



UNIVERSITY OF GOTHENBURG



# Distributed Software Development Agile Risk Management Framework: A Systematic Literature Review

*Master of Science Thesis in Software Engineering and Management*

NAVID VAJDI  
RAJA MANZAR ABBASS

Department of Computer Science and Engineering  
CHALMERS UNIVERSITY OF TECHNOLOGY  
UNIVERSITY OF GOTHENBURG  
Göteborg, Sweden, June 2014

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

## Distributed Software Development Agile Risk Management Framework: A Systematic Literature Review

Navid. Vajdi,  
Raja Manzar. Abbas,

© Navid. Vajdi, June 2014.

©Raja Manzar. Abbas, June 2014.

Examiner: Matthias Tichy

Chalmers University of Technology  
University of Gothenburg  
Department of Computer Science and Engineering  
SE-412 96 Göteborg  
Sweden  
Telephone + 46 (0)31-772 1000

Distributed Software Development Agile Risk Management Framework: A Systematic Literature Review

Department of Computer Science and Engineering  
Göteborg, Sweden June 2014

## *Abstract*

The software industry has evolved during the last two decades due to the globalization, maintenance and development of the world. The development and maintenance of software has shifted from being single site to different geographical locations across the globe. Distributed Software Development adds new factors to development, such as distance in culture and time, which complicate development further. Distributed Software Development, therefore, become much more difficult to manage than co-located projects. To manage DSD projects and to restrict risks, software risk management plays a vital role in minimizing risks. Our report presents the findings of a systematic review of the literature related to the techniques for managing the risks concerning Globally Distributed Software Development, and risks that are managed through these techniques. At the end, we present our own DSD Agile Risk Management framework to manage risks and minimize the damage from these risks and evaluated our framework via a semi-structured case study intended to evaluate distributed software development projects.

**Keywords:** Distributed Development, Global Software Development, Risk Management, Risk Frameworks, Risk resolution techniques.

**Contents**

- Abstract ..... 3
- 1. Introduction..... 5
- 2. Related Work ..... 6
  - 2.1 Background and Goals ..... 7
- 3. Design and Methodology ..... 8
  - 3.1 Systematic literature Review Procedure ..... 9
  - 3.2 Interviews ..... 11
- 4. Result ..... 11
  - 4.1 *Techniques and Risks* ..... 11
  - 4.3 *Requirements* ..... 14
  - 4.5 *Communication* ..... 17
  - 4.6 *Knowledge management and Awareness* ..... 18
  - 4.7 *Culture and Distance* ..... 19
  - 4.8 *Project Delivery Failure* ..... 20
  - 4.8 *Trusts* ..... 20
  - 4.9 *Supplier and Customer* ..... 21
  - 4.10 *Miscellaneous Risks* ..... 22
- 5. Risk Management Framework ..... 23
- 6. Evaluation..... 25
- 7. Discussion about Evaluation ..... 28
- 8. Conclusion and Outlook..... 30
- 9. Limitations and Future Work ..... 33
- 10. References ..... 35
- 11. Primary Studies. .... 38
- 12. Appendix ..... 42
  - Appendix A: Checklist for risks..... 42
  - Appendix B: Questions asked for evaluating our DSD Agile Risk Management Framework 43

# 1. Introduction

The globalization of the world during the last two decades has created changes in the software industry as the development and maintenance of software shifts from being single site to different geographical locations across the globe. This relatively recent trend is called “global” or “distributed” software development [1]. The favorable factor in doing distributed software development is the availability of cheap and skilled labor force which leads to economic and political benefits. The main aim of this is cost and resource optimization [2]. But along with these advantages, it brings some disadvantages as well. A lot of single site software development projects are already complicated, and going distributed increases the risk of under achieving agreed quality, time constraints and overdue in time (or something else one could "under achieve"), budget and also increases the chance of failure of project [3]. Distributed Software Development also adds new factors to development, such as distance in culture, time and space [2]. Due to these, Distributed Software Development is facing a variety of risks such as cross-cultural management, communication, collaboration, coordination, and complicate development further. Distributed Software Development, therefore, becomes much more difficult to manage compared to co-located projects and often operates at a suboptimal performance level [2]. To manage these projects and minimize the risks in Distributed Software Development, Software Risk Management is an important project management activity [10].

Software Engineering Institute (SEI) defines risk as “the possibility of suffering loss” [4]. Loss in development projects is defined as “the impact to the project which could be in the form of diminished quality of the end product, increased costs, delayed completion, loss of market share, or failure.” [4]. A risk can be identified and addressed with the help of Software Risk Management. Software Risk Management is “a discipline whose objectives are to identify, address, and eliminate software risk items before they become either a threat to successful software operation or major sources of software rework” [5]. Advantage of using Software Risk Management is that it identifies risks early and systematically in a project and action can be taken before they can do some harm to the project [7].

It is frequently observed that companies move to more simple tactics such as near shoring or onshore site once they are over with the stage of focusing on labor costs [2]. However, work distribution in industry is frequently done unsystematically focusing only on few decisive factors such as cost rates, availability, and expertise [26]. The fact that qualified practitioners in DSD are conscious and can tell failure stories but as we observed in the industry, the usual threats of DSD projects and their reasons are not adequately measured [27]. Moreover, work is distributed ad-hoc and not in a clear manner, devoid of any definite process or guidelines. The individual expertise of the decision maker is responsible for the quality of the distributed decision. To have a better understanding of these problems and to manage these challenges, software risk management plays an important role. We achieved explaining about software risk management with the help of a Systematic Literature Review and present our own

framework (DSD Agile Risk Management Framework) to manage risks in DSD and minimize their impacts.

We address these problems via two contributions:

- Based on SLR we identify techniques and risks.
- Proposed an experience based framework and evaluate the framework in industrial context.

The study is organized as follows. Section 2 gives an overview of related work of SLR-based risk management in DSD and some background of our study. Section 3 explains applied SLR procedure, case study method and research questions. Section 4 presents techniques found related to maintaining risks in Distributed Software Development and our proposed framework. Section 5 presents analysis of the results of SLR and evaluation of our proposed framework through a company and conclusion and further work is proposed in section 6.

## *2. Related Work*

As mentioned in the previous section, there is a gradual increase in the number of studies in the area of Risk Management in Distributed Software Development. Most exemplify particular risks and challenges of distributed development work and mention strategies practiced to manage the risks. Some of the risks are, for example, mentioned by Ebert, et al [s1], who focuses on software project and product related risks, by Islam, et al. [s4], who analyze on goal and risk factors in Offshore Outsourced Software Development from Vendor's viewpoint. Massachusetts [s6], developed a new framework to identify the dynamic risks in GDSD projects and mitigate them using agile risk management practices. Framework and strategies are reported by Magnusson and Sung-Chun Chou [s7], who provides a risk and compliance management framework for outsourced DSD of financial applications and ERP systems, by Lopez, et al [s11], who provided risks and safeguards for the requirements engineering process in Global Software Development. Sven Overhage, et al [s32], proposed a method to evaluate the suitability of requirements specifications for Offshore Projects, or by Betz, et al [s12], who focused on knowledge transfer in IT Offshore Outsourcing Projects. Some studies have collected risks and techniques for managing risks using SLR. All these studies have focused either on risks or techniques or on single area of software development rather than software development as whole. Whereas, our area of concern is to find risks and techniques in distributed software development. As our main focus is on doing SLR, some studies about SLR are mentioned below:

In 2009, Hossain Babar et al [s29] did a SLR about risk identification and mitigation processes for using Scrum in Distributed Software development. They identified key challenges, due to global project distribution, that restrict the use of Scrum and explore the strategies used by project managers to deal with these challenges. They did not cover other methodologies and they only focused on Scrum.

In 2009, Alejandro Lopez et al [s11] also did a SLR which led to the compilation of a repository which gathers risks concerning requirement engineering (RE) when developed in a distributed software development environment, as well as presented set of safeguards, helping overcoming such risks. However, their study was limited only to requirement engineering.

Stefan Schneider et al [s45] provided solutions in DSD by doing Systematic Literature Review. They analyzed solutions associated with DSD, while also evaluating the level of empirical validation of said solutions. As a starting point they presented a DSD model, designed to categorize solutions into process areas, useful for the analysis of the research community's contributions to state-of-the-art. They point out that Requirements Management/Development and Distributed Team and Project Management are very well populated, that is, a lot of research and subsequent solutions have been presented to tackle DSD-related problems. There are however several areas, most pre-dominantly Product Integration and Project Monitoring and Control, that are very scarcely populated. This study focuses on identifying and filling gaps in studies by providing solutions and do not concentrate on problems and no framework is given for risk management as a whole.

In 2010, da Silva et al [14] did a SLR about challenges and solutions in Distributed Software Development Project Management. The objective was to collect and systematize reported knowledge in terms of what are the difficulties in managing DSD projects, what are the best practices to overcome these difficulties, and how existing models and tools support these practices. But due to their limitations they suggested to enhance the SLR work and do further research by enhancing primary studies.

Stouby Persson and Lars Mathiassen [s31] also did a SLR on DSD risk management in 2009. They did a SLR and provided their own framework to manage risks in Distributed Software Development. They focused on synthesizing what they knew about risks and risk resolution techniques into an integrative framework for managing risks in distributed contexts. They searched approaches for predefined risk fields [9] and then integrated these approaches to make a framework. This paper was published in 2009, but the SLR was done on the literature from 1995 to 2005 and they mentioned that there were no Frameworks available for managing risks related to geographical distribution until 2005.

## *2.1 Background and Goals*

Distributed Software Development, becomes much more difficult to manage compared to co-located projects and often operates at a suboptimal performance level [2]. Hence, there is more need of risk management in DSD than co-located projects and it needs to be managed carefully.

A lot of studies have focused on risk management as shown in related work. In general, the related work shows about risks or techniques or frameworks in DSD are either very generic or very much narrowed down on selected criteria [s1, s4, s6, s11, s32]. The proclaimed risk management approaches for DSD projects, on the other hand, do not systematically consider the impact of risk management on DSD projects and therefore cannot be used either for

supporting techniques to manage risks or to find risk areas. In addition, very little is known about frameworks which have been validated and evaluated in industrial context.

As mentioned in the Section 2, da Silva et al [11] did a SLR about challenges and solutions in Distributed Software Development Project Management. The objective was to collect and systematize reported knowledge in terms of what are the difficulties in managing DSD projects, what are the best practices to overcome these difficulties, and how existing models and tools support these practices. But due to their limitations they suggested to enhance the SLR work and do further research by enhancing primary studies. Also they have mentioned about techniques given in literature to manage DSD challenges but they did not mention related risks with these techniques and have not provided any framework to manage these risks. Hence, they inspired us that there is need to enhance the primary studies to find the techniques to manage risks and have risk areas with the related techniques.

Another study which used SLR to collect different risks and resolutions in all aspects of Distributed Software Development was done by Stouby Persson and Lars Mathiassen [8]. They did a SLR and provided their own framework to manage risks in Distributed Software Development. They focused on synthesizing what they knew about risks and risk resolution techniques into an integrative framework for managing risks in distributed contexts. They searched approaches for predefined risk fields [9] and then integrated these approaches to make a framework. This paper was published in 2009, but the SLR was done on the literature from 1995 to 2005 and they mentioned that there were no Frameworks available for managing risks related to geographical distribution until 2005 and based on our pre study on literature from 2000 to 2013, we realized that 89% of them were published after 2006 and as there is a gradual increase in number of studies in the area of managing risks in Distributed Software Development in recent years.

On this background and increasing number of studies in recent years, we will perform on our SLR on studies from 2006 to 2013 to fill this gap and to identify the techniques which are mentioned in the literature and propose our own framework to manage these risks.

This SLR can help those researchers and Software project managers, who are involved or work with the field of Distributed Software Development to gain knowledge about existing techniques to manage risks.

### *3. Design and Methodology*

Our research methodology is composed of two parts, first one is an explicit quantitative meta-analysis of available data on risk management by doing systematic literature review and second part is qualitative review to report evidence about our proposed framework.



### *3.1 Systematic literature Review Procedure*

The Systematic literature review aim's to present a fair evaluation of a research topic by using a trustworthy, rigorous, and auditable methodology [6]. As said by Kitchenham [6], SLR permits the identification, evaluation and interpretation of all the available relevant studies related to a particular research question, topic area or phenomenon. It synthesizes existing work to give more scientific value.

A lot of studies [2,5,16] have focused on risk management but mostly on a particular area of project management like in requirement engineering [2], designing, development or testing [5] but not focusing on project management as a whole. In general; risks, techniques or frameworks in DSD are either very generic or very much narrowed down on selected criteria. The proclaimed risk management approaches for DSD projects, on the other hand, do not systematically consider the impact of risk management on DSD projects and therefore cannot be used either for supporting techniques to manage risks or to find risk areas. In addition, as can be seen in related work that very little is known about frameworks which are validated and evaluated in industrial context and we found a need to propose a framework which can be useful for risk management and includes all these techniques and risks in it. Keeping these problems in mind, we formulated following questions:

**Q1:** What are risk areas in distributed software development and by which techniques can they be managed?

**Q2:** What framework can be used for DSD Risk Management which has all the techniques and risks used in Distributed Software Development in question 1?

**Q3:** To what extent is the framework beneficial to practitioners?

We pursued SLR to achieve our goal, which were (1) to know the techniques that are proposed in literature to manage risks and challenges identified in the Distributed Software Development projects, (2) to gather these techniques, risks and develop a framework for DSD Agile Risk Management and after performing SLR, (3) to evaluate the framework by doing a case study to know if the framework supports individual learning or it supports organizational learning. Keeping our goals in mind, SLR protocol was designed and followed.

To find the literature for addressing Research Question 1, we used both automatic and manual searching. To find the proper search term for performing the automatic search, several key words and search terms were tested and following search term was defined:

("Global Software development" AND Risk) OR ( "Global Software Engineering" AND Risk) OR ( "Distributed Software Engineering" AND Risk) OR ( "Distributed Software development " AND Risk) OR ( "Offshore Software Engineering" AND Risk) OR ("Offshore Software Development" AND Risk).

These search terms were used in four important scientific literature databases, namely: IEEE Explore, Science@Direct, ACM Digital Library and SpringerLink. Also, manual searching was done by tracking related references from the primary studies during information extraction.

A study was selected by analyzing the title, abstract, and keywords from the studies retrieved by the automatic search. In some cases, it was necessary to read the entire document to determine its relevance. In total, 439 studies were found and after removing the repeating studies and applying inclusion criteria, 104 studies were included. The inclusion criteria for determining, whether study was relevant was by studies regarding Software field, non-co-located development, published between 2006 -2013 and containing information about managing risk were included.

Next, to obtain the primary studies, exclusion criteria was applied on initial studies. Any study that did not suggest/recommend or contain review or implement/define at least one technique for managing risks as well as book chapters was excluded from initial studies. After applying the exclusion criteria, 43 primary studies were obtained and additionally, we found 7 more primary studies by manual searching.

Figure 1 shows the number of results per data source including the result of manual searching. As it is shown in figure 2, the main data source in primary studies list is IEEE Explore.

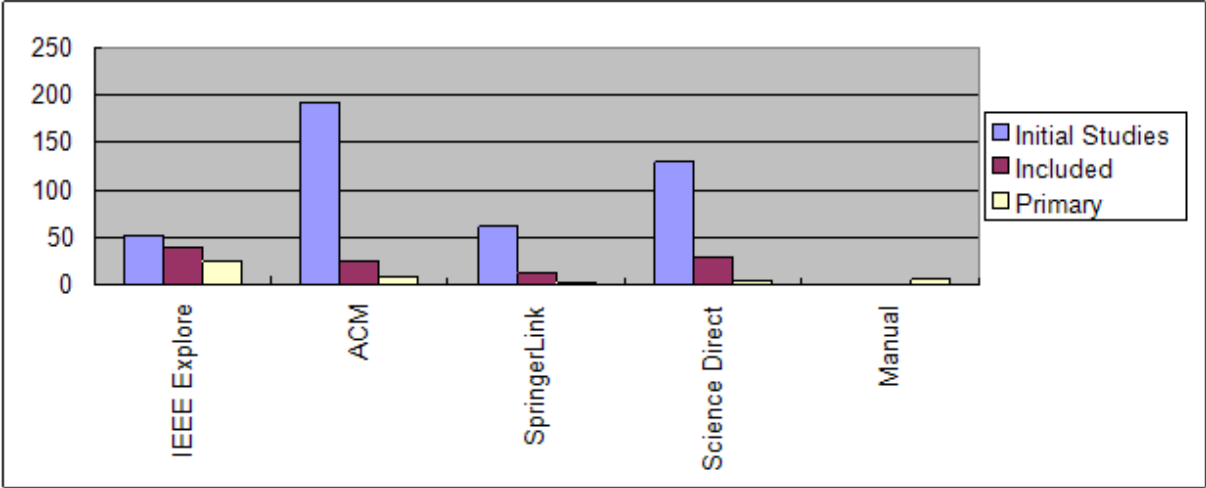


Figure 1: Studies per data source

Information was extracted using data extraction forms which were designed based on research questions. Information extracted was techniques for managing risks and associated risks with techniques. Data extraction forms were examined before final usage. Studies were equally divided between both authors for data extraction. We did peer review for maintaining quality during applying inclusion and exclusion criteria as well as during data extraction.

We mapped our conception of risks and sub risks into different categories based on studies [s1, S11, s31] which have mapped risks into categories based on their conception and we contributed by the merging of these different categories into our main categories. After formulating our main categories, we then placed scattered sub categories from different literature into our main categories, which further diversified the main categories.

## *3.2 Interviews*

The DSD Agile Risk Management framework is developed to evaluate risk management in a Distributed Software Development Organization, trying to understand how to make decisions in a DSD project. It offers a structure for classifying DSD problems and solutions and contributes to the better understanding of the distributed software development projects. It helps researchers and practitioners in identifying new problems and indicates the solutions for them. It analyzes specifically the risk management process, since decisions at the strategic and tactical levels impact project development at the operational level.

The evaluation of framework is done via semi-structured interviews intended to analyze risk management in Distributed Software Development Projects. Semi-Structured Interviews is a qualitative method which is appropriate when studying state of the art situations where practice precedes theory (28). The interviews were done on eight team leaders in three different organizations. The data collected was constituted of primary sources. We interviewed employees of different organizations to gain knowledge about their objectives, and to evaluate our proposed DSD Agile framework. Both face- to-face 'and telephone interviews were conducted. Each lasted from 30 to 45 minutes. We asked interviewees about their use and opinions about the framework and the difference it made to their work and to the organization by using it. Main purpose of the study is to evaluate if our DSD Agile Risk Management framework is useful in their project context or not.

## *4. Result*

To address the first research question about the techniques used for managing risks in Distributed Software Development and the risk areas managed by techniques identified, we identified techniques which were provided for managing risks in Distributed Software Development given in the literature and we extracted the risks which were associated by given techniques in the previous step. According to Boehm [5], risk areas consist of a number of related risk factors, which together possess a threat to the project's success. We categorized risks to give joint assessment of risk areas representing category in which risk might fall.

### *4.1 Techniques and Risks*

In this section, we mapped the given techniques and proposed risks, identified through the SLR. These mappings suggest about the proposed techniques covering the available sub risk categories of risk as well as provide a balanced view of the overall portfolio of ten possible main risk categories.

Column one contains the techniques, whereas column two contains the literature in which these techniques were found and column three contains the respective sub categories of risks which are shown below:

### 4.1.1 Project Management and Coordination

High organizational complexity, scheduling, intellectual property rights management, coordination and cost estimation become more problematic in distributed environments as a result of volatile requirements, changing specifications, cultural diversity, and the lack of informal communication. Especially coordination in multisite developments becomes more difficult in terms of articulation work; problems derived from communication, lack of group awareness, and the complexity of the organization appears which it influences the way of the work must be structured and managed [s36]. All the techniques to manage Project Management and Coordination risks are shown in table 1.

Techniques	Literature	Sub Categories of Risks
<ul style="list-style-type: none"> <li>• Systematically train engineering and management on Intellectual Property Right (IPR)</li> <li>• Rigorously apply a strong policy on IPR protection</li> <li>• Encourage innovation on all Global Security Environment</li> </ul>	S1	IPR Management
<ul style="list-style-type: none"> <li>• Distribute work across regions</li> <li>• Anticipate wage increases</li> <li>• Evaluate your own and suppliers' business models</li> <li>• Combine expert-driven and data-driven cost estimation methods</li> <li>• Provide measures to facilitate decision making</li> </ul>	S1,S16,S21	Wage and Cost inflation
<ul style="list-style-type: none"> <li>• Compute the average project schedule time that incorporates a feature productivity model</li> <li>• Derive the project structure and statistics from a Project Management database to generate a stochastic simulation model</li> </ul>	S9,S5	Schedule failure
<ul style="list-style-type: none"> <li>• Establish a well-defined agile leadership hierarchy</li> <li>• Establish the coordination structure between sites</li> </ul>	S11,S25,S33,S39,S28,S37	Coordination

<ul style="list-style-type: none"> <li>• Use work break down approaches and Architectural Contribution</li> <li>• Use of collaborative tool</li> </ul>		
<ul style="list-style-type: none"> <li>• Use agile tools and techniques</li> <li>• Define requirements as user stories, build out thin slices of the product, and frequently review those features with the customer</li> </ul>	S35	Timeline Risks

Table1: Mapping classification of risk management techniques and Project Management and Coordination risks

For risks related to intellectual property rights, techniques like systematically training engineering and management on intellectual property rights, establishing and rigorously applying a strong policy on intellectual property rights protection and encourage innovation on all global security environment sites and promote patents [s1].

For risks related to wage and cost inflation, techniques were to distribute work across regions , anticipate wage increases as well as carefully protect against supplier lock-in and to evaluate your own and suppliers’ business models over future years [s1].

For risks related to scheduling risks, compute the average project schedule time that incorporates a feature productivity model [s9]. Predict the outcomes of future projects, if certain risks come true as probability of certain events and the extent of possible negative consequences based on experience factory to gain project experience. Also one suggestion was given to involve optimistic employees in planning to contribute in project management [s10].

To solve risks of coordination breakdown, use a prioritized set of coordination risks and potential mitigations for the risks. This allows a project to better understand the impact that their decisions have on coordination thus potentially avoiding decisions that may lead to coordination breakdowns. To solve risks related to coordination, establish a well-defined agile leadership hierarchy, with an analyst playing the role of coordinator. Their responsibilities are well defined, and play their role in a skilled and firm way. Below them in the hierarchy are team heads, which are responsible for the analysts that belong to their work group [s11].

To establish the coordination structure between sites to have a direct impact on the severity of GSD-specific risks such as communication and coordination problems [s33]. Use work break down approaches and Architectural Contribution to reduce coordination risks [s47, s48]. Use collaborative tool to improve support distributed product development projects that enhance the coordination capabilities of the development organization [s39].

Techniques to overcome risks like cost and time in projects were given to combine expert-driven and data-driven cost estimation methods in order to provide a method that can be used in organizations where only little data is available [s16]. To avoid risk of friction between distributed team members, assess processes like change management and knowledge transfer is crucial to avoid the risk of friction between distributed team members [s20].

To reduce the risk of decision-making with regard to cost, time and effort estimation, provide measures to facilitate decision-making and plan off shoring the application. Document and use historical risk data and focus on top risks in order to reduce the effort and time in managing software risks [s21, s23].

For handling the day-to-day operations, telecommunication companies need network management functions such as configuration management, account management, fault management, performance management, and security management and it should be used as a standard [s30]. For solving schedule risks, derive the project structure and statistics from a projects management database to generate a stochastic simulation model [s45].

For mitigation of managerial implications, involve strategies for developing a collaborative work culture with clients [s18]. Using some tools and techniques from agile such as test first development, continuous integration, and pair programming are powerful risk reduction mechanism that ensures the quality of product and reduced uncertainty [s35].

Define requirements as user stories, build out thin slices of the product, and frequently review those features with the customer, can go a long way to making sure about building the right product to mitigate timeline risks [s35].

### 4.3 Requirements

With the division of development work in an inter-organizational and intercultural context, requirements specification becomes the central means to communicate the development scope as explicitly as possible. The suitability of requirements specification hence often is mission critical in offshore projects [s32]. All the techniques to manage requirement risks are shown in table 2.

Techniques	Literature	Sub Categories of Risks
<ul style="list-style-type: none"> <li>Review and sign-off of all requirements and also monitor and control the requirements change index</li> </ul>	S1	Change of Requirements
<ul style="list-style-type: none"> <li>Extend goal-modeling language to accommodate software development risk management activities during requirements engineering phase</li> </ul>	S4	Underspecified Requirements
<ul style="list-style-type: none"> <li>Add Attribute</li> <li>Fine grained specification</li> <li>Evaluate the Suitability of Requirements</li> <li>Use meta-model and associated visual notation</li> </ul>	S11,S13,S22,S32	Rationale Requirements
<ul style="list-style-type: none"> <li>Use methods and tools for clarifying and tracing requirements</li> </ul>	S43,S26	Tracing Requirements

<ul style="list-style-type: none"> <li>• Comprehensive framework to assure the flow down of requirements engineering to design</li> </ul>		
<ul style="list-style-type: none"> <li>• Use groupware supported Distributed Software Architecture evaluation process</li> <li>• Use contracts</li> <li>• Use of Use Cases</li> </ul>	S17,S40	Requirements Negotiation

Table2: Mapping classification of risk management techniques and Requirements risks

To overcome risk of requirements, review and sign-off of all requirements and also monitor and control the requirements change index [s1]. To manage incomplete or underspecified requirements risks, use goal-modeling language to accommodate software development risk management activities during requirements engineering phase. This facilitates to control risks from the early phases of development, as many projects fail due to incomplete or underspecified requirements [s4].

To solve risks related to rationale requirements, define an attribute that should be included within the requirement to explain its rationale and to justify the way they were formulated [s11]. Use fine grained specification to prevent the risk of requirements and evaluate the suitability of requirements specifications [s13, s35].

Use associated visual notation which provides several benefits in term of distributed requirements activities. From an industrial perspective, it enables project managers to plan, analyze, and optimize their distributed requirements engineering processes, so that they can understand their existing processes, identify weaknesses and problems, and establish improved processes and appropriate supporting infrastructure [s22]. It provides a common language for modeling distributed requirements projects and activities, and thereby facilitates comparisons across projects. These comparisons make it possible to identify recurring patterns of collaboration, common obstacles, and best practices used for collaborative requirements engineering activities [s22].

For tracing requirements, assess the different tracing approaches involved in generating, reviewing, and maintaining traces to ensure their correctness overtime and use methods and tools for clarifying and tracing requirements in the project context [s43].

For specification of requirements, assess the appropriateness of a software requirements specification by assessment of three types that are bundled into a comprehensive framework to assure the flow from requirements engineering to design [s46].

To overcome the risk of requirements negotiation, a concept of groupware supported distributed software architecture evaluation process, which does not require the stakeholders to be physically co-located is proposed. The proposed process is expected to address a number of logistical issues such as software inspections and requirements negotiation [s17].

Contracts are used as a mean to support the ongoing negotiation processes to solve negotiation risk in Requirement Engineering. There is a significant need to improve the

ability of supporting the ongoing negotiation processes that are prevalent throughout the project lifecycle [s42].

#### 4.4 Task distribution

As in traditional Software Development, the tasks represent possible risks in GSD, but for slightly different reasons. When the overall project task is divided and distributed across several sites, task uncertainty emerges, because participants may lack information about the task, its purpose and their own contribution to the overall task. Task uncertainty represents lack of information needed to develop the software and it can result in slow change coordination and process and relational conflicts [s31]. All the techniques to manage Task distribution risks are shown in table 3.

Techniques	Literature	Sub Categories of Risks
<ul style="list-style-type: none"> <li>• Execute pooled buffers and establish long-term retention models and periodically conduct employee engagement surveys</li> </ul>	S1	Staff Turnover
<ul style="list-style-type: none"> <li>• Use distribution model</li> </ul>	S15	Decision Support
<ul style="list-style-type: none"> <li>• Granular work breakdown</li> <li>• Support systematic task allocation decision</li> <li>• Phased functionality, prototyping, or pilot testing for breaking projects into small parts</li> </ul>	S19,S33,S34, S48	Task Distribution/Allocation

Table3: Mapping classification of risk management techniques and Task distribution risks

Risks related to staff turnover in task distribution are discussed and gave some techniques to overcome them. Techniques include learning to deal with staff turnover by means of pooled buffers and establishing long-term retention models and periodically conduct employee engagement surveys [s1].

To improve decision support in task distribution, a distribution model should be used to improve the decision support to reduce the risk of task distribution [s15]. Granular work breakdown offers opportunities for parallelization in the project. This is through examining



how tasks can be loosely coupled, taken off the critical path and executed earlier by any available resource [s19].

To solve task allocation risk, a model for supporting a systematic task allocation decision in distributed development projects is used that is based on multiple criteria and influencing factors [s33].

Large projects should be transformed into smaller projects through phased functionality, prototyping, or pilot testing to reduce task related risks and increase success [s34].

### 4.5 Communication

Software Development relies heavily on quick information flows and going global, adds new factors to development, such as distance in culture, time and space. Due to these factors, Distributed Software Development is facing a variety of challenges such as communication. Distributed software projects therefore become much more difficult to manage than collocated projects and often operate at a suboptimal performance level [13]. All the techniques to manage communication risks are shown in table 4.

Techniques	Literature	Sub Categories of Risks
<ul style="list-style-type: none"> <li>• Do contextual training</li> <li>• Establish the communication structure between sites</li> <li>• Easing mismatches with frequent deliveries</li> <li>• Allocate relatively independent modules to different teams</li> <li>• Compare the frequencies of the SPI barrier’s value</li> </ul>	S2,S25,S33,S40,S37	<ul style="list-style-type: none"> <li>• Lack of Contextual Skills</li> <li>• Communication Infrastructure</li> </ul>

Table4: Mapping classification of risk management techniques and Communication risks

In order to improve issues related with communication, contextual training like concepts related to communication, trust acquisition, and cultural differences, should be planned as a mandatory training in the organizations’ on boarding training program and when possible, team members should use communication and collaboration technology [s2].

Establish the communication structure between sites, and thus have a direct impact on the severity of GSD-specific risks such as communication risk [s33].

To improve communication, do easing mismatches with frequent deliveries and making organizational responsibilities more transparent [s40].

A contradictory solution was proposed to minimize cross-site communication by allocating relatively independent modules to different teams [s48].

Compare the frequencies of the Serial Peripheral Interface (SPI) barrier’s values against the occurrences of other SPI barrier’s values; the relative importance of each barrier can be identified as communication [s25].

#### 4.6 Knowledge management and Awareness

Developers need to have as much information as possible at their disposal, and to know the full status of the project and its past history, which will in turn allow them to create realistic assumptions about the project. Frequent changes in processes, lack of continuity in communications, and lack of collaborative tool integration cause remote groups to be unaware of what is important in the project because they do not know what other people are working on. As a consequence, they cannot find the right person and/or timely information which will enable them to work together efficiently, resulting in misalignment, re-planning, redesign, and rework [s36]. All the techniques to manage Knowledge management and Awareness risks are shown in table 5.

Techniques	Literature	Sub Categories of Risks
<ul style="list-style-type: none"> <li>Keep information about analysts and work groups working on each requirement through attributes</li> </ul>	S11, S49, S47	Share Knowledge
<ul style="list-style-type: none"> <li>Start projects with a kind of “boot camp”</li> </ul>	S12	Knowledge Awareness
<ul style="list-style-type: none"> <li>Clear strategy regarding what knowledge needs to be protected must be established before global projects are launched</li> </ul>	S12	Knowledge Protection
<ul style="list-style-type: none"> <li>Define contact persons for the different knowledge areas in a project</li> <li>Install a knowledge management team</li> </ul>	S12, S47	Knowledge Creation
<ul style="list-style-type: none"> <li>Introduce questionnaire</li> </ul>	S27	Knowledge Mining
<ul style="list-style-type: none"> <li>Improve communication with the client</li> </ul>	S18	Knowledge Transfer

Table5: Mapping classification of risk management techniques and Knowledge management and awareness risks

The team members’ experiences, methods, decisions, and skills must be accumulated during the development process through effective information-sharing mechanisms, so that each team member can use the experience of his/her predecessor and the experience of the team accumulated during development, thus saving costs and time by avoiding redundant work [s36].

To share knowledge and information, make an information system that shall keep information about analysts and work groups working on each requirement through attributes stored along with them and requirements shall use attributes to store discussion threads and votes [s11].

To solve issues related to awareness: A best technique is to start projects with a kind of “boot camp”. For knowledge protection issues: a clear strategy regarding what knowledge needs to be protected must be established before distributed projects are launched, for example to protect intellectual property or personal data. For knowledge creation issues: the best technique suggests defining contact persons for the different knowledge areas in a project. They also recommended installing a knowledge management team [s12].

For issues related to knowledge mining, two kinds of questionnaire are designed to realize the Knowledge Mining of Offshore Software Development from experienced project managers in the side of both clients and vendors [s27].

Improve communication with the client in order to mitigate knowledge transfer risk [s18]. Compare the frequencies of the SPI barrier’s values against the occurrences of other SPI barrier’s values, the relative importance of each barrier can be identified by this [s25].

#### 4.7 Culture and Distance

Distance in Distributed Software Development affects a number of things like culture and time zone etc. A number of cultural risks may arise since participants do not necessarily share the same language, traditions, or organizational culture [s31]. All the techniques to manage Knowledge management and Awareness risks are shown in table 6.

Techniques	Literature	Sub Categories of Risks
• Use workflow management and online tools	S1	Distance & cultural clashes
• Do contextual training and planning	S2	Spatial and temporal distances
• Educate people about culture using soft skill training	S42, S46	Language difference and distance problems

Table6 : Mapping classification of risk management techniques and Culture and Distance risks

To manage risks related to culture and distance, Use workflow management and online tools by establishing open communication across multiple channels and having periodic workshops with teams and applying online team-building if visits are not feasible [s1]. Do training and planning to conquer risk related to culture. Contextual training like concepts related to communication, trust acquisition, and cultural differences, should be planned as a mandatory training in the organizations’ on boarding training program [s2].

Use series of goals, questions and metrics for the Cross Cultural Collaborative risk assessment. By using the suggested set of goals, questions and metrics from, a user can perform an assessment of cross-cultural risk in the requirements work of the procurement

project [s41]. The three dimensions of collaboration and work processes, cultural, and context are used as the dimensions of orientation [s41].

Educate people about culture using soft skill training. Soft skills training should include not just people management, but also should involve training on the local culture including, local way of working, values, history and food. Also periodic culture specific trainings can be provided to increase understanding, and to appreciate and respect other cultures and their differences [s42]. To assess cross cultural risk impact, quantitatively assess the degree of cross cultural risk impact on IT software development [s50].

#### 4.8 Project Delivery Failure

A standard risk for many projects, the risk of being late or over budget amplifies in probability and impact due to the intrinsic difficulties of managing a global development team [s1]. All the techniques to manage Project delivery failure risks are shown in table 7.

Techniques	Literature	Sub Categories of Risks
<ul style="list-style-type: none"> <li>• Implement and systematically follow quality gates at work product level</li> <li>• Establish and use of quality indicators within the project</li> <li>• Monitor and use early defect ratio as a warning sign of insufficient specification and code quality</li> </ul>	S1	Quality Decrease
<ul style="list-style-type: none"> <li>• Professionally train all project managers and applying best practices from the CMMI (DEV + ACQ) frameworks</li> </ul>	S1	Less Efficiency

Table7: Mapping classification of risk management techniques and Project delivery failure risks

Decreasing in quality, efficiency or reduced productivity are some of the factors involved in project delivery failure. To avoid the quality decrease, several techniques discussed are implementing and systematically following quality gates at work product level, establishing and using of quality indicators within the project, to monitor and use early defect ratio as a warning sign of insufficient specification and code quality [s1].

Professionally train all project managers and apply best practices from the CMMI (DEV + ACQ) frameworks and target maturity level 3. Furthermore, recommended techniques were maintaining an organization risk repository and emphasizing on the use of lessons learned and root cause analysis reports from previous projects to avoid repetition of problems [s1].

#### 4.8 Trusts

Members of a virtual team tend to be less productive due to feelings of isolation, to the poor socialization and socio-cultural differences. It causes lack of trust and for this reason

and can lead to more complex problems such as boycott of staff, less knowledge sharing between members etc [s36]. All the techniques to manage Project delivery failure risks are shown in table 8.

Techniques	Literature	Sub Categories of Risks
<ul style="list-style-type: none"> <li>• Use communication and collaboration technology</li> </ul>	S2, S45	Bad relationship

Table8: Mapping classification of risk management techniques and Trusts risks

To manage trust risk and improve relationship between members, contextual training related to communication, trust acquisition, and cultural differences, should be planned as a mandatory in the organizations' on boarding program and when possible, team members should use communication and collaboration technology in order to improve the relationship and trust between the distributed project team members [s2].

Compare the frequencies of the SPI barrier's values against the occurrences of other SPI barrier's values, the relative importance of each barrier can be identified [s25].

#### 4.9 Supplier and Customer

One frequent risk with third party suppliers is not meeting the expectations in terms of quality and delivery schedule. Also with Global Software Engineering supplier competition on a global market, external suppliers often start with rather low rates and once the projects are sufficiently large clients might be forced to lock-in with them due to progress of product development and knowledge transition. In the least we may have to face increasing cost inflation [s1]. All the techniques to manage Project delivery failure risks are shown in table 9.

Techniques	Literature	Sub Categories of Risks
<ul style="list-style-type: none"> <li>• Establish a fixed price contract scheme</li> <li>• Evolve towards a partner model with the supplier and training suppliers</li> <li>• Train supplier management on escalation procedures</li> <li>• Avoid the Service Level Agreement (SLA) hammer</li> </ul>	S1	Poor supplier services
<ul style="list-style-type: none"> <li>• Work with multiple partners and distributing critical knowledge</li> <li>• Establish common processes , tools and maintain back-up and recovery mechanisms</li> </ul>	S1	Lock-in threats

<ul style="list-style-type: none"> <li>• Engage multiple suppliers</li> <li>• Sign short-term contracts</li> <li>• Outsourcing standard IT services</li> </ul>	S14, S45	Supplier Power
----------------------------------------------------------------------------------------------------------------------------------------------------------------	----------	----------------

Table9: Mapping classification of risk management techniques and Supplier and Customer risks

To solve Poor supplier services, establish a fixed price contract scheme with agreed supplier management and escalation processes. It recommended evolving towards a partner model with the supplier and training suppliers on required processes. During the ramp-up period, carefully train supplier management on escalation procedures and your own required quality level. Also escalate carefully, step-wise and avoid the Service Level Agreement hammer as well as rigorously highlight insufficient quality and delays or lack of visibility [s1].

For risks related to Lock-in with supplier, mentioned approaches are: working with multiple partners and distributing critical knowledge, establishing common processes , tools and maintaining back-up and recovery mechanisms [s1].

Risk techniques on supplier that have too much power over the customer are engaging multiple suppliers, signing short-term contracts, outsourcing standard IT services for which there are many suppliers capable of delivering good services, and in sourcing highly specific assets [s14].

For risks management of several partners and suppliers, distribute stress on process phases, roles involved in them, communication channels and coordination's of Risk Management actions across several partners and suppliers.

#### 4.10 Miscellaneous Risks

Security risks, dynamic risks and production support risks which are not covered in upper categories are included in Miscellaneous Risks. For production support and security, a tactical approach considering application development and support management is presented [s23]. All the techniques to manage Project delivery failure risks are shown in table 10.

Techniques	Literature	Sub Categories of Risks
<ul style="list-style-type: none"> <li>• Application development and support management.</li> </ul>	S38	Production support and security
<ul style="list-style-type: none"> <li>• Agile practices</li> <li>• Use localization skills, Partnership management skills and Multi-culture management skills</li> </ul>	S6, S50	Dynamics risks

<ul style="list-style-type: none"> <li>• Build market analytical capabilities, Vendor management, centralized supervision over backup systems</li> </ul>		
----------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

Table10: Mapping classification of risk management techniques and Miscellaneous Risks

Application development is mainly a project-based activity, resulting in the timely completion of project deliverables at dominant service level. As there is no 24/7 communication during the execution of the project, security risks of application development are limited [s49].

For Information Systems Security (ISS) risks, use a set of guidance steps which is supported with a free open source software tool. The methodology has five phases: Context Study; Security Requirements Checklist; Threats Study; Identification of Security Objectives and Determination of Security Requirements [s23, s49].

Dynamics risks could be people related, process related, technical related or external related. Agile techniques are used to overcome these risks. For people related dynamics risks, use of Agile People Management Skills was recommended in which Localization skills, Partnership management skills and Multi-culture management skills are proposed [s6].

For process related dynamics risks, use of Agile Project and Process Infrastructure was proposed in which Modular approach, Decentralized knowledge management and Agile task planning are involved [s6].

For risks related to technical related dynamics risks, use of Agile IT Strategy and IT Infrastructure is presented in which techniques like loosely centralized IT strategy; Standardized IT platform and Comprehensive application infrastructure for communication and collaboration are included [s6].

Risks associated with external related dynamics risks were mitigated using Agile Environment Management Skills in which techniques like building market analytical capabilities, Vendor management, and centralized supervision over backup systems, recovery policy and delegation of legal compliance responsibility to local champions are included [s6].

## 5. Risk Management Framework

The nature of global software development brings forward new areas of concern that require careful attention from project managers. While practitioners lack standardized approaches to overcome these risks in distributed software development, various managers are experimenting and quickly adjusting their tactical approaches [29]. Distributed software development risks are not only different but can also lead to project failure if managed inadequately. A software risk management framework plays an important role in DSD as it identifies risks early in a project and takes action before they can do some harm to the project

[7]. Hence, a framework risk management should be used to minimize the impact of risks especially for DSD where risks are more co-located sites and can cause unknown damage to projects due to their nature.

The DSD Agile Risk Management framework described in this paper provides all the necessary preconditions for successful risk management. In addition, the provided DSD Agile Risk Management framework aims at knowledge and experience gathering and prevents loss to the project by experiences of previous projects [29]. In order to organize our findings from SLR, we propose DSD Agile Risk Management Framework based on Experience factory to facilitate users with risk management experience. The Experience Factory is an organizational framework whose objective is to produce, store, and reuse experiences gained in a software development organization [29]. Our DSD Agile Risk Management Framework has different functionality like risk description, its mitigation techniques, and risk treatment experiences and as we are using the previous knowledge through SLR, the experience factory concept best suited our DSD Agile Risk Management Framework idea.

Experience Factories are accepted but seldom used in practical life [29]. However, to inspire the reuse of our existing techniques and risks found in the SLR, we have combined it with practice of risk management. Experience Factories recognize that improving software processes and products requires: (1) continual accumulation of evaluated and synthesized experiences in experience packages; (2) storage of the experience packages in an integrated experience base accessible by different parts of the organization; and (3) creation of perspectives by which different parts of the organization can look at the same experience base in different ways [29].

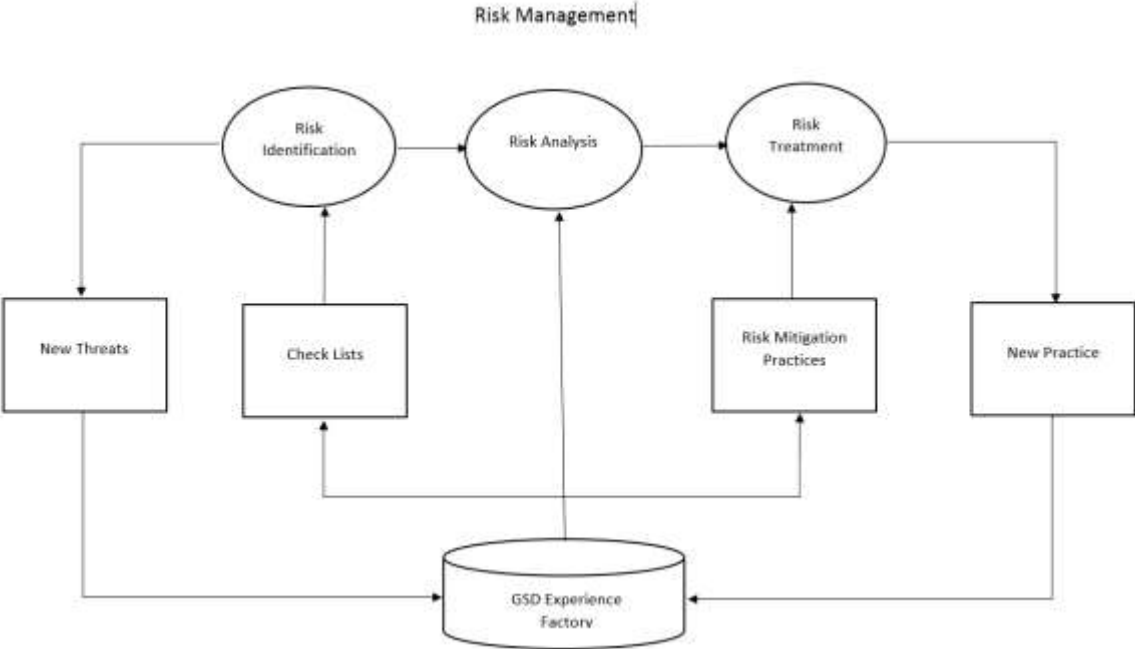


Figure 2: DSD Agile Risk Management Framework



Use of DSD Agile Risk Management Framework is dependent on the rules of the company in which it is to be used and also on the experience of project manager using DSD framework. To get better understanding of usage of DSD Agile Risk Management Framework, we have explained its terminology with the help of small example.

All of our techniques and risks found in SLR are collected, structured and stored in GSD Experience factory. After this they are made accessible and can be activated to help in the future. This, leads to new experiences, starting a new iteration of the lifecycle in DSD Experience factory [29]. From GSD experience factory, the next concept in our framework is checklist for risks. Checklist is a type of informational job aid used to reduce failure by compensating for potential limits of human memory and attention. It helps to ensure consistency and completeness in carrying out a task. Through the checklist one can identify if it is a risk or not and accordingly send it to the risk identification section. Example and usage of the checklist is given in Appendix A. From Checklist next step is to go to risk identification, where risks are identified. If the risk is not available in GSD experience factory, then through new threat, it is added back to GSD experience factory so that next time it is available in GSD experience factory. If risk is not new and already available in GSD experience factory, then next step involves analyzing the risk. In this step risks are analyzed, prioritized and threat level is calculated according to the need of the company. After analyzing comes the treatment of the risks. For risk treatment, it can be mitigated and treated accordingly with the help of techniques gathered in GSD Experience factory. If a risk is treated with a new technique that is not present in GSD experience factory then that new technique or practice is added back to GSD experience factory. Checklists, Risk identification, risk analyze and risk treatment is totally company dependent and may vary from company to company procedures and rules. Our DSD Agile Risk Management Framework aims to provide adequate organizational learning by adding knowledge to the GSD Experience Factory along with continuous improvement of risk management processes within the organization.

## *6. Evaluation*

After the described Experience based framework, we evaluated user acceptance of the experience lifecycle initiative. We did a case study and interviewed eight people out of three different projects to evaluate our DSD Agile Risk Management Framework, most of the interviewees were project leaders or Scrum masters. Out of these eight people, six people were interviewed from one project in which three people were from Pakistan where the project was being developed and tested whereas the remaining three people were interviewed from USA where the project requirement engineering was done. Both the remaining two projects are from Germany and we interviewed their coaches to evaluate our DSD Agile Risk Management Framework.

To evaluate our DSD Agile Risk Management framework, we wanted to know answer of following two questions:

Question 1: Are the techniques and risks in DSD Agile Risk Management Framework relevant to the techniques and risks that were used in your project?

Question 2: Will you try to use experiences in experience factory of our DSD Agile Risk Management Framework for your future projects?

To know the answer of these two questions, we had to know in detail about the projects we were dealing with and a lot of questions were asked which are given in Appendix B.

The case study revealed that 8 out of 8 people (100 %) found our experience factory framework useful. From these 8 people, only 7 (approx. 88%) people thought it useful to try new techniques mentioned in our framework.

To better understand about their projects, we asked them about detail description, techniques and risks involved in their project. We then discussed our framework to know about our techniques and risks available. In the end, we derived four main categories of risks involved in their projects: Communication issues, Scheduling and Collaboration issues, Task Distribution issues and issues directly related to Project Management. We will not be discussing these risks here as they have been explained briefly in the result section. To have a better understanding about the answers of the two questions asked above, we will explain all three projects and their view about our framework. Due to confidential reasons, we are not writing companies name and are represented as Project A, Project B and Project C.

### 5.1 Project A

The undertaken project falls in the category of scrum software development involving two countries – Pakistan and United States of America. Requirement engineering has been done at main center in USA, team in Pakistan does the rest, development and testing of given project. This project encompasses 65 people and its duration is around 8-10 months. Coming to scrum, number of sprints that will make up a complete project is 12 and development team is asked to provide each sprint as a deliverable after 3 sharp weeks. Number of people in each team varies depending on task requirement. When we questioned the Pakistani team about their reason of preferring scrum on other agile and clean room methodologies, they responded that their stakeholder company - in USA - wanted to have the view of the product as soon as possible and also it will then be easier for them – Pakistani team - to remove unwanted functions and bugs prototype - deliverable - wise rather than from full project at once.

On time project delivery, team management, conflict management, decision making (Allocation of tasks) and communicational issues have been the top ranked challenges in this project. Team management issues were resolved by establishing standard policies, rules and protocols beforehand in the beginning of every project. Team lead is a person well equipped with skills of conflict management as handling people is difficult than handling projects in our world. Organizing training programs for team members to get them better informed and skilled for processes and technologies organization was worked upon. Frequent visits between main center and development teams kept the morale high and people do get the feeling of self- actualization. Multiple communication channels or rather a good mix of communication channels were used like – Skype calls, messenger chats, e-mails, telephonic conversations, on site visits etc to enhance the communicational bandwidth up to the desired requirement.

After knowing about their techniques and risks, we gave a description about our framework and asked question number one: Are the techniques and risks in our experience factory relevant to the techniques and risks that were used in your project? Out of 6 people interviewed in this project, all six agreed that techniques and risks in our experience factory were more or less relevant to the techniques and risks that were used in their projects. All their risks were available in our framework, only one technique that was not similar to them was for communication challenge.

Asked if they are willing to use our framework in the future, five out of six people were interested in using knowledge blended in this framework and thought it useful to try different techniques mentioned in the framework for future projects.

## 5.2 Project B:

The undertaken project falls in the category of distributed software development involving two different cities of Germany. This project encompasses 14 people and its duration is around 9 weeks. Pair-programming was created locally and there was no distributed pair programming. They split down a story card (requirement) into 1..n tasks. Each task was worked on by one pair at a time, but pairs were also switched if required for longer tasks. When we questioned their Coach (instructor) about their reason of preferring extreme programming (XP) on other agile and clean room methodologies, he responded that he was interested in using agile and wanted to learn agile in distributed software development and experimented by choosing XP.

Collaboration between teams, scheduling, expertise problem, frustration and communicational issues have been the top ranked challenges in this project. Collaboration issues were resolved by installing supervisor in City B to minimize this problem. Scheduling problem was resolved by long term planning and support from the admin. Coach was the person well equipped with skills of conflict management or on expertise problem. Problem was solved by coaching, sharing of data and teams being divided into different pairs to cover other teams and by allocating experienced person on the task. Also some training programs for team members were organized to get them familiar with the project and methodology. Like the previous project, multiple communication channels were used like – Skype calls, messenger chats, e-mails, telephonic conversations, shared white board etc to enhance the communicational bandwidth up to the desired requirement.

Similarly like Project A, after knowing about Project B techniques and risks, we gave a description about our framework to the coach and asked the question again that are the techniques and risks in our experience factory relevant to the techniques and risks that were used in your project? Coach agreed that techniques and risks in our experience factory were relevant to the techniques and risks that were used in their projects. All their risks were available in our framework, only one technique that was not similar to them was for communication challenge.

Asked if they are willing to use our framework in the future, he was interested in knowing more about this framework and said he could try different techniques mentioned in the framework.

### 5.3 Project C:

The undertaken project falls in the category of Agile distributed software development for the product of a banking sector involving two countries – Germany and India. Requirement engineering and designing has been done at main center in Germany whereas team in India is involved in the development and testing of given product. This project encompasses 12 people and its duration is around 6 months. Varied numbers of people are working in each team depending upon task requirement. When we questioned their coach(project leader) about their reason of preferring agile on other methodologies, he responded that there was need of changes in requirements during project meaning they were unable to try waterfall method and had to rely on agile.

Coordination and scheduling, Requirement understanding and communicational issues have been the top ranked challenges in this project. Coordination and scheduling problem was solved by installing a middle person to coordinate between two sites. They established agile leadership hierarchy and coordination structure between sites. Also they initially used work break down approaches but it did not offer much help. One of the bigger problems in this project was requirement understanding in different sites as people in India were unable to understand what exactly was needed and were unable to understand requirements from the document. Frequent visits were done initially from India to Germany to understand about requirements but due to cost issues it was difficult to continue. More documentation was done and collaborative tools and languages were used to resolve this issue. Also some training programs were organized for team members to get them better informed and skilled for processes and requirements. For communication purpose, a middle man was favored to communicate with Indian people. Meetings were done through Skype or by video conferencing.

After knowing about their techniques and risks, we gave a description about our framework and asked our desired question number one that are the techniques and risks in our experience factory relevant to the techniques and risks that were used in your project? The coach agreed that techniques and risks in our experience factory were more or less relevant to the techniques and risks that were used in their projects. All their risks were available in our framework as well as the techniques to manage them. Asked if he was willing to use our framework in the future, coach replied that he will have a go through in detail about techniques as the techniques mentioned in our framework look more technical but he was interested in some techniques and said that it will be useful using knowledge blended in this framework for future projects.

## *7. Discussion about Evaluation*

Generally, project managers found our framework useful to use and were interested in using this framework in their future projects. With almost seven out of eight people were (approx. 88%) interested in using this framework, based on this we concluded that this framework is useful and can be used in different projects for risk management. As found during different

projects and described by Gerhard Fischer [30], we understood that experience based factory is an evolutionary process in which techniques and risks of DSD risk management are determined through an iterative process of collaboration among multiple project managers, rather than being completely specified before system development occurs. We focused on real projects and knowledge based integration of a framework with the principles of: (1) they must evolve because they cannot be completely designed prior to use, (2) they must evolve to some extent at the hands of the users.

We have worked on these two principles by evaluating our framework through three different distributed projects across the globe. As focused and published by Gerhard Fischer [30], The Seeding, Evolutionary Growth, Reseeding (SER) Process Model best describes our evaluation of knowledge based experienced factory. The SER model describes evolution in terms of three levels (artifact, domain, and project managers), three phases (seeding, evolutionary growth, and reseeded) and project managers or team leaders as input feeders. Figure 3 illustrates the interplay of those layers in the context of our SER model. The dashed arrow from the experience factory and Artifact level indicates that artifacts (techniques and risks) are built using domain knowledge obtained in the projects. The arrows (along the time dimension) indicate that artifacts are not simply designed at one time, but instead evolve over time. The solid arrows from artifact to experience factory level indicate that the design and evolution of artifacts (techniques and risks) produces new domain knowledge at the end.

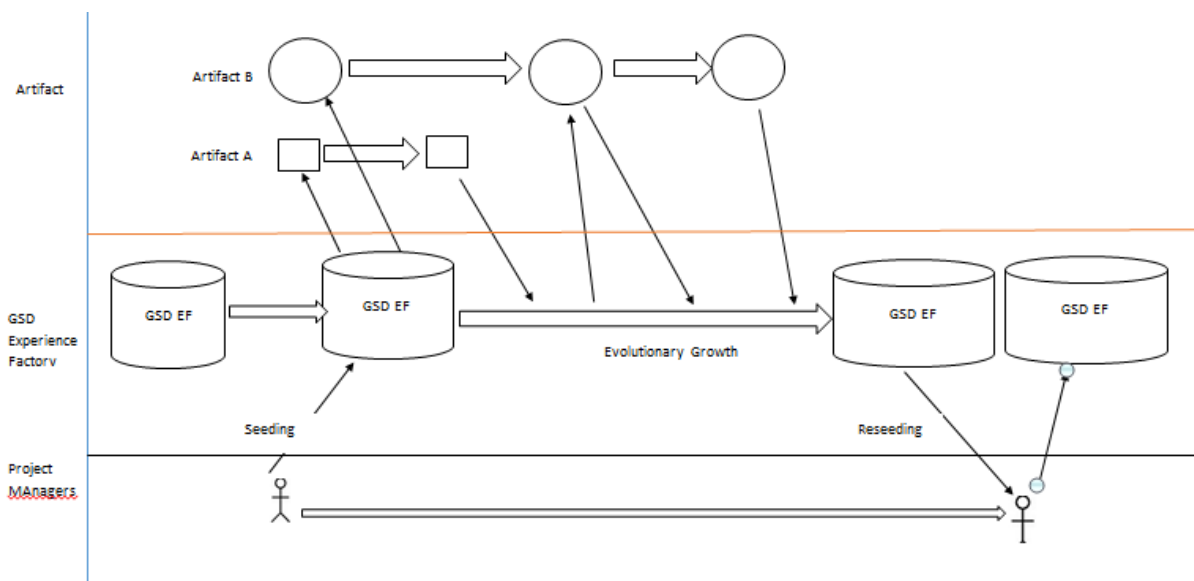


Figure 3: The SER Model: A process model for the development and evolution of experience factory [30]

The Experience Factory is an organizational infrastructure whose goal is to produce, store, and reuse experiences gained in a software development organization [31]. DSD Agile Risk Management framework organizes a software development enterprise into two distinct organizations, each specializing in its own primary goals. The Project Organization focuses on delivering the software product and the Experience Factory in DSD Agile Risk Management

framework focuses on learning from experience and improving software development practice in the organization. Although the roles of the Project Organization and the Experience Factory are separate, they interact to support each other's objectives [31]. The feedback between the two parts of the organization flows along well-defined channels for specific purposes, as shown in Figure 3.

From these three projects, we can conclude that DSD Agile Risk Management framework recognizes that improving distributed software processes and products requires: (1) continual accumulation of evaluated and synthesized experiences in experience packages; (2) storage of the experience packages in an integrated experience base accessible by different parts and people of the organization. From this we can evaluate about DSD Agile Risk Management Framework that it is a) valuable for individuals in an organization (part of individual learning) and b) also allows to share lessons and experience learnt by individuals throughout the organization (part of organizational learning) [s49].

The SEL Process Improvement Paradigm provides a practical method for facilitating product-based process improvement within a particular organization, based on effective use of that organization's own experience. Because it directly ties process improvement to the products produced, it allows an organization to optimize its process for the type of work that it does. Using this approach, the SEL has reduced development costs by 74%, decreased error rates by 85%, and increased reuse by over 300% over the past 15 years [31]. Due to limited time, we could only evaluate our DSD Agile Risk Management Framework and did not verify.

## *8. Conclusion and Outlook*

Number of studies in the area of managing risk in Distributed Software Development is on the rise and until recently no updated SLR has been done on this area. This paper aims to identify the techniques used to manage risks in Distributed Software Development and their related associated risks using SLR on studies from 2006 to 2013 and at the end provides the framework for risk management. The result of this SLR contributes to Risk Management in Distributed Software Development in several ways. Firstly, it can help those researchers and Software project managers, who are involved or work with the field of Distributed Software Development to gain knowledge about existing techniques to manage risks by placing all techniques in one place and diversifying related risks. Secondly, with knowledge based experience factory, it shows the further work in areas that opens opportunities for future researches. Total 50 primary studies out of 431 initial studies were reviewed and extracted techniques from them were categorized in different risk categories. We extracted risk areas, which were covered by identified techniques and categorized them in to 10 possible categories as: Project Management and Coordination, Requirements, Communication, Knowledge Management and Awareness, Task distribution, Culture and distance, Technical issues, Trust and contracts, Project delivery failure, Supplier and Customer, Miscellaneous Risks. We summarized all 10 possible risk categories and sub categories as well as number of literature containing the techniques per risk category as shown in result section.

In 2010, Da Silva et al [14] conducted a SLR to identify challenges and best practices to improve project management in Distributed Software Development. In our study, we use their

identified list of challenges to discuss more about the current study results. Figure 4 shows their top five identified challenges based on the frequency of occurrences of each category in their studies:

They found communication as the most challenging aspect of Distributed Software Development. With a small difference they found cultural difference as the second challenge and the coordination issues as the third one. They realized that overall, the most solutions were provided to overcome the communication challenges and based on evidence, and they believed that emphasis should also be given to cultural difference and coordination.

In current study, we found most of the techniques were provided for the area of project management and coordination with 30% of studies. Techniques for risks in requirements were covered by 18% of studies showing big difference with the techniques provided for challenges related to project management and coordination. Task distribution was covered by 11% of the studies highlighting the importance of this issue and with small difference were communication as well as knowledge management and awareness, covered by 9% of the studies. 7% of studies were about the culture and distance concern, while 4% of the studies covered risks related to trust and supplier and customers and least number of studies belonged to project delivery failure solutions. Finally, we had Miscellaneous Risks with 6% of the studies covering it.

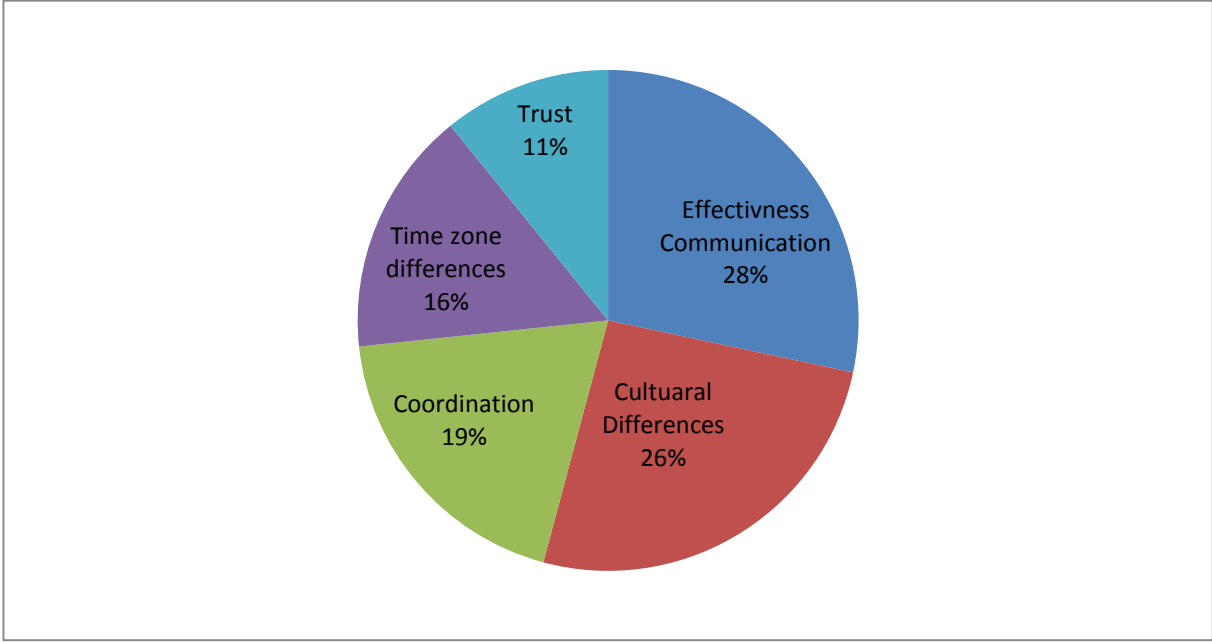


Figure 4: Challenges of Project Management of Distributed Software Development

Figure 5 shows the risk and percentage of studies covering the techniques for managing risks.

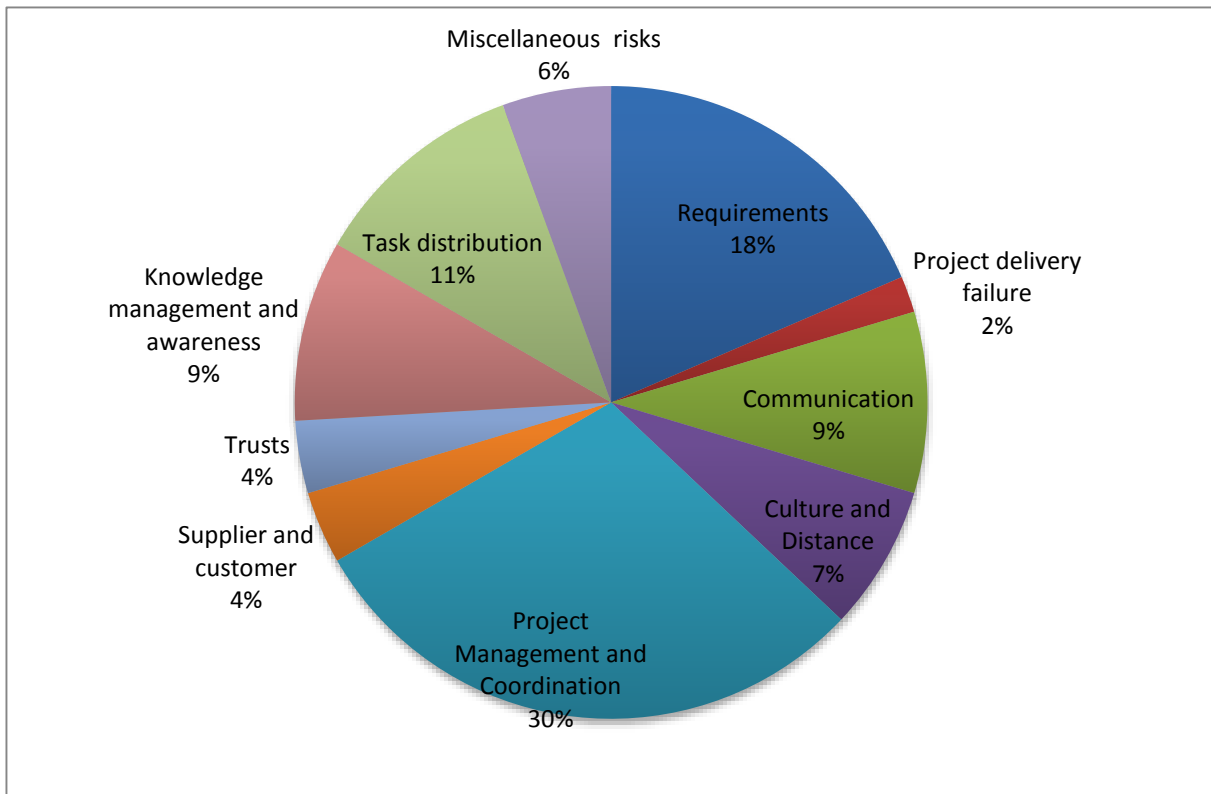


Figure 5: Risks areas found in this SLR

Unlike [14], as it is shown in figure 5, our study found that more provided techniques are related to project management and coordination and communication is in fourth place with knowledge management whereas communication was first challenges with most practices in their challenges list and it shows big difference between our findings and [14] results.

Furthermore, we realized that requirements is a critical phase in Distributed Software Development as [s11] says that the rise of new development paradigms such as Global Software Development forces requirements engineering to face up to new challenges and risks are not common in traditional development models. When an organization first embarks upon a Distributed Software Development project it exposes itself to plenty of risks. Requirements engineering has the second place in our study whereas it is not identified by [14] as a separate challenge. Cultural and trust are among the [14] top challenges whereas they are not among our five top risks with most provided techniques and it shows need of more works in this areas. Additionally, Task distribution/allocation and knowledge management are among our five top risks with most provided techniques while they are not among [14] top five challenges.

Even after finding technique's and risks associated with distributed software development, we were unable to find lot of frameworks to support distributed risk management and only limited number of given frameworks were validated or evaluated through industry so we presented our own knowledge based DSD Agile Risk Management framework to ease the work load on project managers to be aware at operational level to identify risks at the higher levels and he/she could plan response actions for these risks. We evaluated our DSD Agile Risk





Software Development, and it can be an interesting area for further research as during case study we found more techniques related to these projects than being mentioned in the literature. Based on current study we can say that research in risk management in Distributed Software Development is still in its early stages and requires maturation and further work. Also we have evaluated our DSD Agile Risk Management framework due to limited time scale, it would be better in future to verify and validate this framework by using it in DSD projects.

## 10. References

1. A Research Agenda for Distributed Software Development by Bikram Sengupta, auz Khas, Satish Chandra, Vibha Sinha
2. Jiménez, M. et al., Challenges and Improvements in Distributed Software Development: A Systematic Review. *Advances in Software Engineering*, 2009
3. Risk Management for Web and Distributed Software Development Projects by Ayad Ali Keshlaf and Steve Riddle. *The Fifth International Conference on Internet Monitoring and Protection*.
4. R. Williams, G. Pandelios, and S. Behrens, "Software Risk Evaluation (SRE) Method Description (Version 2.0)," *Software Engineering Institute (SEI)* December 1999.
5. B. W. Boehm, *Software Risk Management: IEEE Computer Society Press*, 1989.
6. B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," *Tech. Rep. EBSE-2007-001*, Keele University & Durham University Joint Report, Staffordshire, UK, 2007.
7. Identifying Software Project Risks: An International Delphi Study by Roy Schmidt, Kalle Lyytinen, Mark Keil and Paul Cule. *Journal of Management information system* Volume 17 Issue 4, Number 4/Spring 2001.
8. A. Keshlaf and K. Hashim, "A Model and Prototype Tool to Manage Software Risks," *Proc. of First Asia-Pacific Conference on Quality Software*, IEEE Computer Society, 2000, pp. 297–305.
9. F. W. McFarlan, "Portfolio approach to information systems," *Harv. Bus. Rev.*, vol. 59, no. 5, pp. 142–150, 1981.
10. Challenges and Improvements in Distributed Software Development: A Systematic Review by Miguel Jimenez, Mario Piattini and Aurora Vizcaíno.
11. Lamsweerde, Van A. *Requirements Engineering: From System Goals to UML Models to Software Specifications*, Wiley, January 2009.
12. Global software development by JD Herbsleb, Deependra Moitra, - *Software*, IEEE, 2001 - [ieeexplore.ieee.org](http://ieeexplore.ieee.org)
13. Meeting the Challenge of Communication in Offshore Software Development By Henrik Munkebo Christiansen NNIT A/S, Lottenborgvej 24, 2800 Lyngby, Denmark

14. Challenges and Solutions in Distributed Software Development Project Management: A Systematic Literature Review ,da Silva, F.Q.B.; Costa, C.; França, A.C.C.; Prikladinicki, R.; Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on
15. Mohtashami, M., et al., Risk Management for Collaborative Software, Development. Information Systems Management, 2006. 23(4): p. 20 - 30.
16. Kliem, R., Managing the Risks of Offshore it Development Projects. EDPACS, 2004. 32(4): p. 12 - 20.
17. J. Kontio, "The Riskit Method for Software Risk Management, Version 1.00 CS-TR-3782 / UMIACS-TR- 97-38," University of Maryland, Maryland 1997.
18. Kontio j. and B. V. R., "Empirical Evaluation of a Risk Management Method," in SEI Conference on Risk Management Atlantic City, Nj, USA, 1997
19. B. Freimut, S. Hartkopf, P. Kaiser, J. Kontio, and W. Kobitzsch, "An Industrial Case Study of Implementing Software Risk Management," Proc. of the 8th European Software Engineering Conference held jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering Vienna, Austria, ACM, 2001, pp. 277- 287.
20. J. Dhlamini, I. Nhamu, and A. Kachepa, "Intelligent Risk Management, Tools for Software Development," Proc. of (SACLA 09) Mpekwani , Beach Resort, South Africa: ACM, 2009, pp. 33 - 40.
21. D. Callegari and R. Bastos, "Project Management and Software Development Processes: Integrating RUP and PMBOK," Proc. of the 2007 International Conference on Systems Engineering and Modeling, IEEE, 2007, pp. 1- 8.
22. W. R. Duncan, "A Guide to the Project Management Body of Knowledge PMBOK- PMI," Project management Institute, Boulevard -Newtown Square, USA 1996.
23. L. Briand, K. El Emam, and F. Bomarius, "COBRA: A hybrid method for software cost estimation, benchmarking, and risk assessment", International Conference on Software Engineering, ICSE 1998, pp. 390-399
24. F. Ferrucci, C. Gravino, and S. Di Martino, "Estimating Web Application Development Effort Using Web-COBRA and COSMIC: An Empirical Study", EUROMICRO-SEAA 2009, pp. 306-312
25. Basili, V., Caldiera, G., Rombach, D. 1994. Goal/question/metric paradigm. In Encyclopedia of Software Engineering. Vol. 1, J. C. Marciniak, Ed. John Wiley and Sons, New York, 528-532
26. Edwards, H. K, Kim, J. H., Park, S., and Al-Ani, B. 2008. Global software development: Project decomposition and task allocation. International Conference on Business and Information, BAI '08. Academy of Taiwan Information Systems Research, ISSN: 1729-9322.

27. Lamersdorf, A. Münch, J., and Rombach, D. 2009. A survey on the state of the practice in distributed software development: Criteria for task allocation. Proceedings of the International Conference on Global Software Engineering, ICGSE '09. 41-50.
28. Yin, R. K. "Case study research: design and methods", Sage, USA, 1994.
29. Basili, Victor R. The Experience Factory and its Relationship to Other Improvement Paradigms. 4th European Software Engineering Conference (ESEC), Garmish-Partenkirchen, Germany. The Proceedings appeared as the Springer-Verlag Lecture Notes in Computer Sciences Series 717, September 1993.
30. Seeding, Evolutionary Growth, and Reseeding: Constructing, Capturing, and Evolving Knowledge in Domain-Oriented Design Environments by Gerhard Fischer: Center for LifeLong Learning and Design (L3D), Department of Computer Science and Institute of Cognitive Science.
31. De Mendonça Neto, Manoel Gomes, Carolyn B. Seaman, Victor R. Basili, and Yong-Mi Kim. "A Prototype Experience Management System for a Software Consulting Organization." In SEKE, pp. 29-36. 2001.
32. Meyer, Sebastian, Anna Averbakh, Torsten Ronneberger, and Kurt Schneider. "Experiences from establishing knowledge management in a joint research project." In Product-Focused Software Process Improvement, pp. 233-247. Springer Berlin Heidelberg, 2012.

## *11. Primary Studies.*

S1 Managing risks in global software engineering: principles and practices, Ebert, C.; Murthy, B.K.; Jha, N.N.; on Global Software Engineering, 2008 - computer.org

S2 Improving Contextual Skills in Global Software Engineering: A Corporate Training Experience, R Prikladnicki, L Pilatti - Global Software Engineering, 2008. ..., 2008 - ieeexplore.ieee.org

S3 A Rule-Based Model for Customized Risk Identification in Distributed Software Development Projects, Lamersdorf, A. ; Munch, J. ; del Viso Torre, A.F. ; Sá ; nchez, C.R. ; Heinz, M. ; Rombach, D. ; on Global Software ..., 2010 - computer.org

S4 Goal and Risk Factors in Offshore Outsourced Software Development from Vendor's Viewpoint, Islam, S. ; Joarder, M.M.A. ; Houmb, S.H. ; Conference on Global Software Engineering 2009 - computer.org

S5 Risk assessment on distributed software projects, Adailton Magalhães Lima, Proceeding ICSE '10 Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2

S6 A New Perspective on GDSD Risk Management: Agile Risk Management, Manage. Sci. & Inf. Syst., Univ. of Massachusetts Boston, Boston, MA, USA, Global Software Engineering (ICGSE), 2010 5th IEEE International Conference.

S7 Risk and Compliance Management Framework for Outsourced Global Software Development , Magnusson, C. ; Sung-Chun Chou ; Global Software Engineering (ICGSE), 2010 5th IEEE International Conference.

S8 A Coordination Risk Analysis Method for Multi-site Projects: Experience Report ,Bass, M. ; Herbsleb, J.D. ; Lescher, C. ; Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference.

S9 An Empirical Approach for the Assessment of Scheduling Risk in a Large Globally Distributed Industrial Software Project,Avritzer, A. ; Lima, A. ; Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference.

S10 Project Outcome Predictions: Risk Barometer Based on Historical Data, Smite, D. ; Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference.

S11 Risks and Safeguards for the Requirements Engineering Process in Global Software Development, Lopez, A. ; Nicolas, J. ; Toval, A. ; Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference.

S12 Knowledge Transfer in IT Offshore Outsourcing Projects: An Analysis of the Current State and Best Practices, Betz, S. ; Oberweis, A. ; Stephan, R. ; Global Software Engineering (ICGSE), 2010 5th IEEE International Conference.

S13 Collaborative Software Engineering with a Digital Ecosystem, Thomas Heistracher ; Thomas Kurz ; Giulio Marcon ; Claudius Masuch ; Global Software Engineering, 2006. ICGSE '06. International Conference.

S14 A review of the IT outsourcing literature: Insights for practice , Mary C. Lacity, , Shaji A. Khana and Leslie P. Willcocksb , The Journal of Strategic Information Systems Volume 18, Issue 3, September 2009, Pages 130-146

S15 Towards a Multi-criteria Development Distribution Model: An Analysis of Existing Task Distribution Approaches, Lamersdorf, A. ; Munch, J. ; Rombach, D. ; Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference.

S16 Estimating the Effort Overhead in Global Software Development, Lamersdorf, A. ; Mü ; nch, J. ; Torre, A.F.V. ; Sá ; nchez, C.R. ; Rombach, D. ; Global Software Engineering (ICGSE), 2010 5th IEEE International Conference.

S17 An empirical study of groupware support for distributed software architecture evaluation process, Muhammad Ali Babar, , Barbara Kitchenham, Liming Zhu, Ian Gorton , Ross Jeffery, Journal of Systems and Software, Volume 79, Issue 7, July 2006, Pages 912-925

S18 Risks in offshore IT outsourcing: A service provider perspective, Madhuchhanda Das Aundhe, Saji K. Mathew, European Management Journal, Volume 27, Issue 6, December 2009, Pages 418-428

S19 New Opportunities Presented by Novel Work Breakdown Techniques for Distributed Software Development , Mohan, S. ; Fernandez, J. ; Global Software Engineering (ICGSE), 2010 5th IEEE International Conference.

S20 Knowledge Transfer in Global Software Development - Leveraging Ontologies, Tools and Assessments, Salger, F. ; Sauer, S. ; Engels, G. ; Baumann, A. ; Global Software Engineering (ICGSE), 2010 5th IEEE International Conference.

S21 IT Application Assessment Model for Global Software Development, Ravishankar Kuni ; Navneet Bhushan ; Global Software Engineering, 2006. ICGSE '06. International Conference.

S22 A Taxonomy and Visual Notation for Modeling Globally Distributed Requirements Engineering Projects, Laurent, P. ; Mä ; der, P. ; Cleland-Huang, J. ; Steele, A. ; Global Software Engineering (ICGSE), 2010 5th IEEE International Conference.

S23 Risk Management for Web and Distributed Software Development Projects, Keshlaf, A.A. ; Riddle, S. ; Internet Monitoring and Protection (ICIMP), 2010 Fifth International Conference.

S24 DRiMaP--A Model of Distributed Risk Management Process , Kajko-Mattsson, M. ; Sjokvist, K. ; Soderstrom, J. ; Krogdahl, D. ; INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference.

S25 Software Process Improvement barriers: A cross-cultural comparison, Mahmood Niazia, b, , Muhammad Ali Babarc, , June M. Verner , Information and Software Technology, Volume 52, Issue 11, November 2010, Pages 1204-1216

S26 A Practical Management and Engineering Approach to Offshore Collaboration, Cusick, J.; Alpana Prasad; Software, IEEE ,2006

S27 Knowledge Mining for Risk Analysis of Offshore Software Development, Zhongqi Sheng ; Moriyasu, T. ; Tsuji, H. ; Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference.

S28 Distributed software development projects: work breakdown approaches to overcome key coordination challenges, Sanjay Mohan, Jude Fernandez, ISEC '10 Proceedings of the 3rd India software engineering conference

S29 Risk Identification and Mitigation Processes for Using Scrum in Global Software Development: A Conceptual Framework , Hossain, E. ; Babar, M.A. ; Hye-young Paik ; Verner, J. ; Software Engineering Conference, 2009. APSEC '09. Asia-Pacific

S30 Systems engineering of complex software systems , Huen, W.H. ; Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE '07. 37th Annual

S31 Managing Risks in Distributed Software Projects:An Integrative Framework by John Stouby Persson, Lars Mathiassen, Member, IEEE, Jesper Boeg, Thomas Stenskrög Madsen, and Flemming Steinson in IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, VOL. 56, NO. 3, AUGUST 2009

S32 A Method to Evaluate the Suitability of Requirements Specifications for Offshore Projects , Sven Overhage, Oliver Skroch and Klaus Turowski, Business & Information Systems Engineering , Volume 2, Number 3, 155-164, DOI: 10.1007/s12599-010-0106-8

S33 A multi-criteria distribution model for global software development projects , Ansgar Lamersdorf and Jürgen Münch, From the issue entitled "Global Software Engineering". Journal of the Brazilian Computer Society , Volume 16, Number 2, 97-115, DOI: 10.1007/s13173-010-0010-6

S34 A US Client's learning from outsourcing IT work offshore , Joseph W. Rottman and Mary C. Lacity, From the issue entitled "Special Issue on Information Systems (IS) Sourcing, Guest



Editors: Rudy Hirschheim, Jens Dibbern and Armin Heinzl", Information Systems Frontiers, Volume 10, Number 2, 259-275, DOI: 10.1007/s10796-007-9061-4

S35 The Good and Bad of Agile Offshore Development, Cottmeyer, , M.; Agile, 2008. AGILE '08. Conference Digital Object Identifier: 10.1109/Agile.2008.18 , Publication Year: 2008 , Page(s): 362 - 367

S36 Challenges and improvements in distributed software development: a systematic review, Miguel Jiménez Alhambra-Eidos, Technology Innovation Center, Albacete, Advances in Software Engineering archive Volume 2009, January 2009

S37 Architecture leadership and management in globally distributed software development, Raghvinder S. Sangwan, Johannes Ros, LMSA '08 Proceedings of the first international workshop on Leadership and management in software architecture

S38 A Practical Management and Engineering Approach to Offshore Collaboration , Cusick, J.; Alpana Prasad; Software, IEEE

S39 End-to-end features as meta-entities for enabling coordination in geographically distributed software development, Marcelo Cataldo, James D. Herbsleb, SDG '09 Proceedings of the 2009 ICSE Workshop on Software Development Governance

S40 Global Software Engineering: The Future of Socio-technical Coordination, James D. Herbsleb Carnegie Mellon University, FOSE '07 2007 Future of Software Engineering

S41 Cross-cultural collaboration in ICT procurement, V Dalberg, E Angelvik, DR Elvekrok, Asle Kristian Fossberg; GSD '06 Proceedings of the 2006 international workshop on Global software development for the practitioner

S42 Cultural differences in software engineering, Lavanya R. Abraham Tata Consultancy Services, Geneva, Switzerland, ISEC '09 Proceedings of the 2nd India software engineering conference

S43 Risk management with enhanced tracing of requirements rationale in highly distributed projects, Matthias Heindl, Stefan Biffel, GSD '06 Proceedings of the 2006 international workshop on Global software development for the practitioner

S44 Cross-Cultural Risk Assessment Model, Wattanapokasin, W.; Rivepiboon, W.; 2009 International Conference on Signal Processing Systems

S45 Solutions in global software engineering: A systematic literature review by Stefan Schneidera, Richard Torkarb, and Tony Gorschek in International Journal of Information Management xxx (2012) xxx– xxx.

S46 Bugayenko, Yegor. "Competitive Risk Identification Method for Distributed Teams." In Software Engineering Approaches for Offshore and Outsourced Development: Third International Conference: Proceedings: SEAFOOD 2009, Zurich, Switzerland, July 2-3, 2009, vol. 35, p. 96. Springer, 2009.

S47 Verner, J. M., O. P. Brereton, B. A. Kitchenham, M. Turner, and M. Niazi. "Risks and risk mitigation in global software development: A tertiary study." *Information and Software Technology* 56, no. 1 (2014): 54-78.

S48 Basharat, Iqra, Tayyaba Nafees, and Muhammad Abbas. "Risks factors identification and assessment in virtual projects of software industry: A survey study." In *Science and Information Conference (SAI)*, 2013, pp. 176-181. IEEE, 2013.

S49 Betz, Stefanie, Susan Hickl, and Andreas Oberweis. "Risk Management in Global Software Development Process Planning." In *Software Engineering and Advanced Applications (SEAA)*, 2011 37th EUROMICRO Conference on, pp. 357-361. IEEE, 2011.

S50 Khan, Arif Ali, Shuib Basri, and P. D. D. Dominic. "A propose framework for requirement Change Management in Global Software Development." In *Computer & Information Science (ICCIS)*, 2012 International Conference on, vol. 2, pp. 944-947. IEEE, 2012.

## 12. Appendix

### Appendix A: Checklist for risks

Name	Done	Info	Priority	Time Left	Status	Complete
<b>Category: 1: Evaluating Resource Risks [ My Tasks ]</b>						
1.1 Organization	<input type="checkbox"/>		Normal	2495d 1h 43m	Created	0%
<ul style="list-style-type: none"> <li>- Check if project resources are sufficient (management, testers, QA, and other engaged parties)</li> <li>- Estimate the project scope and find out if the project is the largest one for the organization</li> <li>- Check if there is a well-determined process for software engineering and company management</li> </ul>						
1.2 Financing	<input type="checkbox"/>		Normal	2495d 1h 43m	Created	0%
<ul style="list-style-type: none"> <li>- Check if the funding is adequate and sufficient to complete project</li> <li>- Allocate funding for training and mentoring</li> <li>- Check budget limitations that can deliver the project at a fixed cost or be subject to cancellation</li> <li>- Assess project costs accurately</li> </ul>						
1.3 People	<input type="checkbox"/>		Normal	2495d 1h 43m	Created	0%
<ul style="list-style-type: none"> <li>- Ensure there are enough employees available in the project</li> <li>- Check if these employees have appropriate skills and experience</li> <li>- Check if employees worked together before</li> <li>- Estimate employees' belief in project succeeding and take measure to strengthen it</li> <li>- Check participation of customer and user representatives for product reviews</li> <li>- Check participation of crucial experts in the project</li> </ul>						
1.4 Time	<input type="checkbox"/>		Normal	2495d 1h 43m	Created	0%
<ul style="list-style-type: none"> <li>- Check if project schedule is realistic</li> <li>- Check the functionality to be scope-managed to meet project schedules</li> <li>- Estimate how the delivery date is critical</li> </ul>						
<b>Category: 2: Evaluating Business Risks [ My Tasks ]</b>						
2.1 Estimate potential consequences if a competitor reaches the market first	<input type="checkbox"/>		Normal	2495d 1h 43m	Created	0%
2.2 Evaluate jeopardy of project funding	<input type="checkbox"/>		Normal	2495d 1h 43m	Created	0%
2.3 End user to assess sufficient product features	<input type="checkbox"/>		Normal	2495d 1h 43m	Created	0%

My Tasks	13	13
1. Evaluating Resource Risks	4	4
2. Evaluating Business Risks	6	6
3. Evaluating Technical Risks	3	3

Customization

Columns

Actual Time

## Appendix B: Questions asked for evaluating our DSD Agile Risk Management Framework

- 1) What is project about?
- 2) How many people were involved in this project?
- 3) What is the duration of the project?
- 4) Is going distributed difficult?
- 5) Why did you choose agile as methodology?
- 6) What are the challenges faced during the project?
- 7) What were the techniques used to counter these challenges?
- 8) Is it important to mitigate every challenge/risk or some risks can be ignored?
- 9) Are the techniques and risks in DSD Agile Risk Management Framework relevant to the techniques and risks that were used in your project?
- 10) Will you try to use experiences in experience factory of our DSD Agile Risk Management Framework for your future projects?
- 11) Do you find DSD Agile Risk Management Framework useful for future projects?