Risk Generating Situations of Requirement Engineering in Global Software Development

Huma Hayat Khan, Mohd. Naz'ri bin Mahrin, Suriayati bt Chuprat
Advanced Informatics School
Universiti Teknologi Malaysia (UTM)
Kuala Lumpur, Malaysia
hkhuma2@live.utm.my, mdnazrim@utm.my , suria@ic.utm.my

ABSTRACT

Challenges of Requirements Engineering (RE) become adequate when it is performed in global software development (GSD) paradigm. "Risks" are the important elements for making RE more challenging. For having its in depth understanding, first there is a need to identify those factors which may generate risks during RE in GSD. Therefore a systematic literature review is done. The initial list leads to progressive enhancement for assisting in RE activities in GSD paradigm. This work is especially useful for the, less experience people working in GSD.

KEYWORDS

Systematic literature review (SLR), software Requirement engineering (RE), global software development (GSD), Risks

1 INTRODUCTION

The emergence of globalization concepts has impacted almost every industry, both in positive and negative ways. The word globalization takes into account with multi cultural stakeholders on a single platform. Software industry is also influenced by this globalization by allowing multicultural stakeholders to work together in global platform recognized as GSD environment [1].

RE in GSD paradigm is one of the interesting research topics as described by cheng et al. [2]. This research rise for globalization in software industry is due to the number of advantages it has comparing with the traditional software development process. The advantages include; round the clock development, hiring workforce at low cost, maximum chance to the access the highly qualified global pool etc.

The GSD paradigm describes the fact of under going changes to many RE activities as the participants are not collocated. The new paradigm of GSD increases the risks of project failure irrespective of its huge number of advantages. To cope up this issue, RE pitfalls due to GSD should be overcome. These pitfalls are mostly due to the differences of culture, languages, knowledge, times zone etc which vary among software development organizations in GSD paradigm. These changing situation factors are the main source of software failure which is specifically influential in RE process as discussed by [3, 4].

The goal of this paper is to identify and enlist the factors and the risks generated by these factors during RE in GSD paradigm. The work compiles the changing situation factors which should be taken care with to minimize the risks, which may lead to project failure. The identification is done from the existing literature by performing systematic literature review (SLR) [5]. In order to have unique identification of each of the risk generating factors, Grounded theory's [6] data coding, constant comparison and memoing techniques are adopted.

The rest of the paper is structured as follows. Section II describes the background of the study, section III illustrate the methodology of the study, section IV comprises of results, section V consist of discussion and section VI is the conclusion of the study.

2 BACKGROUND

Wiegers et al. [7] described requirement as a statement which relates to customer needs, objectives, capability or a condition that must be possessed by the product to satisfy and give value to a stakeholder. So we can say that requirement is something which a system must have or satisfy or perform which is being identified by the client side. Now coming towards RE, it is essential to notify that generally, RE is concerned with understanding about what are the things which system must do (the 'what').

A definition by Zave [8] states "Requirements Engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems". Sommerville & Sawyer [9] argue about RE as "the activities that cover discovering, analyzing, documenting and maintaining a set of requirements for a system". By supporting the definitions described above Wiegers [7] recommend that RE cover all the software project lifecycle activities related to the understanding of not only capabilities but also the attributes of a system. Similarly in the same year Deb Jacobs argues about the importance of RE and says "the cost of incorrect, misunderstood, and not agreed upon requirements affects all of us in terms of time, money, and lost opportunities" [10].

Few years back researchers Fowler [11] argued that "Everything else in software development depends on the requirements. If you cannot get stable requirements, you cannot get a predictable plan". Carmel [12] argues that defining and acquiring the software needs for the new system is challenging and it is one of the crucial phases of software development as discussed by Darke [13]. Davis [4] Anthony [14] explains that it is crucial because it has a direct impact on success and failure of any software. Software requirement specification argued by Greenspan [15] is the outcome document of RE phase consisting of specified requirements. Continuing to the previous era researchers, there are some more researchers who in 21st century says that this RE phase is difficult and crucial enough when it is done in co-located environment as described by Damian [16] and, it is further argued by the researchers Damian [16, 17], Espinosa [18], MacGregor [19] that it becomes even more difficult and challenging when different stakeholders, sitting in different geographies having distant cultures, time zones etc. specify requirements.

RE becomes more complicated in globally distributed development paradigm, due to fact of having multiple stakeholders with varying backgrounds, for example. Having requirements common understanding is already a difficult or complex task to takes place within one organization in co located environment, but it becomes even more complex or harder when the stakeholders are having varying tacit knowledge, different time zones as it makes communication much harder. Platform of GSD further complicates RE due to social and cultural aspects related with not only gathering but also managing requirements [20].

The more risk occurrences in GSD paradigm may be the main contributor in the RE process complexity. Now first of all there is a need to identify the risk generating factors, which ultimately may influence the RE in GSD. These risk generating factors may be related to culture, social aspects, technologies, humans etc [21, 22, 23], which may results in project failure. Researchers also gave careful consideration to the aspect, that these risk generating factors also leads to changing situations among the organizations working is GSD paradigm.

It is evident from the literature, that the researchers have not only focused their work to the criticality of RE in GSD [21, 22, 23], but they have also describe the importance of control of changing situations among software development organizations in order to have successful project. By linking the facts, we come to know the importance of changing situations while performing RE in GSD. However, to the best of our knowledge, there is a lack of study which describes the list of factors propagating risks during RE in GSD. Therefore, this study focuses on identification and listing risks with that of factors which may acts as a source of these risks while performing RE in GSD paradigm.

3 RESEARCH METHODOLOGY

In our effort to review, we follow the method described in [5]. We have decomposed the

research in three parts: *Review planning, Review conduction* and *Results reporting.*

3.1 Review planning:

Review planning deals with the selection of papers for review. Research goals and research questions are identified in this phase. Besides this, keywords with sources, queries, inclusion/exclusion criteria are also identified here. *Research goal and research questions:* The goal of this literature review is to identify the factors which may generate risks during RE in GSD paradigm. *RQ1:* What are the factors which may generate risks during RE in GSD paradigm? *Identifying the keywords:* Base upon the research question stated above, we go for keywords which in fact facilitate us in queries construction with that of the selection of relevant papers from the datasets: Software Requirement engineering (RE) risks, distributed requirement engineering risks, software risks factors, distributed requirement engineering risk factors. *Identifying the sources:* The databases we considered for the search are: ACM Digital Library, Emerald, IEEE, Springer-Link, Science Direct, Wiley online and JSTOR. *Identifying the queries:* The following abstract query for the research questions is used: '((software "requirement engineering") AND ("Risks")) AND (“distributed software development” OR “global software development”)'. Due to the different search interface of each of the search engines, the query is divided into sub queries as some of them does not accept long query. *Identifying the inclusion/exclusion criteria* we have three levels for inclusion and exclusion. First we excluded all those papers which are either table of contents or some information about the full proceedings of conference and workshops etc. The second level is associated with checking papers on basis of keywords. So if the paper does not has any of the keywords (“requirement” OR “requirement engineering”) AND (“risk” OR “risk factors”), then that particular paper is excluded from the dataset. All the papers which must consist of keywords (“requirement” OR “requirement engineering”) AND (“risks” OR “risk factors”) with the other one as mentioned above are included in the dataset. The third level of exclusion is on basis of repetition; like if a paper is repeated, then it is included only once.

3.2 Review Conduction:

Step 1 consists of building the initial data sets and filtration of papers based on inclusion and exclusion criteria. To accomplish this step, we selected seven data sources to get the papers related to our study by using automated query.

Building the initial dataset: The papers are collected on basis of keywords and queries. From ACM we got (113) papers, similarly from Emerald (43) papers, IEEE (67) papers, JSTOR (5) papers, Willey online (67) papers, Science Direct (89) papers and Springer Link (90) papers. The papers are filtered on basis of exclusion and inclusion criteria as described above. Total papers which we found to be included in our research are 172 but after performing filtrations on basis of type, keywords and repletion, the papers covered are 105.

4 RESULTS

4.1 Result Reporting:

This stage of our research comprises of steps from step2 to step6, where we report all the results of our study with that of the comments from the experts related to the factors identified.

Step2 deal with the extraction of data units from the most related filtered. These data units are actually the identified situational factors, which may generate risks while performing RE in GSD paradigm. These identified factors which are large in numbers have to go through a filtration process. This time the filtration is done on basis of conceptual and explicit duplications.

Step3 consist of steps to remove the duplication (explicit or conceptual). In order to do this, Grounded theory’s [6] data coding constant comparison and memoing techniques are adopted. Data coding not only deals with getting raw data but also converting it to a conceptual level as argued by Corbin [24]. In this research, a number of data codes from the multiple data sources are compared via constant comparison for getting the main factors by removing the duplications. Glaser argues about complexity of data coding with constant comparison and says that in order to deal with this complexity researchers argue that the memoing process can be a solution [25]. Hence this step consists of scanning the data to find the duplication instances. As there is a direct mapping

among the individual data units that are joined or combined in this comparison, so memoing is not required as it shows the thoughts that influenced the joining or combination of data units. Where while combining the data units, careful steps are taken to maintain the source information. So, if two different data units are having same textual depiction or explanation and meaning, they are combined into a single unit – but it is also to be making sure that both sources of data units are apparent and clear in the combined data unit testimony in the main table. Now following the identification and removal of clear duplication among the data units, conceptual duplications in data units are identified. We combine two data units which are not having same textual tags but are having same meaning. In order to do this, memoing is also done which records the thought process behind this. Besides this, as previously done with the removal of clear duplication stage, combined data units sources links are maintained.

Step4 comprises of factors labeling and initial classification. Here each of the data units (factors) is given a label. These labeling are on basis of factors relatedness to the specific area they are concern with. Memos are used to record the justification for these labeling. Similarly actions are taken to come up with initial classification for data units (factors). This time also, memos draft the motivation or justification that has initiated or created the initial classification. Under this initial classification, factors are present with their appropriate labeling. Similar to the previous steps, memos are used here to record the justification for classification and labeling.

Step5 comprises of constant comparison for evaluating the precision of the factor labels and classifications. The labels are renamed when suitable data units are moved to alternating factors or to completely novel factors – as believed suitable in regard of the rising factors and classification. Similarly some of the classifications are renamed or combined or decomposed as considered appropriate. For this step similar to the previous one, memos are used to draft the thought process with that of the historical trace of the actions. This assists in envisioning the factors and classification sources, and allows an assessment of the impact of each data sources on the main list as it come forward.

Step6 comprises of expert reviews. Once the initial list of situational factors at its final form is created, then it is forwarded to the experts from academia to evaluate the initial list of Situational Factors which may generate risks during RE process in GSD paradigm. The experts are selected from academia on basis of their experience in RE field. More specifically two experts are selected having more than five years of experience in field of RE. The tasks provided to them is to evaluate the list for its comprehensiveness, as well as to review if the factors are grouped under right classification with that of any recommended modification both at factors and classification level.

The experts recommended some modification in the initial list of situational factors which may generate risks during RE process in GSD. Factors *Interaction tools* and *Interaction medium or technology* are combined and given the name of *"Interaction medium, technology and tools"*, similarly factor *national culture* is combined with *cultural background, social climate* is combined with *social background*, *organization policies and strategies* besides with the *organization person retention strategy* and *organization structure and boundaries* are grouped into single factor named as "*organization structure, policy and strategy"*, *technical expertise* is moved from the classification "*tools, technologies, techniques and standards"* to *"stakeholders"*. *Partner power* and *position in organization* which are previously considered to be same factors are now grouped separately under classification *"stakeholders"*. Under the same *"stakeholders"* classification, *client involvement* is combined with the factor *client commitment*. The recommendations from the experts are included in the list as shown in *table 1*.

In total there are 8 classifications and 74 factors. "*Communication and distance"* has following factors: Interaction skills, Interaction styles, Interaction medium, technology and tools, Interaction infrastructure and Distance. "*Differences in culture, background, language, organization and time"* has following factors: Language, Cultural background, Work

environment, View points, Time zones, Social background, Organizational culture, Political difference, Time shifts, Inter group culture, Organization structure policy and strategy. "*Knowledge management and awareness"* has following factors: Team awareness, Data repository, Domain knowledge level, Knowledge management techniques and procedures, Business knowledge, Access management, Tacit knowledge level, Requirement engineering practice knowledge, configuration management, Knowledge management awareness, Requirement management. "*Management"* has following factors: Coordination skills, Coordination techniques, Competence, Decisions, Supervisor sub-ordinate relationship, Management strategy, Cooperation approach. "*Tools, Technologies, Techniques and Standard Selection"* has following factors: Technique selection, Standard selection, Tools selection, Technology selection. "*Stakeholders*" has following factors: Team members competence and experience with in application, Team members motivation level, Team members familiarity with each other, Team members preferences related to project, Team knowledge exchange ability, Team members background, Personnel trust, Leadership skills, Personnel/group relationship, Team members decision capability, Stakeholders utility values, Team members level of receiving help with heavy work load, Stakeholder priority to situation urgency, Team members knowledge level, Team members international work experience, Team member relation to the project (trust, commitment.), Partner power, Client availability, Client commitment, Members carriers prospects position in organization, Stakeholder background, Partner interpretation skills, Team management capabilities, Stakeholders common work experience, Technical expertise. *"Project and Process"* has following factors: Project phase distribution, Requirements engineering process, Economic process, Process maturity, Collaboration process, Management process. *"Requirements"* has following factors: Requirement specification document format, Requirement engineering effort, Requirement representation style, Requirement interpretation. These classifications, factors besides with the risks which these factors may generate are shown in Table attached in Appendix.

5 DISCUSSIONS

In this section, contribution, future work and limitations of our study is discussed in detail.

5.1Utilizing Situational Factors:

This work provides direction towards a comprehensive list of factors. The identified situational factors which may generate risks during RE process in GSD paradigm can be an important reference for researchers. The initial list can also act as a guide for the researchers and practitioners working in situational RE, to consider and control these factors in order to overcome the risks. Besides with the initial list of factors, this work also contributes in listing the risks against each of the situational factor, hence we also come to have a repository of risks which may be generated by the situational factors during RE process in GSD paradigm.

5.2 Future Work:

The initial list of factors identified is from the literature; hence the list does not contain the data from the industry. So in future a more comprehensive list can be generated covering both the literature and industry. This work also shows the directions towards the future work of impact analysis of these factors on RE in GSD paradigm, hence extending the investigation to a more comprehensive level.

5.3 Limitation:

This work comprises results from state of knowledge only, hence does not cover the state of practice. We make every effort to cover all the related papers discuss risks for RE process in GSD, but still it is possible that we may miss any published work. Similarly the paper is forwarded to other researchers in order to deal with biasness about the search protocol used, but still biasness aspect can not be ignored as well. The threat of misinterpretation can also not be ignored, as it is one of the must factor in every literature review, although we tried our best to overcome these aspect by dealing it carefully.

6 CONCLUSION

The goal of this paper is to identify and enlist the factors and the risks generated by them during RE

process in GSD paradigm. The work compiles the changing situations factors which should be taken care with, to minimize the risks related to various aspects which, may lead to project failure.

In our effort to review we have decomposed the research in three parts: *Review planning, Review conduction* and *Results reporting*. The initial list of situational factors is reported in the paper consisting of 74 factors grouped under 8 classifications. This initial list is an important input to the comprehensive list of situational factors, which is our future work. The initial list can also act as a guide for the researchers and practitioners working in situational RE, to consider and control these factors in order to overcome the risks with changing situations faced by them while performing RE process in GSD paradigm. Besides with the initial list of factors, this work also contributes in listing the risks against each of the situational factor, hence we also come to have a repository of risks which may be generated by the situational factors during RE process in GSD paradigm.

AKNOWLEDGMENT

"The authors would like to thank 'The Ministry of Higher Education (MOHE)' and Research Management Centre, Universiti Teknologi Malaysia for supporting this research work via Grant Number: **07J53**"

7 REFERENCES

1 Fryer, K. and M. Gothe. Global software development and delivery: Trends and challenges. 2008.

2 Betty H.C. Cheng, J.M.A., *Research directions in requirements engineering. In Future of Software Engineering*, in *FOSE '07*. 2007. 285-303.

3 Loucopoulos, P. and V. Karakostas, *System Requirements Engineering*. 1995, New York.

4 Davis, A.M., *Software Requirements: Analysis and Specification*. 1989: Prentice Hall.

5 Kitchenham, B. and S. Charters, *Guidelines for performing Systematic Literature Reviews in Software Engineering*. 2007.

6 Glaser, B.G., *Doing Grounded Theory: Issues & Discussion* 1998: Sociology Pr.

7 Wiegers, K., *Process Impact*. 2007. <http://www.processimpact.com>.

8 Zave, P., *Classification of research efforts in requirements engineering.* ACM Comput. Surv., 1997. **29**(4): 315-321.

9 Sommerville, I. and P. Sawyer, *Requirements Engineering: A Good Practice Guide*. 1997: John Wiley & Sons, Inc. 404.

10 Jacobs, D., *Requirements Engineering so Things Don't Get Ugly*, in *Companion to the proceedings of the 29th International Conference on Software Engineering*. 2007, IEEE Computer Society. 159-160.

11 Fowler, M., *The New Methodology.* 2005. <http://www.martinfowler.com>.

12 Carmel, E., *Global software teams: collaborating across borders and time zones*. 1999 NJ, USA Prentice Hall PTR Upper Saddle River.

13 Darke, P. and G. Shanks, *User viewpoint modelling: understanding and representing user viewpoints during requirements definition.* Information Systems Journal, 1997. **7**(3): 213-219.

14 Terry Anthony Byrd, K.L.C., Robert W. Zmud, *A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques.* MIS Quarterly, 1992. Vol. 16(No. 1): 117-138.

15 Greenspan, S., J. Mylopoulos, and A. Borgida, *On formal requirements modeling languages: RML revisited*, in *Proceedings of the 16th international conference on Software engineering*. 1994, IEEE Computer Society Press: Sorrento, Italy. 135-147.

16 E. Damian, D. and D. Zowghi, *RE challenges in multi-site software development organisations.* Requirements Engineering, 2003. **8**(3): 149-160.

17 Damian, D.E. and D. Zowghi, *The Impact of Stakeholders? Geographical Distribution on Managing Requirements in a Multi-Site Organization*, in *Proceedings of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering*. 2002, IEEE Computer Society. 319-330.

18 J. Alberto Espinosa , E.C., John Wiley, *The impact of time separation on coordination in global software teams: a conceptual foundation*, in *software process improvement and practice*. 2005.

19 MacGregor, E., Y. Hsieh, and P. Kruchten, *Cultural patterns in software process mishaps: incidents in global projects.* SIGSOFT Softw. Eng. Notes, 2005. **30**(4): 1-5.

20 Hanisch, J. and B. Corbitt. *Requirements engineering during global software development: Some impediments to the requirements engineering process – a case study*. in *Proceedings of the Twelfth European Conference on Information Systems*. 2004.

21 Ågerfalk, P.J. and J. Ralyté, *Situational Requirements Engineering Processes: reflecting on method engineering and requirements practice.* Software Process: Improvement and Practice, 2006. **11**(5): 447-450.

22 Jo Hanisch, T.T., Brian Corbitt, *Exploring the cultural and social impacts on the requirements engineering processes - highlighting some problems challenging virtual team relationships with clients.* Journal of Systems and Information Technology, 2001. **Vol. 5**(Iss: 2):1 - 20.

23 Lopez, A., J. Nicolas, and A. Toval. *Risks and Safeguards for the Requirements Engineering Process in Global Software* in *Global Software Development. 2009. Fourth IEEEEngineering, 2009. ICGSE International Conference on*. 2009.

24 Corbin, J.M. and A.L. Strauss, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. 2008: Sage Publications, Incorporated.

25 Bryant, A. and K. Charmaz, *The Sage Handbook of Grounded Theory*. 2010:Sage Publications.

26 Robinson, W.N., S.D. Pawlowski, and V. Volkov, *Requirements interaction management.* ACM Comput. Surv., 2003. **35**(2): 132-190.

27 Lopez, A., J. Nicolas, and A. Toval. *Risks and Safeguards for the Requirements Engineering Process in Global Software Development*. in *Global Software Engineering,2009. ICGSE 2009. Fourth IEEE International Conference on*. 2009.

28 Ramesh, B., L. Cao, and R. Baskerville, *Agile requirements engineering practices and challenges: an empirical study.* Information Systems Journal, 2010. **20**(5): 449-480.

29 Parviainen, P. and M. Tihinen, *Knowledge-related challenges and solutions in GSD.* Expert Systems, 2011.

30 Komi-Sirviö, S. and M. Tihinen, *Lessons learned by participants of distributed software development.* Knowledge and Process Management, 2005. **12**(2): 108-122.

31 Puhl, S. and R. Fahney. *How to assign cost to avoidable requirements creep: A step towards the waterfall's agilization*. in *Requirements Engineering Conference (RE), 2011 19th IEEE International*. 2011.

32 Omoronyia, I., Ferguson, J., Roper, M., Wood, M., *A review of awareness in distributed collaborative software engineering.* Software: Practice and Experience, 2010. **40**(12): 1107-1133.

33 Ramzan, M., Batool, A., Minhas, N., Qayyum, Z., Jaffar, M., *Automated Requirements Elicitation for Global Software Development (GSD) Environment Software Engineering, Business Continuity, and Education*, T.-h. Kim, et al., Editors. 2011, Springer Berlin Heidelberg. 180-189.

34 Perini, A., Qureshi, N., Sabatucci, L., Siena, A., Susi, A., *Evolving Requirements in Socio-Technical Systems: Concepts and Practice Conceptual Modeling – ER 2011*, M. Jeusfeld, L. Delcambre, and T.-W. Ling, Editors. 2011, Springer Berlin / Heidelberg. 440-447.

35 Skroch, O., *A method to evaluate the suitability of requirements specifications for offshore projects Developing Business Application Systems*. 2010, Gabler Verlag. 59-78.

36 Calefato, F., D. Damian, and F. Lanubile, *Computer-mediated communication to support distributed requirements elicitations and negotiations tasks.* Empirical Software Engineering, 2012. **17**(6): 640-674.

37 Šmite, D. and J. Borzovs, *A Framework for Overcoming Supplier Related Threats in Global Projects Software Process Improvement*, I. Richardson, P. Runeson, and R. Messnarz, Editors. 2006, Springer Berlin / Heidelberg. 50-61.

38 Lamersdorf, A. and J. Münch, *A multi-criteria distribution model for global software development projects.* Journal of the Brazilian Computer Society, 2010. **16**(2): 97-115.

39 Boden, A., B. Nett, and V. Wulf, *Articulation work in small-scale offshore software development projects*, in *Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*. 2008, ACM: Leipzig, Germany. 21-24.

40 Dorairaj, S., J. Noble, and P. Malik, *Effective Communication in Distributed Agile Software Development Teams Agile Processes in Software Engineering and Extreme Programming*, A. Sillitti, et al., Editors. 2011, Springer Berlin Heidelberg. 102-116.

41 Narayanaswamy, R. and R.M. Henry, *Effects of culture on control mechanisms in offshore outsourced IT projects*, in *Proceedings of the 2005 ACM SIGMIS CPR conference on Computer personnel research*. 2005, ACM: Atlanta, Georgia, USA.139-145.

42 Ning, A., Hou, H., Hua, Q., Yu, B., Hao, K., *Requirements Engineering Processes Improvement:A Systematic View*, in *SPW 2005*. 2005: Berlin Heidelberg. 151 – 163.

43 Lings, B., Lundell, B., Ågerfalk, P., Fitzgerald, B., *Ten Strategies for Successful Distributed Development The Transfer and Diffusion of Information Technology for Organizational Resilience*, et al., Donnellan, B. Editors. 2006, Springer Boston. 119-137.

44 Moe, N. and D. Šmite, *Understanding Lacking Trust in Global Software Teams: A Multi-case Study Product-Focused Software Process Improvement*, J. Münch and P. Abrahamsson, Editors. 2007, Springer Berlin / Heidelberg. 20-34.

45 Šmite, D., *A Case Study: Coordination Practices in Global Software DevelopmentProduct Focused Software Process Improvement*, F. Bomarius and S. Komi-Sirviö, Editors. 2005, Springer Berlin / Heidelberg. 25-46.

46 Herbsleb, J.D., Mockus, A., Finholt, T. A., Grinter, R. E., *An empirical study of global software development: distance and speed*, in *Proceedings of the 23rd International Conference on Software Engineering*. 2001, IEEE Computer Society: Toronto, Ontario, Canada. 81-90.

47 Luna-Reyes, L.F., Black, L. J., Cresswell, A. M., Pardo, T.A., *Knowledge sharing and trust in collaborative requirements analysis.* System Dynamics Review, 2008. **24**(3): 265-297.

48 Lamersdorf, A. and J. Münch, *Model-Based Task Allocation in Distributed Software Development Software Engineering Approaches for Offshore and Outsourced Development*, M. Nordio, et al., Editors. 2010, Springer Berlin Heidelberg. 37-53.

49 Jiménez, M. and M. Piattini, *Problems and Solutions in Distributed Software Development: A Systematic Review Software Engineering Approaches for Offshore and Outsourced Development*, K. Berkling, et al., Editors. 2009, Springer Berlin Heidelberg. 107-125.

50 Siffat, U.K., Niazi, M., Ahmad, R., *Critical Success Factors for Offshore Software Development Outsourcing Vendors: An Empirical Study Product-Focused Software Process Improvement*, M. Ali Babar, M. Vierimaa, and M. Oivo, Editors. 2010, Springer Berlin / Heidelberg. 146-160.

51 Sengupta, B., S. Chandra, and V. Sinha, *A research agenda for distributed software development*, in *Proceedings of the 28th international conference on Software engineering*. 2006, ACM: Shanghai, China. 731-740.

52 Mikulovic, V. and M. Heiss, *"How do I know what I have to do?": the role of the inquiry culture in requirements communication for distributed software development projects*, in *Proceedings of the 28th international conference on Software engineering*. 2006, ACM: Shanghai, China. 921-925.

53 Valentine, C. and R. Ita. *Project Management within Virtual Software Teams*. in *Global Software Engineering, 2006. ICGSE '06. International Conference on*. 2006.

54 Hossain, E., et al. *Risk Identification and Mitigation Processes for Using Scrum in Global Software Development: A Conceptual Framework*. in *Software*

Engineering Conference, 2009. APSEC '09. Asia-Pacific. 2009.

55 Orlikowski, W.J., *Knowing in Practice: Enacting a Collective Capability in Distributed Organizing.* Organization Science, 2002. **13**(3): 249-273.

56 Calefato, F., F. Lanubile, and C. Ebert, *Practice: Collaborative Development Environments*, in *Global Software and IT*. 2011, John Wiley & Sons, Inc.109-123.

57 Musio, I., *IBM Industry Practice: Challenges in Offshore Software Development from a Global Delivery Center Software Engineering Approaches for Offshore and Outsourced Development*, O. Gotel, M. Joseph, and B. Meyer, Editors. 2009, Springer Berlin Heidelberg. 4-13.

58 Krishnan, P. and C. Ranganathan, *Boundary spanning in offshored ISD projects: a project social capital perspective*, in *Proceedings of the special interest group on management information system's 47th annual conference on Computer personnel research*. 2009, ACM: Limerick, Ireland. 225-230.

59 Lamersdorf, A., et al. *A Rule-Based Model for Customized Risk Identification in Distributed Software Development Projects*. in *2010 5th IEEE International Conference on Global Software Engineering (ICGSE),* 2010.

60 Khoshroo, B.M. and H. Rashidi. *Distributed development ; A literature analysis on challenges and solutions*. in *GCC Conference & Exhibition, 2009 5th IEEE*. 2009.

61 Šmite, D., *Global software development projects in one of the biggest companies in Latvia: is geographical distribution a problem?* Software Process: Improvement and Practice, 2006. **11**(1): 61-76.

62 Casey, V. and I. Richardson, *Implementation of Global Software Development: a structured approach.* Software Process: Improvement and Practice, 2009. **14**(5): 247-262.

63 Jensen, R.E. and P. Bjørn, *Divergence and Convergence in Global Software Development: Cultural Complexities as Social Worlds From Research to Practice in the Design of Cooperative Systems: Results and Open Challenges*, J. Dugdale, et al., Editors. 2012, Springer London. 123-136.

64 Berlizev, A. and N. Guelfi, *Correct analysis for embedded system modeling: an outcome of east-west scientific cooperation*, in *Proceedings of the 2008 international workshop on Software Engineering in east and south europe*. 2008, ACM: Leipzig, Germany. 23-30.

65 Dai, C.-f. and M.-l. Wang, *Distributed requirement elicitation and negotiation based on the hall for workshop of meta-synthetic engineering*, in *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*. 2009, ACM: Seoul, Korea.1452-1456.

66 Al-Ani, B., et al., *Trust and surprise in distributed teams: towards an understanding of expectations and adaptations*, in *Proceedings of the 4th international conference on Intercultural Collaboration*. 2012, ACM: Bengaluru, India. 97-106.

67 Dalberg, V., et al., *Cross-cultural collaboration in ICT procurement*, in *Proceedings of the 2006 international workshop on Global software development for the practitioner*. 2006, ACM: Shanghai, China. 51-57.

68 Sudhakar, G.P., A. Farooq, and S. Patnaik, *Soft factors affecting the performance of software development teams.* Team Performance Management, 2011. **Vol. 17**(Iss: 3): 187 - 205.

69 Miller, J.A., R. Ferrari, and N.H. Madhavji, *An exploratory study of architectural effects on requirements decisions.* Journal of Systems and Software, 2010. **83**(12): 2441-2455.

70 Gorschek, T. and A.M. Davis, *Requirements engineering: In search of the dependent variables.* Information and Software Technology, 2008. **50**(1–2): 67-75.

71 Šmite, D., N. Moe, and R. Torkar, *Pitfalls in Remote Team Coordination: Lessons Learned from a Case Study Product-Focused Software Process Improvement*, A. Jedlitschka and O. Salo, Editors. 2008, Springer Berlin / Heidelberg. 345-359.

72 Gutwin, C., R. Penner, and K. Schneider, *Group awareness in distributed software development*, in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. 2004, ACM: Chicago, Illinois, USA. 72-81.

73 Berenbach, B., *Impact of organizational structure on distributed requirements engineering processes: lessons learned*, in *Proceedings of the 2006 international workshop on Global software development for the practitioner*. 2006, ACM: Shanghai, China. 15-19.

74 Sakthivel, S., *Managing risk in offshore systems development.* Commun. ACM, 2007. **50**(4): 69-75.

75 John, M., et al., *Human and social factors of software engineering: workshop summary.* SIGSOFT Softw. Eng. Notes, 2005. **30**(4): 1-6.

76 Hossain, E., et al. *Risk Identification and Mitigation Processes for Using Scrum in Global Software Development: A Conceptual Framework*. in *Software Engineering Conference, 2009. APSEC '09. Asia-Pacific*. 2009.

77 Šmite, D., et al., *Empirical evidence in global software engineering: a systematic review.* Empirical Software Engineering, 2010. **15**(1): 91-118.

78 Hossain, E., M.A. Babar, and J. Verner, *How Can Agile Practices Minimize Global Software Development Co-ordination Risks? Software Process Improvement*, R.V. O'Connor, et al., Editors. 2009, Springer Berlin Heidelberg. 81-92.

79 Siakas, K.V. and B. Balstrup, *Software outsourcing quality achieved by global virtual collaboration.* Software Process: Improvement and Practice, 2006. **11**(3): 319-328.

80 Richardson, I., et al., *A Process Framework for Global Software Engineering Teams.* Information and Software Technology, 2012. **54**(11): 1175-1191.

81 Ramachandran, S., S. Dodda, and L. Santapoor, *Overcoming Social Issues in Requirements Engineering.* CCSIT, 2011. **3**(133): 310–324.

82 Mäkiö, J., S. Betz, and A. Oberweis, *OUTSHORE Maturity Model: Assistance for Software Offshore Outsourcing Decisions Collaborative Software Engineering*, I. Mistrík, et al., Editors. 2010, Springer Berlin Heidelberg. 329-341.

83 Narayanaswamy, R. and R.M. Henry, *Effects of culture on control mechanisms in offshore outsourced IT projects*, in *Proceedings of the 2005 ACM SIGMIS CPR conference on Computer personnel research*. 2005, ACM: Atlanta, Georgia, USA. 139-145.

84 Lin, L., S.J. Prowell, and J.H. Poore, *The impact of requirements changes on specifications and state machines.* Software: Practice and Experience, 2009. **39**(6): 573-610.

85 Chang, K.T. and K. Ehrlich, *Out of sight but not out of mind?: Informal networks, communication and media use in global software teams*, in *Proceedings of the 2007 conference of the center for advanced studies on Collaborative research*. 2007, IBM Corp.: Richmond Hill, Ontario, Canada. 86-97.

86 Kwan, I. and D. Damian, *Extending socio-technical congruence with awareness relationships*, in *Proceedings of the 4th international workshop on Social software engineering*. 2011, ACM: Szeged, Hungary. 23-30.

87 Islam, S., et al. *Offshore-outsourced software development risk management model*. in *Computers and Information Technology, 2009. ICCIT '09. 12th International Conference on*. 2009.

88 Betz, S., S. Hickl, and A. Oberweis. *Risk Management in Global Software Development Process Planning*. in *2011 37th EUROMICRO Conference on software Engineering and Advanced Applications (SEAA),* 2011.

89 Anantatmula, V. and M. Thomas, *Managing global projects: A structured approach for better performance.* Project Management Journal, 2010. **41**(2): 60-72.

90 Macke, D. and T. Galinac, *Optimized Software Process for Fault Handling in Global Software Development Making Globally Distributed Software Development a Success Story*, Q. Wang, D. Pfahl, and D. Raffo, Editors. 2008, Springer Berlin / Heidelberg. 395-406.

91 Lin, T., *Investigating Culture in IT Offshoring: A Literature Review Nordic Contributions in IS Research*, H. Salmela and A. Sell, Editors. 2011, Springer Berlin Heidelberg. 124-137.

92 Kauppinen, M., et al., *Implementing requirements engineering processes throughout organizations: success factors and challenges.* Information and Software Technology, 2004. **46**(14): 937-953.

93 Karlsson, L., et al., *Requirements engineering challenges in market-driven software development – An interview study with practitioners.* Information and Software Technology, 2007. **49**(6): 588-604.

94 Karandikar, H. and S. Nidamarthi, *A model for managing the transition to a global engineering network spanning industrialized and emerging economies.* Journal of Manufacturing Technology Management, 2006. **Vol. 17**(Iss:8): 1042 - 1057.

95 Gandhi, S.J., A. Gorod, and B. Sauser, *Prioritization of outsourcing risks from a systemic perspective.* Strategic Outsourcing: An International Journal, 2012. **Vol. 5**(Iss: 1): 39 - 71.

96 Islam, S. and S.H. Houmb. *Integrating risk management activities into requirements engineering*. in *2010 Fourth International Conference on Research Challenges in Information Science (RCIS),*2010.

97 Mistrík, I., et al., *Collaborative Software Engineering: Challenges and Prospects Collaborative Software Engineering*, I. Mistrík, et al., Editors. 2010, Springer Berlin Heidelberg. 389-403.

98 Asnar, Y., et al. *Computer Aided Threat Identification*. in *2011 IEEE 13th Conference on Commerce and Enterprise Computing (CEC),*. 2011.

99 Sadraei, E., et al., *A field study of the requirements engineering practice in Australian software industry.* Requirements Engineering, 2007. **12**(3):145-162.

100 Lee, W.-T., et al., *Change impact analysis with a goal-driven traceability-based approach.* International Journal of Intelligent Systems, 2010. **25**(8): 878-908.

101 Crnkovic, I., *Component-based software engineering — new challenges in software development.* Software Focus, 2001. **2**(4): 127-133.

102 Lin, L., S.J. Prowell, and J.H. Poore, *The impact of requirements changes on specifications and state machines.* Software: Practice and Experience, 2009. **39**(6): 573-610.

103 Rodriguez, F., et al., *Evaluating Collaboration Platforms for Offshore Software Development Scenarios Software Engineering Approaches for Offshore and Outsourced Development*, B. Meyer and M. Joseph, Editors. 2007, Springer Berlin / Heidelberg. 96-108.

104 Lawrence, B., K. Wiegers, and C. Ebert, *The top risk of requirements engineering.* Software, IEEE, 2001. **18**(6): 62-63.

105 Piri, A., T. Niinimäki, and C. Lassenius, *Fear and distrust in global software engineering projects.* Journal of Software: Evolution and Process, 2012. **24**(2): 185-205.

106 Krieg, R., *Impact of structured product definition on market success.* International Journal of Quality & Reliability Management, 2004. **Vol. 21** (Iss: 9): 991 - 1002.

107 Pitts, M.G. and G.J. Browne, *Improving requirements elicitation: an empirical investigation of procedural prompts.* Information Systems Journal, 2007. **17**(1): 89-110.

108 Breaux, T.D., A.I. Antón, and E.H. Spafford, *A distributed requirements management framework for legal compliance and accountability.* Computers & Security, 2009. **28**(1–2): 8-17.

109 Lin, L., S.J. Prowell, and J.H. Poore, *The impact of requirements changes on specifications and state machines.* Software: Practice and Experience, 2009. **39**(6): 573-610.

110 Boehm, B. and R. Valerdi, *Impact of software resource estimation research on practice: a preliminary report on achievements, synergies,and challenges*, in *Proceedings of the 33rd International Conference on Software Engineering*. 2011, ACM: Waikiki, Honolulu, HI, USA.1057-1065.

111 Ernst, N.A., J. Mylopoulos, and Y. Wang, *Requirements Evolution and What (Research) to Do about It Design Requirements Engineering: A Ten-Year Perspective*, K. Lyytinen, et al., Editors. 2009, Springer Berlin Heidelberg. 186-214.

112 Kontio, J., et al. *Managing commitments and risks: challenges in distributed agile development*. in *ICSE 2004 Proceedings. 26th International Conference on Software Engineering, 2004.*

113 Houston, D.X., G.T. Mackulak, and J.S. Collofello, *Stochastic simulation of risk factor potential effects for software development risk management.* Journal of Systems and Software, 2001. **59**(3): 247-257.

114 Ocker, R.J. and J. Fjermestad, *Communication differences in virtual design teams: findings from a multi-method analysis of high and low performing experimental teams.* SIGMIS Database, 2008. **39**(1): 51-67.

115 Hall, R.J. *LSS: A Tool for Large Scale Scenarios*. in *21st IEEE/ACM International Conference on Automated Software Engineering, 2006. ASE '06.* 2006.

116 Liu, X., M. Azmoodeh, and N. Georgalas, *Specification of Non-functional Requirements for Contract Specification in the NGOSS Framework for Quality Management and Product Evaluation*, in *Proceedings of the 5th International Workshop on Software Quality*. 2007, IEEE Computer Society.

117 Staats, B.R., *Unpacking Team Familiarity: The Effects of Geographic Location and Hierarchical Role.* Production and Operations Management, 2012. **21**(3): 619-635.
118 Newell, S., G. David, and D. Chand, *An analysis of trust among globally distributed work teams in an organizational setting.* Knowledge and Process Management, 2007. **14**(3): 158-168.
119 Muethel, M., F. Siebdrat, and M. Hoegl, *When do we really need interpersonal trust in globally dispersed new product development teams?* R&D Management, 2012. **42**(1).
120 MacGregor, E., Y. Hsieh, and P. Kruchten, *Cultural patterns in software process mishaps: incidents in global projects.* SIGSOFT Softw. Eng. Notes, 2005. **30**(4): 1-5.
121 Li, Y., et al., *The role of team problem solving competency in information system development projects.* International Journal of Project Management, 2011. **29**(7): 911-922.
122 Casey, V., *Imparting the importance of culture to global software development.* ACM Inroads, 2011. **1**(3): 51-57.
123 Baiqiang, X. and Z. Deming. *Engineering safety information in software intensive systems*. in *2011 9th International Conference on. Reliability, Maintainability and Safety (ICRMS),* 2011.
124 Ebert, C., *Requirements Engineering*, in *Global Software and IT*. 2011, John Wiley & Sons, Inc. 37-44.
125 Ferreira, S., et al., *Reducing the risk of requirements volatility: findings from an empirical survey.* Journal of Software Maintenance and Evolution: Research and Practice, 2011. **23**(5): 375-393.
126 Yang, H., et al., *Analysing anaphoric ambiguity in natural language requirements.* Requirements Engineering, 2011. **16**(3): 163-189.
127 Ralyte, J., et al. *A framework for supporting management in distributed information systems development*. in *Research Challenges in Second International Conference on Information Science, 2008. RCIS 2008.*
128 da Silva, F.Q.B., et al., *An evidence-based model of distributed software development project management: results from a systematic mapping study.* Journal of Software Maintenance and Evolution: Research and Practice, 2011.

APPENDIX

References	Risks	Situational Factors	Classification
[26] [27] [28] [29]	Lack of privacy, Lack of efficiency, Lack of knowledge sharing, Ineffective information	Interaction skills	

References	Risks	Situational Factors	Classification
[30]	sharing, Lack of efficient coordination and collaboration, Emergence of group issues, Lack of quality individual decision making, Negative relationship building, Loss of cohesion, Lack of requirement existence and stability, Lack of inadequate user development interaction, Lack of synchronization and act, Inaccurate task allocation, Misunderstanding of remote participants action or spoiling relationships, Chaotic and uneven knowledge transfer		
[26] [27] [31] [28] [29] [30] [32] [33] [36] [37] [38] [39] [40]	Lack of privacy, Lack of efficiency, Lack of knowledge sharing, Ineffective information sharing, Lack of efficient coordination and collaboration, Emergence of group issues, Lack of quality individual decision making, Inefficient articulation work, Loss of development speed, Requirements misunderstanding, Outcome failure, Negative relationship building, Loss of cohesion, Lack of requirement existence and stability, Lack of inadequate user development interaction, Inaccurate task allocation	Interaction styles	
41] [42] [43] [44] [45] [46] [27] [28] [29] [30] [38] [37]	Loss of development speed, Lack of trust, Lack of shared team identity, Lack of awareness of members activity, Lack of team members effort coordination, Lack of effective leadership, Lack of effective knowledge sharing, Lack of determination of appropriate task technique, Outcome failure, Delay, Negative relationship building, Loss of cohesion, Lack of requirement existence and stability, Lack of inadequate user development interaction, Lack of synchronization and act, Inaccurate task allocation, Chaotic and uneven knowledge transfer, Lack of uniform software development environment, Lack of work awareness	Interaction medium or technology	***Communication and Distance***
[37] [29] [47] [32]	Lack of individual and shared knowledge understanding, Chaotic and uneven knowledge transfer, Lack of uniform software development environment, Lack of work awareness, Lacking proximity	Interaction tools	
[48] [49] [43] [50] [35] [37]	Effort overhead, High defect frequency, Lack of client involvement, Hidden cost, quality work, product, Lacking proximity	Interaction infrastructure	
[51][52] [53] [54] [55] [29] [30] [56] [36] [57]	Inefficient communication, inadequate communication, Inappropriate knowledge management, Cultural diversity, Time zone difference, Lack of adequate requirement capturing, High rate of integration errors, High rate of organizational differences, Lack of trust, Lack of efficient collaboration process, Lack of effective outcome, Delay, Lack of efficient coordination and communication control, Conflicts over priorities.	Distance	
[52] [46] [58] [59] [60] [53] [27] [61] [62] [38] [63] [50] [45]	Requirement misunderstanding, Lack of quality outcome, Ineffective management practices, Scope creep, Lack of timely project completion, Increase rate of coordination problems, Lack of workflow communication, varying methodologies, Lack of effective communication, Decrease in team productivity, Lack of requirement comprehensiveness, Incorrect reporting from remote team, Verbal contact avoidance, Inaccurate task allocation, Lack of common goals and client involvement	Language	***Differences with respect to Culture, Background, Language, Organization and Time***
[52] [64] [65] [66] [67]	Requirement misunderstanding, Lack of quality outcome, Varying meaning for a situation, Lack of effective communication, Decrease in team productivity, Barriers to work ethics, Incorrect reporting from remote team, Lack of quality decisions.	Cultural background	
[34] [58] [67] [68] [27] [69] [70] [71]	Increase in requirement evolution rate, Lack of quality outcome, Lack of work accuracy, Lack of improvisation skills, Lack of information and artifact sharing, Lack of quality requirement document confusion of remote participant actions,	Work Environment	
[34]	Increase in requirement evolution rate	View point	
[72] [52] [58] [73] [74] [75] [59] [53] [76] [54] [55] [61] [56] [33] [40] [77] [78]	Lack of coordination, Ineffective management practices, Scope creep, Lack of timely project completion, Lack of workflow communication, varying methodologies, Lack of quality requirement document, Lack of efficient requirement reviews and effective communication, Decrease in team productivity,	Time zone	
[67] [58][76][54][55][79]	Lack of quality outcome, Varying meaning for a situation, Lack of effective communication and social interaction, Project mismanagement	Social background	
[58][67] [27] [55] [80] [81] [82]	Lack of quality outcome, Challenging cooperation, Barriers to work ethics, Misunderstanding of remote participant actions, Lack of social interaction, Incorrect reporting from remote team, Lack of interests, Lack of quality decisions	National culture	
[52] [83] [73] [59] [60] [76] [27] [55] [30] [41] [84] [80] [33] [36] [34] [81]	Lack of quality outcome, Challenging cooperation, Ineffective management practices, Scope creep, Lack of timely project completion, Increase rate of coordination problems, Lack of workflow communication, varying methodologies, Lack of efficient requirement reviews, Lack of effective communication, Decrease in team productivity, Inefficient collaboration and communication process, Lack of work accuracy, Lack of improvisation skills, Misunderstanding of remote participant actions, Personal loss, Lack of social interaction, Lack of uniform software development environment, Complex problem escalation	Organizational culture	
[85] [76] [79] [82]	Decrease in team productivity, Inefficient collaboration and communication process, Lack of social interaction, Project mismanagement, Lack of quality decisions	Social climate	
[79] [34]	Project mismanagement Lack of social interaction,	Political difference	
[38]	Inaccurate task allocation	Time shifts	
[67] [83] [27] [34] [33]	Challenging cooperation, Scope creep, Lack of timely project completion, Increase rate of coordination problems, Lack of workflow communication, Lack of effective communication, Barriers to work ethics, Lack of improvisation skills, Lack of social interaction, Lack of information and artifact sharing, Lack of interests, Lack of quality requirement document, Inaccurate task allocation	Inter group culture	
[86] [52] [87] [58] [88] [89] [50] [91]	Lack of effective coordination and collaboration, lack of quality outcome, inadequate client involvement, inefficient process for task completion, lack of team motivation, lack of quality	Organization Structure and	

References	Risks	Situational Factors	Classification
[45]	decisions, partners week relationships, lack of trust, lack of common goals, weak contractual relations.	boundaries	
[92]	Loss of key personnel.	Organization person retention strategy	
[92] [93]	Inadequate client involvement, inefficient requirement engineering process, collecting data without improving requirement engineering process.	Organization policies and strategies	
[85] [72][86] [27] [90]	Non-effective coordination, lack of efficient information seeking, infrequent communication, lack of exchanging information, lack of maintained awareness, lack of trust, lack of knowledge sharing, delay, un-aware of remote team member skills, unaware of changing requirements, unaware of job responsibilities of remote team members, high defects frequency, lack of control.	Team Awareness	***Knowledge management and awareness***
[94] [95]	Data loss, lack of requirement specification quality	Data repository	
[73] [87][59]	Lack of quality outcome, erroneous requirements, budget and schedule overrun, productivity downfall,	Domain knowledge level	
[54]	Lack of frequent communication, lack of competence	Internal knowledge level	
[96][60][88][27][33][97]	Lack of frequent communication, low technical efficiency, lack of competence, lack of quality management, lack of project quality, lack of exchanging information, lack of maintained awareness, lack of trust, lack of knowledge sharing, delay, team ineffectiveness, lack of quality documentation, lack of awareness about development project.	Knowledge management techniques and procedures	
[61]	Delay in problem domain clarification, extra cost for rework	Business knowledge	
[98]	Unintended data editing, un/intentional disclosure for personal gain.	Access management	
[29]	Lack of knowledge sharing, lack of synchronization and act	Tacit knowledge level	
[99]	Lack of accurate requirements fault modeling.	RE practice knowledge	
[32][48]	Effort overhead, work unawareness	Configuration management	
[27]	Lack of quality decisions, lack of control, lack of maintained awareness, , unaware of job responsibilities of remote team members, high defects frequency	Knowledge management awareness	
[28][100][101][102] [103][104]	Lack of requirement stability, intrinsic schedule flaws, high frequency of system failure, overlooking crucial requirements, not understanding the needs behind the requirements, overlooking non functional requirements, not inspecting requirements, reducing the solution domain by representing in design form, insufficient change management, lack of careful requirement handling, project completion failure.	Requirement management	
[52] [97] [77] [27][38] [45][50]	Lack of effective collaboration, unfamiliarity with technology, increase cost, loss of development speed, inaccurate task allocation, unawareness about development project, unsuccessful collaboration, lack of trust, lack of personal contact, lack of team involvement.	Coordination skills	***Management***
[27] [30] [52] [45] [50]	Lack of effective collaboration, unfamiliarity with technology, increase cost, loss of development speed, requirements misunderstanding, lack of coordination, misalignment of tools with expectations, unrealistic estimation, lack of effective traceability, delay, lack of work awareness, collaboration, lack of trust, lack of personal contact, lack of team involvement, hidden cost	Coordination technique	
[27] [32]	Lack of skilled analyst role, lack of efficient team performance, lack of organizational performance, lack of contingency, delay, lack of work awareness, lack of understanding the project scope, lack of quality requirement artifacts, scope creep, lack of quality decisions,	Competence	
[27]	Misalignment of tools with expectations, unrealistic estimation	Decisions	
[105]	Lack of proactive transparency	Supervisor sub-ordinate relationship	
[92]	Increase in requirement uncertainty, lack of understanding of project scope, loss of key personals	Management strategy	
[45][50]	Lack of understanding activities accelerating the knowledge sharing, lack of trust, troublesome disagreements, less social capital.	Cooperation approach	
[108] [107] [106] [93] [40]	Inaccurate estimation, poor quality outcome, lack of quality requirement engineering activities, inadequate customer representation, requirement misunderstanding, , inadequate requirements, lack of propagation of relevant changes to artifacts, scope creep, ineffective communication, lack of shared understanding,	Technique selection	***Tools, Technologies, Techniques and Standards***
[108] [109][110] [48] [31]	Poor quality outcome, Inaccurate estimation, inadequate customer representation, requirement misunderstanding, , inadequate requirements, lack of propagation of relevant changes to artifacts, scope creep, ineffective communication, lack of shared understanding	Technical expertise	
[60] [78]	Low technical efficiency, high frequency of conflicting requirements, lack of standards, lack of shared understanding, lack of effective coordination, lack of group awareness.	Standards	
[93] [30][27] [78] [80] [111] [40]	Lack f requirement engineering quality outcome, lack of integrated tools, lack of access to requirements history, lack of reporting about fulfillment of preconditions, lack of allowing to requirements documents elaboration, lack of requirements remote negotiation and discussion facilitation, lack of uniform software development environment, lack of trust, lack of requirement management, lack of effective communication, requirement misinterpretations	Tools	
[112][30][69][84]	Lack of team work performance, software project failure, lack of uniform software	Technology	

References	Risks	Situational Factors	Classification
[113] [33] [91]	development environment, lack of early architecture quality, greater frequency of requirements uncertainty, lack of information and artifact sharing, lack of decision making quality.		
[115] [67] [114] [113]	Requirements conflicts, requirements misunderstanding, lack of quality requirement representation, wrong expectations, undetected activity errors, lack of team performance, lack of efficient collaboration, lack of quality outcome, mistrust, lack of quality management, wrong team setup and adjustment, inefficient requirement engineering process, inaccurate task allocation, effort overhead, lack of quality decisions.	Competence and experience with in application	***Stakeholders***
[114] [82]	Lack of team performance, activity errors, wrong team set up and adjustment.	Motivation level	
[116] [85] [117]	Inadequate communication, increase in staff problems, lack of team effectiveness, mistrust, lack of work team cohesion, lack of effective collaboration.	Familiarity with each other	
[67] [109]	Inefficient collaboration, poor quality outcome.	Preferences related to project	
[67] [27]	Lack of team effectiveness, inefficient collaboration, ineffective communication, lack of requirement rationale understanding, inefficient requirement engineering process.	Knowledge Exchange ability	
[29]	Lack of shared understanding.	Background	
[67] [41]	Lack of shared understanding, lack of efficient information sharing, lack of efficient collaboration	Personnel trust	
[92]	Inefficient requirement engineering process, lack of efficient information flow tailoring, person becomes bottleneck.	Leadership skills	
[105] [112] [67] [118][119]	Lack of effective collaboration, lack of quality outcome, on stake security, mistrust, weak team cohesion.	Personnel/group relationship	
[92]	Lack of accurate estimation, lack of quality outcome.	Decision capability	
[67]	Lack of project efficiency.	Utility values	
[67] [96] [27]	Lack of team effectiveness, inefficient collaboration, ineffective communication, lack of requirement rationale understanding, inefficient requirement engineering process.	Knowledge level	
[67]	Lack of efficient collaboration.	International work experience	
[67]	Inefficient collaboration, lack of quality outcome, delay.	Relation to the project	
[120]	Lack of satisfied requirements, inefficient communication, and high rate of requirements conflicts.	Power/position in organization	
[27][76] [121]	Lack of client implication, conflicting approaches to requirement engineering process, requirements variability, higher issues with user abilities and concurrences.	Client involvement	
[67]	Loss of competency.	Carriers prospects	
[27]	Client misalignment with project goal, requirement variability.	Client commitment	
[27]	Lack of participation from clients, higher issues with user abilities.	Client availability	
[65]	Lack of efficient requirement elicitation and negotiation	Background	
[92]	Lack of effect group problem solving ability, measurement scale misconception, inappropriate requirement validation, lack of quality outcome, delay.	Partner interpretation	
[80]	High rate of workforce turnover, lack of remote staff information.	Team management capabilities	
[67]	Inaccurate task allocation, mistrust, delay, effort overhead, inefficient collaboration and communication.	Common work experience	
[61]	Problematic overall joint management, problematic responsibility share, extra management needed at each location.	Project phase distribution	***Project and Process***
[59] [104] [122] [28] [70]	Lack of quality outcome, undetected errors, erroneous requirements, budget and schedule overrun, poor communication, not inspecting requirements, attempting to perfect requirements before construction, ignoring non functional requirements	Requirements engineering process	
[123]	Unacceptable results.	Economic process	
[42] [82]	Productivity downfall, inefficient communication, lack of quality requirements documentation, effort overhead,	Process maturity	
[52] [90]	Lack of communication, collaboration and control.	Collaboration process	
[91]	Lack of quality decision making.	Management process	
[96][87] [28][124][125][128] [126]	Overlooking crucial requirements, not understanding the needs behind the requirements, ignoring non functional requirements, reducing solution domain by representing it in design form, insufficient change management, lack of synchronization, inconsistent specification, erroneous requirement, flaws in design requirements.	Requirement specification document format	***Requirements***
[31]	Scope creep, lack of attention to importance concerns, unclear requirements.	RE effort	
[127][28]	Choosing wrong solution for requirement implementation, reducing the solution domain.	Requirement representation style	
[30][126][35] [35]	Effort overhead, inadequate solution, inaccurate requirement capturing and understanding, lack of quality outcome, delay.	Requirement interpretation	