Wibisono, R. Govindaraju, I. Sudirman, and D. Irianto, "System testing optimization in a globally distributed software engineering team," in *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*, 2015, pp. 99-103.

[35] R. Vivian, H. Tarmazdi, K. Falkner, N. Falkner, and C. Szabo, "The capabilities of offshore information technology vendor," in *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, 2015, pp. 93-97.

[36] M. J. Monasor, J. Parkes, J. Noll, A. Vizcaíno, M. Piattini, and S. Beecham, "The development of a dashboard tool for visualising online

teamwork discussions," in *ICSE '15 Proceedings of the 37th International Conference on Software Engineering*, 2015, pp. 380-388.

[37] M. Zahedi, M., Shahin, and M. A. Babar, "A systematic review of knowledge sharing challenges and practices in global software development," *International Journal of Information Management*, vol. 36, no. 6, pp. 995-1019, 2016.

[38] N. B. Moe, D. Šmite, G. K. Hanssen, and H. Barney, "From offshore outsourcing to insourcing and partnerships: four failed outsourcing attempts," *Empirical Software Engineering*, vol. 19, no. 5, pp. 1225-1258, 2014.

[39] S. Ali, and S. U. Khan, "Software outsourcing partnership model: An evaluation framework for vendor organizations," *Journal of systems and software*, vol. 117, pp. 402-425, 2016.

[40] W. Hussain, and T. Clear, "Spreadsheets as collaborative technologies in global requirements change management," in *2014 IEEE 9th International Conference on Global Software Engineering*, 2014, pp. 74-83.

[41] A. Lamersdorf, J. Münch, and D. Rombach, "Towards a multi-criteria development distribution model: An analysis of existing task distribution approaches," in *2008 IEEE International Conference on Global Software Engineering*, 2008, pp. 109-118.

[42] C. Manteli, B. Van den Hooff, and H. Van Vliet, "The effect of governance on global software development: An empirical research in transactive memory systems," *Information and Software Technology*, vol. 56, no. 10, pp. 1309-1321, 2014.

[43] H. Boeije, "A Purposeful Approach to the Constant Comparative Method in the Analysis of Qualitative Interviews," *Quality & Quantity*, vol. 36, pp. 391–409, 2002.

[44] S. Deshpande, I. Richardson, V. Casey, and S. Beecham, "Culture in global software development - A weakness or strength?" in *2010 5th IEEE International Conference on Global Software Engineering*, 2010, pp. 67-76.

[45] J. Portillo-Rodríguez, A. Vizcaíno, M. Piattini, and S. Beecham, "Using agents to manage socio-technical congruence in a global software engineering project," *Information Sciences*, vol. 264, pp. 230-259, 2014.

[46] J. G. Guzmán, J. S. Ramos, A. A. Seco, and A. S. Esteban, "How to get mature global virtual teams: A framework to improve team process management in distributed software teams," *Software Quality Journal*, vol. 18, no. 4, pp.409-435, 2010.