International Journal of Digital Information and Wireless Communications (IJDIWC) 4(1): 63-78
The Society of Digital Information and Wireless Communications, 2014 (ISSN: 2225-658X)

Factors Generating Risks during Requirement Engineering Process in Global Software Development Environment

Huma Hayat Khan, Mohd. Naz'ri bin Mahrin, Suriayati bt Chuprat
Advanced Informatics School,
Universiti Tecknologi Malaysia,
hkhuma2@live.utm.my, mdnazrim@utm.my, suria@ic.utm.my

ABSTRACT

Challenges of Requirements Engineering become adequate when it is performed in global software development paradigm. There can be many reasons behind this challenging nature. "Risks" can be one of them, as there is more risk exposure in global development paradigm. So it is may be one of the main reasons of making Requirement Engineering more challenging. For this first there is a need to identify the factors which actually generate these risks. This paper therefore not only identifies the factors, but also the risks which these factors may generate. A systematic literature review is done for the identification of these factors and the risks which may occur during requirement engineering process in global software development paradigm. The list leads to progressive enhancement for assisting in requirement engineering activities in global software development paradigm. This work is especially useful for the, less experience people working in global software development.
.

KEYWORDS

Systematic literature review (SLR), Requirement engineering (RE), global software development (GSD), Software, Risks, Factors, distributed software development (DSD).

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 global software development environment [1].

Requirement engineering 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 global software development paradigm describes the fact of undergoing 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 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. The information used for this identification is taken from the literature by performing systematic literature review (SLR) [5]. In order to have unique identification of each of the factors generating risks, Grounded theory's [6] constant comparison and memoing steps 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 Requirement Engineering, 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 Requirement Engineering 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 Requirement Engineering 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 requirement engineering 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 requirement engineering phase consisting of specified requirements. Continuing to the previous era researchers, there are some more researchers who in 21st century says that this requirement engineering 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.

Requirement engineering process 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 global software development further complicates requirement engineering due to social and cultural aspects related with not only gathering but also managing requirements [20].

The issue of having more complicated RE process in GSD is due to many reasons. 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 factors which may generate these risks, which ultimately may influence the RE process in GSD. There are numbers of factors which may impact RE process in GSD. These factors may be related to culture, social aspects, technologies etc, as discussed above, which may generate risks for the RE process [21, 22, 23]. These risks may results in project failure and one of the various sources of these risks are the factors such as: technology, culture, human etc leading to changing situation among the software industry working in GSD paradigm. Besides this, authors in their work also identified the sources for requirement engineering risks in global software development environment. They defined them as *communication and* *distance*, which occur due to the reason of dispersion of multiple stakeholders across multiple countries and time zones; *knowledge management and awareness*, which occur due to the difficulties of keeping awareness, knowledge cohesion and knowledge coherence when various working groups concurrently access it; *cultural differences*, risks which are derived where people who have different culture interact with each other; *management and project coordination*, these are the risks derived from the organization establishments, definition of roles and coordination of procedures; *tools which support the processes*, risks that are derived due to the lacking of tools that support the requirement engineering process; and *clients*, these are, risks due to the interaction among clients not present in collocated sites [23].

So from the previous literature evidence we come to know that the researchers have not only focused their work to the criticality of RE process 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 from [3, 21, 22, 23], we come to know the importance of changing situations while performing RE process 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 process 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 process in GSD paradigm.

3 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 requirement engineering process in global software development paradigm.

RQ1: What are the factors which may generate risks during Requirement Engineering process in global software development (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 of 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.
We have divided our study into several steps. *Figure 1* describes the steps taken to identify the factors and the risks generated by those factors.

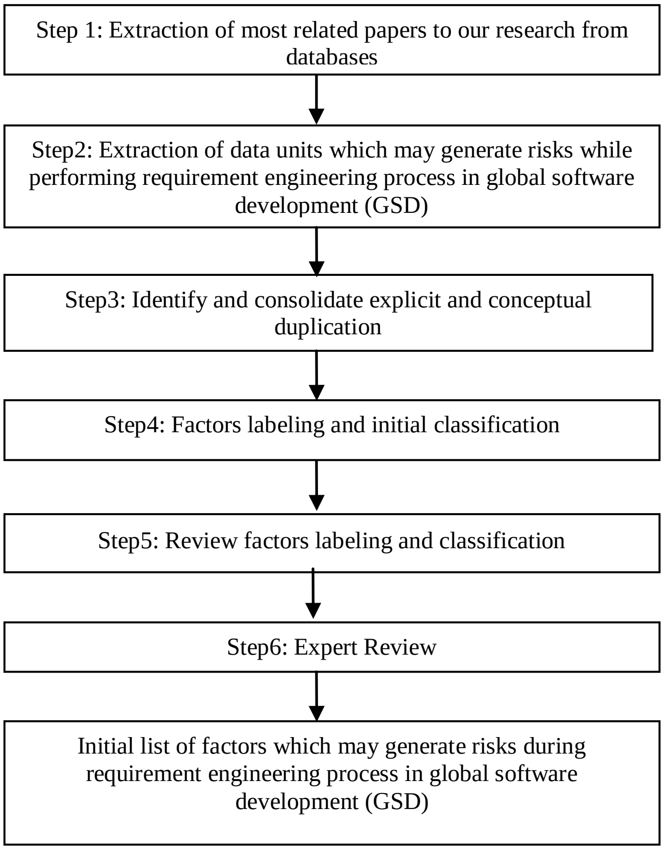

Figure 1. Steps for Generating Initial List of Factors

3.2. Review Conduction:

Step 1 as shown in *figure 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.
Papers filtration based on the inclusion/ exclusion criteria: We perform filtration on initial data set as described above. These filtrations help us to decide whether particular papers should stay in or excluded from the data set. We filtered papers on basis of their type, keywords and repetition. So on basis of "Type" we excluded all the papers which are either table of contents or any definition document or information about full proceedings of conferences or workshops. After that we excluded 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. In the last level of exclusion we excluded all the repeated papers and included them only once.

Table 1. Papers Distribution

Data sources	Total	Covered
ACM	47	24
Emerald	4	4
IEEE	26	16
JSTOR	2	1
Willey Online	29	22
Science Direct	28	10
Springer Link	36	28
Total		105

Table1 shows the statistics of papers. Total papers which we find 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 as shown in *figure 1* 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 requirement engineering process in global software development (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 as shown in *figure 1* consist of steps to remove the duplication (explicit or conceptual). In order to do this, Grounded theory's data coding, constant comparison and memoing techniques are adopted. This theory by Glaser [6] is the methodology for analyzing the data in a systematic. Data coding deals with not only 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 [6] by saying that ''simultaneously many categories and their properties may be emerging at different levels of conceptualization and different ways of being related by theoretical codes''. So 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 as shown in *figure 1* 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 Requirement Engineering field. More specifically two experts are selected having more than five years of experience in field of requirement engineering. 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 recommendation from the experts is included in the list as shown in *table 2*.

Table 2 comprises of four columns: *Classification*, *Situational factors*, *Risks* and *References*. *Classification* and *factors* column contain the grouping of factors under related classification. Similarly the *Risk* column details all the related risks which may be generated by each of the situational factor identified. The last *references* column links the identified list of factors to the literature from where they are taken from.

The initial list of factors which may generate risks as shown above in *table 2* consist of 74 factors grouped under 8 classifications. Each of the classification is named based on the criteria of most frequently used classifications in existing published work. These classifications are further reviewed by the experts for its naming conventions and for the group of factors it contains as described above.

Communication and Distance: comprises of factors; *Interaction skills* (abilities that one can posses for better interaction), *Interaction styles* (formal, informal etc), *Interaction medium* (synchronous or en-synchronous communication mediums), *technology and tools* (web based discussion tools, groupware, search engines etc), *Interaction infrastructure* (copper cable, fiber, or wireless technologies etc) and *Distance* (geographic location).

Cultural, Background, Language, Organizational and Time Differences: comprises of factors; *Language* (language people use for communication), *Cultural background* (one's life experience as shaped by membership in groups based on ethnicity, race, socioeconomic status, gender, exceptionalities, language, religion, sexual orientation, and geographical area), *Work Environment* (the place that one works), *View point* (a technique of composition that provides a vocabulary for thinking about and acting upon movement and gesture), *Time zone* (time differences), *Social background* (how any one is raised), *Organizational culture* (it deals with group behavior, organization values, organization visions, organization norms, organization working language etc), *Political difference* (it deals with the government controls as it is argued that everything is controlled by them. So business cannot be started without visiting them), *Time shifts* (relates to different job timing like evening, morning shift and night shifts), *Inter group culture* (way of dealing, communicating, performing work etc among the team or group), *Organization Structure, policy and strategy* (deals with aspects like task allocation, coordination, task supervision, what is the criteria to choose members, leaders and decisions etc).

Knowledge Management and Awareness: comprises of factors; *Team Awareness* (team members ability to perceive, feel and consciousness etc), *Data repository* (how data is stored and maintained), *Domain knowledge level* (degree of valid knowledge to the specific discipline), *Knowledge management techniques and procedures* (techniques and procedures use to manage knowledge), *Business knowledge* (experience, design and process, files or documents, plans for future activities etc), *Access management* (deals with the access provided for different aspects like data etc), *Tacit knowledge level* (degree of tacit knowledge which is harder to explain but necessary for tasks), *Requirement Engineering practice knowledge* (knowledge related to RE practices), *Configuration management* (task to control and manage changes), *Knowledge management awareness*

(awareness about managing knowledge) and *Requirement management* (deals with ways to manage requirements that is add, delete modify etc)

Management: comprises of factors; *Coordination skills* (deals with ones skill to coordinate with others in the team), *Coordination technique* (techniques which are adopted to have effective coordination like, change of command, effective leadership and supervision etc) *Competence* (deals with ones ability to perform any job or task), *Decisions* (deals with management decisions), *Supervisor sub-ordinate relationship* (it deals with the relationship between the managers or team leads with the one who are working under their supervision e.g. friendly, strict etc), *Management strategy* (strategies related to managing things and work) and *Cooperation approach* (ways to make things balance or in harmony)

Tools, Technologies, Techniques and Standards Selection: comprises of factors*; Technique selection* (it deals with the criteria of technique selection for performing any task), *Standards selection* (it deals with the criteria of standard selection to be considered for task fulfillment), *Tools selection* (it deals with the criteria of tools selection for using in order to complete any task) and *Technology selection* (it deals with the criteria of technology selection for task completion).

Stakeholders: comprises of factors*; Team members competence and experience with in application* (it deals with the abilities and experience of team member related to application they are working on), *Team members motivation level* (how much desirable one to do work), *Team members familiarity with each other* (deals with the concept that whether the team members working together are strangers or they already know each other), *Team members preferences related to project* (it deals with individual attitude towards decision making or evaluation judgment), *Team knowledge exchange ability* (how well the team members involve in the project tasks share or exchange the knowledge among them), *Team members background* (experience, academic qualification, expertise etc), *Personnel trust* (the level of trust among the team members), *Leadership skills* (it deals with the person's ability to lead the people working under him/her), *Personnel/group relationship* (type of relationship among team members like friendly, honesty etc), *Team members decision capability* (ability of team members to take appropriate decisions), *Stakeholders utility values* (it deals with the stakeholder values related to multiple resources and facilities they want from organization they are working with), *Team members level of receiving help with heavy work load* (it deals with the help given to the team members for their heavy work load accomplishment in form of moral support or physical help etc), *Stakeholder priority to situation urgency* (this factor deals with different stakeholders priorities which they give to different situations like it may happen that one conflict or problem is seemed urgent to be resolved but for others it may not be given such priority), *Team members knowledge level* (deals with the knowledge of the team members not specific to the application but related to other aspects also), *Team members international work experience* (whether the team member has some past experience of doing work with people who are from some other countries), *Team member relation to the project* (team members trust and commitment for the project completion), *Partner power* (the authorities, team members from the other organization have and don't have), *Client availability* (deals with the availability of client for discussing and validating things), *Client commitment* (it deals with the client dedication towards project), *Members carriers prospects* (team members job alternatives), *position in organization* (corporate or business title of the team member like manager, executive etc), *Client background* (knowledge, culture, experience, expertise, type (technical or non-technical) etc of the client), *Partner interpretation skills* (deals with abilities to have accurate meaning of things), *Team management capabilities* (deals with the capabilities and abilities related to team management), *Stakeholders common work experience* (have they previously worked together or not) and *Technical expertise* (deals with the stakeholders expertise related to technical aspects.)

Project and Process: comprises of factors*; Project phase distribution* (it deals with the resources in terms of time, cost, effort etc allocated each of the phase of software project development), *Requirements engineering process* (the set of activities involve for requirement engineering process accomplishment), *Economic process* (deals with the activities undertaken for the production and management of objects wealth), *Process maturity* (deals with the appropriateness and quality of a process use to accomplish task), *Collaboration process* (deals with the activities undertaken in order to realize shared goals) and *Management process* (deals with the planning and controlling the execution of the activities).

Requirements: comprises of factors*; Requirement specification document format* (the standard or format organization follow to generate the documents), *Requirement engineering effort* (amount of exertion expended, work done or attempts), *Requirement representation style* (deals with the styles of requirement representation like visual styles, natural language etc) and *Requirement interpretation* (deals with the understanding of requirements or meaning of requirements).

5 DISCUSSIONS

In this section, we talk about how this work can be helpful for research communities. This work also leads to some possible future work areas. In the end some limitations are also discussed.

5.1. Utilizing Situational Factors:

This work gives the 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, who are working in situational requirement engineering, as it shows the initial list of factors which may generate risks and which directly or indirectly may influence the requirement engineering process in GSD paradigm.

The initial list can also act as a guide for the researchers and practitioners working in situational requirement engineering, 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.

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 requirement engineering process in GSD paradigm, hence extending the investigation to a more comprehensive level.

5.3. LIMITATIONS

The scope of the initial list of situational factors is limited to only risks factors and for this only state of the knowledge is concerned and covered. Hence it does not cover the identification of factors from industry. So it is only from state of knowledge not including the state of practice. The steps to overcome this limitation is taken by designing a questionnaire and surveying the situational factors from the industry which will be ultimately included in the final comprehensive list of situational factors (which will be generated after covering and including the factors from various other domains related to RE process) affecting RE process in GSD.

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 cannot 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. We also cannot ignore the threat related to precision. In our work we tried to have high precision rate but still the maximum precision is not assured.

6 CONCLUSIONS

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. The information used for this identification is taken from the literature by performing systematic literature review (SLR) [5]. In order to have unique identification of each of the factors generating risks, Grounded theory's [6] constant comparison and memoing steps are adopted.

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 requirement engineering, 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.

7 REFERENCES

[1] Kathryn Fryer, M.G., 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. p. pages 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. 352 pages.

[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): p. 315-321.

[9] Ian Sommerville, P.S., Requirements Engineering: A Good Practice Guide 1ed. 1997: Wiley. 404 pages.

[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. p. 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] Peta Darke, G.G.S., User viewpoint modelling: understanding and representing user viewpoints during requirements definition. Information Systems Journal, 1997. **Volume 7**(Issue 3): p. pages 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): p. pp. 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. p. 135-147.

[16] E. Damian, D. and D. Zowghi, RE challenges in multi-site software development organisations. Requirements Engineering, 2003. **8**(3): p. 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. p. 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. p. 2004.

[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): p. 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): p. 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 InformationTechnology, 2001. **Vol. 5**(Iss: 2): p. pp.1 - 20.

[23] Lopez, A., J. Nicolas, and A. Toval. Risks and Safeguards for the Requirements EngineeringProcess in Global Software Development. in Global Software Engineering, 2009. ICGSE 200 Fourth IEEE 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): p. 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): p. 449-480.
[29] Parviainen, P. and M. Tihinen, Knowledge-related challenges and solutions in GSD. Expert Systems, 2011: p. no-no.
[30] Komi-Sirviö, S. and M. Tihinen, Lessons learned by participants of distributed software development. Knowledge and Process Management, 2005. **12**(2): p. 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., et al., A review of awareness in distributed collaborative software engineering. Software: Practice and Experience, 2010. **40**(12): p. 1107-1133.
[33] Ramzan, M., et al., Automated Requirements Elicitation for Global Software Development (GSD) EnvironmentSoftware Engineering, Business Continuity, and Education, T.-h. Kim, et al., Editors. 2011, Springer Berlin Heidelberg. p. 180-189.
[34] Perini, A., et al., 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. p. 440-447.
[35] Skroch, O., A method to evaluate the suitability of requirements specifications for offshore projects Developing Business Application Systems. 2010, Gabler Verlag. p. 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): p. 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. p. 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): p. 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. p. 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. p. 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. p. 139-145.
[42] Ning, A., et al., Requirements Engineering Processes Improvement:A Systematic View, in SPW 2005. 2005: Berlin Heidelberg. p. pp. 151 – 163.
[43] Lings, B., et al., Ten Strategies for Successful Distributed Development The Transfer and Diffusion of Information Technology for Organizational Resilience, B. Donnellan, et al., Editors. 2006, Springer Boston. p. 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. p. 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. p. 25-46.
[46] Herbsleb, J.D., et al., 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. p. 81-90.
[47] Luna-Reyes, L.F., et al., Knowledge sharing and trust in collaborative requirements analysis. System Dynamics Review, 2008. **24**(3): p. 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. p. 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. p. 107-125.
[50] Ullah Khan, S., M. Niazi, and R. Ahmad, 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. p. 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. p. 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. p. 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): p. 249-273.
[56] Calefato, F., F. Lanubile, and C. Ebert, Practice: Collaborative Development Environments, in Global Software and IT. 2011, John Wiley & Sons, Inc. p. 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. p. 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. p. 225-230.

[59] Lamersdorf, A., et al. A Rule-Based Model for Customized Risk Identification in Distributed Software Development Projects. in Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on. 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): p. 61-76.

[62] Casey, V. and I. Richardson, Implementation of Global Software Development: a structured approach. Software Process: Improvement and Practice, 2009. **14**(5): p. 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. p. 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. p. 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. p. 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. p. 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. p. 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): p. pp. 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): p. 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): p. 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. p. 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. p. 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. p. 15-19.

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

[75] John, M., et al., Human and social factors of software engineering: workshop summary. SIGSOFT Softw. Eng. Notes, 2005. **30**(4): p. 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): p. 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. p. 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): p. 319-328.

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

[81] Ramachandran, S., S. Dodda, and L. Santapoor, Overcoming Social Issues in Requirements Engineering. CCSIT, 2011. **3**(133): p. pp. 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. p. 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. p. 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): p. 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. p. 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. p. 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 Software Engineering and Advanced Applications (SEAA), 2011 37th EUROMICRO Conference on. 2011.

[89] Anantatmula, V. and M. Thomas, Managing global projects: A structured approach for better performance. Project Management Journal, 2010. **41**(2): p. 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. p. 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. p. 124-137.
[92] Kauppinen, M., et al., Implementing requirements engineering processes throughout organizations: success factors and challenges. Information and Software Technology, 2004. **46**(14): p. 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): p. 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): p. pp. 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): p. pp. 39 - 71.
[96] Islam, S. and S.H. Houmb. Integrating risk management activities into requirements engineering. in Research Challenges in Information Science (RCIS), 2010 Fourth International Conference on. 2010.
[97] Mistrík, I., et al., Collaborative Software Engineering: Challenges and Prospects Collaborative Software Engineering, I. Mistrík, et al., Editors. 2010, Springer Berlin Heidelberg. p. 389-403.
[98] Asnar, Y., et al. Computer Aided Threat Identification. in Commerce and Enterprise Computing (CEC), 2011 IEEE 13th Conference on. 2011.
[99] Sadraei, E., et al., A field study of the requirements engineering practice in Australian software industry. Requirements Engineering, 2007. **12**(3): p. 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):
p. 878-908.
[101] Crnkovic, I., Component-based software engineering — new challenges in
software development. Software Focus, 2001. **2**(4): p. 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): p. 573-610.
[103] Rodriguez, F., et al., Evaluating Collaboration Platforms forOffshore Software Development Scenarios Software Engineering Approaches for Offshore and Outsourced Development, B. Meyer and M. Joseph, Editors. 2007, Springer Berlin / Heidelberg. p. 96-108.
[104] Lawrence, B., K. Wiegers, and C. Ebert, The top risk of requirements engineering. Software, IEEE, 2001. **18**(6): p. 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): p. 185-205.
[106] Krieg, R., Impact of structured product definition on market success. International Journal of Quality & Reliability Management, 2004. **Vol. 21** (Iss: 9): p. pp.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): p. 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): p. 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): p. 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. p.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. p. 186-214.
[112] Kontio, J., et al. Managing commitments and risks: challenges in distributed agile development. in Software Engineering, 2004. ICSE 2004.Proceedings. 26th International Conference on. 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): p. 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): p. 51-67.
[115] Hall, R.J. LSS: A Tool for Large Scale Scenarios. in Automated Software Engineering, 2006. ASE '06. 21st IEEE/ACM International Conference on. 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. p. 7.
[117] Staats, B.R., Unpacking Team Familiarity: The Effects of Geographic Location and Hierarchical Role. Production and Operations Management, 2012. **21**(3): p. 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): p. 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): p. 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): p. 911-922.
[122] Casey, V., Imparting the importance of culture to global software development. ACM Inroads, 2011. **1**(3): p. 51-57.
[123] Baiqiang, X. and Z. Deming. Engineering safety information in software intensive systems. in Reliability, Maintainability and Safety (ICRMS),2011 9th International Conference on. 2011.
[124] Ebert, C., Requirements Engineering, in Global Software and IT. 2011, John Wiley & Sons, Inc. p. 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): p. 375-393.

[126] Yang, H., et al., Analysing anaphoric ambiguity in natural language requirements. Requirements Engineering, 2011. **16**(3): p. 163-189.
[127] Ralyte, J., et al. A framework for supporting management in distributed information systems development. in Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on. 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: p. n/a-n/a.

Classification	**Situational Factors**	**Risks**	**References**
	Interaction skills	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, 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] [28] [29] [30]
	Interaction styles	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	[26] [27] [31] [28] [29] [30] [32] [33] [36] [37] [38] [39] [40]
	Interaction medium or technology	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,	41] [42] [43] [44] [45] [46] [27] [28] [29] [30] [38] [37]

Classification	Situational Factors	Risks	References
Communication and Distance		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 tools	Lack of individual and shared knowledge understanding, Chaotic and uneven knowledge transfer, Lack of uniform software development environment, Lack of work awareness, Lacking proximity	[37] [29] [47] [32]
	Interaction infrastructure	Effort overhead, High defect frequency, Lack of client involvement, Hidden cost, quality work, product, Lacking proximity	[48] [49] [43] [50] [35] [37]
	Distance	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.	[51][52] [53] [54] [55] [29] [30] [56] [36] [57]
Differences with respect to Culture, Background, Language, Organization and Time	Language	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	[52] [46] [58] [59] [60] [53] [27] [61] [62] [38] [63] [50] [45]
	Cultural background	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.	[52] [64] [65] [66] [67]
	Work Environment	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,	[34] [58] [67] [68] [27] [69] [70] [71]
	View point	Increase in requirement evolution rate	[34]
	Time zone	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,	[72] [52] [58] [73] [74] [75] [59] [53] [76] [54] [55] [61] [56] [33] [40] [77] [78]
	Social background	Lack of quality outcome, Varying meaning for a situation, Lack of effective communication and social interaction, Project mismanagement	[67] [58][76][54][55][79]
	National culture	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	[58][67] [27] [55] [80] [81] [82]
	Organizational culture	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	[52] [83] [73] [59] [60] [76] [27] [55] [30] [41] [84] [80] [33] [36] [34] [81]
	Social climate	Decrease in team productivity, Inefficient collaboration and communication process, Lack of social interaction, Project mismanagement, Lack of quality decisions	[85] [76] [79] [82]
	Political difference	Lack of social interaction, Project mismanagement	[79] [34]
	Time shifts	Inaccurate task allocation	[38]
	Inter group culture	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	[67] [83] [27] [34] [33]
	Organization Structure and boundaries	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 decisions, partners week relationships, lack of trust, lack of common goals, weak contractual relations.	[86] [52] [87] [58] [88] [89] [50] [91] [45]
	Organization person retention strategy	Loss of key personnel.	[92]
	Organization policies and strategies	Inadequate client involvement, inefficient requirement engineering process, collecting data without improving requirement engineering process.	[92] [93]
	Team Awareness	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.	[27] [86] [72] [85] [90]
	Data repository	Data loss, lack of requirement specification quality	[94] [95]
	Domain knowledge	Lack of quality outcome, erroneous requirements, budget and schedule overrun, productivity	[73] [87][59]

Classification	Situational Factors	Risks	References
Knowledge management and awareness	level	downfall,	
	Internal knowledge level	Lack of frequent communication, lack of competence	[54]
	Knowledge management techniques and procedures	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.	[96][60][88][27][33][97]
	Business knowledge	Delay in problem domain clarification, extra cost for rework	[61]
	Access management	Unintended data editing, un/intentional disclosure for personal gain.	[98]
	Tacit knowledge level	Lack of knowledge sharing, lack of synchronization and act	[29]
	RE practice knowledge	Lack of accurate requirements fault modeling.	[99]
	Configuration management	Effort overhead, work unawareness	[32][48]
	Knowledge management awareness	Lack of quality decisions, lack of control, lack of maintained awareness, , unaware of job responsibilities of remote team members, high defects frequency	[27]
	Requirement management	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.	[28][100][101][102][103][104]
Management	Coordination skills	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.	[52] [27][38] [77] [97] [45][50]
	Coordination technique	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	[27] [30] [52] [45] [50]
	Competence	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,	[27] [32]
	Decisions	Misalignment of tools with expectations, unrealistic estimation	[27]
	Supervisor sub-ordinate relationship	Lack of proactive transparency	[105]
	Management strategy	Increase in requirement uncertainty, lack of understanding of project scope, loss of key personals	[92]
	Cooperation approach	Lack of understanding activities accelerating the knowledge sharing, lack of trust, troublesome disagreements, less social capital.	[45][50]
Tools, Technologies, Techniques and Standards	Technique selection	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,	[106] [107] [108] [93] [40]
	Technical expertise	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	[109][110] [108] [48] [31]
	Standards	Low technical efficiency, high frequency of conflicting requirements, lack of standards, lack of shared understanding, lack of effective coordination, lack of group awareness.	[78] [60]
	Tools	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	[78] [27] [30] [93] [80] [111] [40]
	Technology	Lack of team work performance, software project failure, lack of uniform software development environment, lack of early architecture quality, greater frequency of requirements uncertainty, lack of information and artifact sharing, lack of decision making quality.	[112][30][69][84] [113] [33] [91]
Stakeholders	Competence and experience with in application	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.	[114] [67] [115] [113]
	Motivation level	Lack of team performance, activity errors, wrong team set up and adjustment.	[114] [82]
	Familiarity with each other	Inadequate communication, increase in staff problems, lack of team effectiveness, mistrust, lack of work team cohesion, lack of effective collaboration.	[116] [85] [117]
	Preferences related to project	Inefficient collaboration, poor quality outcome.	[67] [109]
	Knowledge	Lack of team effectiveness, inefficient collaboration, ineffective communication, lack of	[67] [27]

Classification	Situational Factors	Risks	References
	Exchange ability	requirement rationale understanding, inefficient requirement engineering process.	
	Background	Lack of shared understanding.	[29]
	Personnel trust	Lack of shared understanding, lack of efficient information sharing, lack of efficient collaboration	[67] [41]
	Leadership skills	Inefficient requirement engineering process, lack of efficient information flow tailoring, person becomes bottleneck.	[92]
	Personnel/group relationship	Lack of effective collaboration, lack of quality outcome, on stake security, mistrust, weak team cohesion.	[67] [112] [105] [118][119]
	Decision capability	Lack of accurate estimation, lack of quality outcome.	[92]
	Utility values	Lack of project efficiency.	[67]
	Knowledge level	Lack of team effectiveness, inefficient collaboration, ineffective communication, lack of requirement rationale understanding, inefficient requirement engineering process.	[67] [96] [27]
	International work experience	Lack of efficient collaboration.	[67]
	Relation to the project	Inefficient collaboration, lack of quality outcome, delay.	[67]
	Power/position in organization	Lack of satisfied requirements, inefficient communication, and high rate of requirements conflicts.	[120]
	Client involvement	Lack of client implication, conflicting approaches to requirement engineering process, requirements variability, higher issues with user abilities and concurrences.	[27][76] [121]
	Carriers prospects	Loss of competency.	[67]
	Client commitment	Client misalignment with project goal, requirement variability.	[27]
	Client availability	Lack of participation from clients, higher issues with user abilities.	[27]
	Background	Lack of efficient requirement elicitation and negotiation	[65]
	Partner interpretation	Lack of effect group problem solving ability, measurement scale misconception, inappropriate requirement validation, lack of quality outcome, delay.	[92]
	Team management capabilities	High rate of workforce turnover, lack of remote staff information.	[80]
	Common work experience	Inaccurate task allocation, mistrust, delay, effort overhead, inefficient collaboration and communication.	[67]
Project and Process	Project phase distribution	Problematic overall joint management, problematic responsibility share, extra management needed at each location.	[61]
	Requirements engineering process	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	[122] [104] [59] [28] [70]
	Economic process	Unacceptable results.	[123]
	Process maturity	Productivity downfall, inefficient communication, lack of quality requirements documentation, effort overhead,	[42] [82]
	Collaboration process	Lack of communication, collaboration and control.	[52] [90]
	Management process	Lack of quality decision making.	[91]
Requirements	Requirement specification document format	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.	[96][87] [28][124][125][128] [126]
	RE effort	Scope creep, lack of attention to importance concerns, unclear requirements.	[31]
	Requirement representation style	Choosing wrong solution for requirement implementation, reducing the solution domain.	[127][28]
	Requirement interpretation	Effort overhead, inadequate solution, inaccurate requirement capturing and understanding, lack of quality outcome, delay.	[30][126][35] [35]