Chapter 1

Introduction

The methodology that has dominated software development, for decades, is called ‘waterfall’, which was a term coined by Winston Royce in 1970. Most evolved form of approach supports a serial method of development and management of software projects through five stages of requirement analysis and specification, design, implementation, verification and maintenance (Serena, 2007). The essence of waterfall software development is that complex software systems can be built in a sequential, phase-wise manner where all of the requirements are gathered at the beginning, all of the design is completed next, and finally the master design is implemented into production quality software. This approach holds that complex systems can be built in a single pass, without going back and revisiting requirements or design ideas in light of changing business or technology conditions.

Under the waterfall approach, traditional IT managers laid down plans for detailed tasks and dependencies for each member of the development group, months or years down the line. It assumes that up-front planning is enough to take into account the variables that could impact the development process. Although, the managers put copious efforts in detailing every possible risk, mitigation plan and contingency, but it is not possible to predict any and all the variables that could possibly affect a software project. As a result, only 9% to 16% of software projects were considered on-time and on-budget. The inherent uncertainty and complexity in all the software projects requires an adaptive development plan to cope up with uncertainty and high number of unknown variables (Victor, 2004). Hence, ‘iterative and incremental’ approach of software development was being used, in which, the phases of development were revisited in order to drive the project towards success.

Modern software organizations are working under tight time and cost constraints. Development of software occurs in highly volatile environment due to the changes in the product requirement, business and market needs. Hence, many organizations are adopting agile methodology for software development as it is able to deliver products closer to customer requirements and faster
than the traditional method (Kähkönen & Abrahamsson, 2003; Abrahamsson, Salo, Ronkainen & Warsta, 2002). Agile approach is iterative and advocates small teams working together with stakeholders to define quick prototypes, proof of concepts, or other visual means to describe the problem to be solved.

Along with this, there has been a steady, irreversible trend toward the globalization of business, and of software-intensive high-technology businesses in particular. Software industry has started relocating its production units in decentralized zones in order to take the advantage of skilled workforce. This type of development of software which allows the team members to be located in various remote sites during the software lifecycle and thus making up a network of distant sub-teams is called Distributed Software Development (DSD) (Jiménez, Piattini, & Vizcaíno, 2009). The number of organizations distributing their software development processes worldwide is constantly increasing. Distributed Software Development (DSD) is gaining popularity as it helps in saving cost and reduces the time to market.

There is a growing interest in applying agile practices in DSD projects to leverage the combined advantages of both the approaches. Hence, many organizations are using distributed agile development (DAD), in order to build low-cost solutions to cater to the changing needs of the businesses. Although, distributed agile development has attracted attention due its potential benefits like shorter time to market, reduced development cost, and managing late requirements’ changes, there are significant challenges involved.

Distributed software development and agile methods differ significantly in their key tenets, which causes various risks to occur, which need attention. Agile principles revolve around the importance of communication and collaboration in software development. These methods are well suited when customers and developers are collocated and there is frequent interaction amongst them (Boehm & Turner, 2005). On the other hand, DSD is characterized by distance, time zone and cultural differences, which, in turn, gives rise to problems related to communication and collaboration. Hence, incorporation of agile in distributed software development becomes a threat to the successful delivery of software solutions.
There are few studies which have reported success in the application of DAD for developing solutions (Young & Terashima, 2008; Danait, 2005), while a relatively more number of studies discuss about the risks and challenges involved (Jalali & Wohlin, 2010; Hossain, Babar, Paik & Verner, 2009; Mudumba & Lee, 2010). Various case studies explore the issues and challenges related to distributed agile development (DAD), but they discuss issues which are related to individual context only. Moreover, these studies are scattered and hence the community is unable to get an overall picture of the challenges faced during distributed agile development (Simon, 2002; Fowler, 2006; Mattsson, Azizyan & Magarian, 2010). Hence, there is a need to perform a thorough analysis of the practices being used in various organizational settings for DAD implementation (Jalali & Wohlin, 2010) and consolidate the same, which will be useful for effectively managing DAD projects.

This research study aims to identify and evaluate the most important risks which impact DAD projects and the frequently used risk resolution techniques for controlling them. This involves the consolidation of the knowledge from the contemporary literature, exploratory research and quantitative descriptive research study, for the creation of a risk management framework for DAD projects. The study has a focus on Scrum and Extreme programming (XP) agile methods due to their acceptability in the industry. The research will be relevant for the academic discipline of project management, practicing project managers who are handling distributed agile projects as well as the distributed agile development community.

In the next section, we discuss the motivation behind this research in which we discuss about the reasons for taking up this research work. This is followed by broad research objectives addressed by this work in section 1.2. In section 1.3 we describe, in brief, the research process used and the research findings in section 1.4. The research contribution is presented in section 1.5 and the organization of the thesis is given in section 1.6.

1.1 Motivation for Research
Combining DSD with agile is gaining strength as it helps the organizations to get the benefits of both the growing trends (Ramesh, Cao, Mohan & Xu, 2006). Software organizations are able to build good quality solutions in lesser time and cheaper price.
Although distributed agile is able to deliver products close to customer requirements and faster than the traditional method (Kähkönen & Abrahamsson, 2003), there are many risks and challenges involved. Firstly, distributed development goes against the notion of physical proximity while agile teams, rely on intense person to person communication, both with team members and the customer. Secondly, most offshore organizations favor a plan-driven approach where detailed requirements or designs are sent offshore to be constructed (Fowler, 2006). On the other hand agile technique allows the developers to forego the detailed documentation as it encourages more face-to-face communication (Bowen & Maurer, 2002). These methods focus on establishing close collaboration between customers and developers, and delivering software within time and budget constraints. Thus, these two trends i.e. distributed development and agile approach, face difficulty when it comes to compatibility issues.

Most of the agile methodologies (e.g., scrum) assume that the team is located in a single room. Unfortunately, this principle does not fit in the real scenario where agile teams are also distributed across the geographical locations. A survey conducted by VersionOne, states that organizations are constantly scaling agile beyond single team and single project (VersionOne, 2013). According to an earlier survey by VersionOne in 2008, 57% of respondents had stated that their teams were distributed (VersionOne, 2008). Further, 41% of respondents stated that they were currently using or plan to combine agile with outsourced development (Miller, 2008). These facts clearly show that the current requirement of software industry is not in line with the agile concept of the entire agile team working in a single room. Thus, there is a need to extend the agile practices to distributed software development.

A recent survey by Scott Ambler on scaling agile shows that, greater is the level of geographic distribution, greater is the risk due to communication and coordination challenges, resulting in lower success rate (Ambler, 2012\textsubscript{a}). Another survey result shows that 60% of co-located agile project are successful, while roughly 25% can be considered as failed projects. On the other hand, although, more than 50% distributed agile projects have been successful, but 50% of them have failed too (Ambler, 2012\textsubscript{b}).
Higher failure rate of projects using DSD in an agile environment is indicative of the extent of difficulty and associated risks in executing them. Thus, risk management becomes a critical for the success of DAD projects (Prikladnicki & Yamaguti, 2004).

Research literature reveals the scarcity of work in the area of risk management for distributed agile projects (Jalali & Wohlin, 2010). Although, previous case studies have shown that agile methods, such as scrum can be easily and successfully customized to distributed projects (Simon, 2002; Fowler 2006), there is a paucity of work on the risks in distributed agile development (Paasivaara, Durasiewicz & Lassenius, 2009). None of the agile approaches suggest any risk management taxonomy, nor do they suggest any systematic way of describing and classifying risks (Nyfjord & Mattsson, 2007). Authors asserted that neither of the majority traditional risk management frameworks except IEEE 1540 and SEI -Software Risk Evaluation nor agile models provide any template for collecting, structuring and communicating risk related information. They suggested that agile should learn to integrate traditional risk management practices in order to ensure effective risk management. Current literature does not clearly address the key risks due to distributed agile development and how they can be dealt with (Hossain et al., 2009; Mattsson et al., 2010).

Some work has been done on risk management in distributed development environment but does not take agile development in account. A pioneering work of Prikladnicki, Audy, Nicholas, & Evaristo (2006) proposes a reference model for Global Software Development (GSD), wherein, the processes and challenges in various phases in GSD have been addressed, but the work does not considers agile environment. Another significant doctoral work by Persson (2009) uses literature-based and case study approach to develop a framework for risk management in distributed software development. This study is limited to North American and European research and practice and is applicable to traditional development approach only. Mudumba & Lee (2010) acknowledged the multiplicity in the nature of the distributed development projects caused by multi-locations, multi-cultures, multi-groups, multi-standards, and multi-technologies. This multiplicity leads to more dynamic risks as the existing risks change and new risks arise. Authors considered the dynamism in risks related to internal elements (people, process and technology) and external elements (project environment) and accordingly, suggested the risk
mitigation techniques. This work addresses the risks at abstract level and does not validate the suggested framework. It also proposes agile project management methods to control the DSD risks, but the risks of distributed agile development still remain unaddressed.

Specifically, in the area of distributed agile development, a conceptual framework on risk identification and mitigation process using Scrum in Global Software Development environment was created by using systematic review of literature (Hossain et al., 2009). This framework identifies the key risks in GSD while using Scrum and also suggests mitigation strategies for each risk. The work is not comprehensive, does not include inputs from industry and is based purely on the existing literature review.

Other works on risk management in distributed development based on systematic literature review have been by Jiménez, Mario & Aurora (2009) and Silva, Fabio, Costa, Cesar & Prikladnicki (2010). Silva et al. (2010) consolidated the findings from the studies on risks in management of distributed development projects published in quality journals and periodicals. Jiménez et al. (2009) found the risks related to communication, collaboration, project and process management and appraised the implications of distributed development on software quality and its measurement. These studies contributed through systematic literature review and overlooked the experiences of industry practitioners. Moreover, their focus on risks related to communication and collaboration amongst stakeholders, project management and infrastructure resulted in missing out many other areas of concerns like risks in software development life cycle, risks related to customers, other vendors and third party involvement in solution development.

Eventually, there is a clear need to explore and identify the risks in distributed agile development and design solutions to manage those risks for successful execution of such projects. A lack of comprehensive work that addresses the risks impacting DAD projects and their corresponding risk response strategies led to the present research work.
1.2 Research Objectives

This research study has the following objectives.

1. To explore the risks which occur in DAD projects and identify the significant ones.
2. To identifying the most frequently used risk management approaches in practice that can be used for controlling the impact of those risks.
3. As software projects are executed under various constraints, the research aims to study the impact of those risks on a particular project constraint of time, cost and quality of the DAD project.

These objectives are detailed in the chapter 3 in section 3.1. In the next section, we explain the research process adopted by us for identifying the risks which impact the distributed agile projects.

1.3 The Research Process

Since, there was scarcity of existing work in the area of risk management for DAD projects, we started by laying down the foundation by performing an extensive exploratory work. The exploration started with a thorough review of literature in order to understand the risks in DAD projects, if have already been recognized and the practices which are being suggested to control them. The literature review was done with the following objectives:

1. The reasons for software organizations to adopt DAD for software development and the benefits they are getting by using this approach.
2. The risks and challenges, that the organizations are facing while executing DAD projects and the suitable practices suggested for controlling those risks.
3. The extent to which the existing studies are able to fulfill the problems encountered by organizations for managing DAD projects.
4. Understand the directions which can be followed to make a significant contribution in the area of risk management in DAD project.

The detailed explanation of the review of literature has been given in the next chapter.

Due to the scarcity of literature which gives an overall view of the risks in DAD projects, we decided to perform an exhaustive exploration by collecting field data. This involved in-depth
interviews of practicing professionals (twelve) along with the analysis of project work documents (twenty eight) of DAD projects. The in-depth reading of the documents was supplemented by discussions with the project managers and the team members involved in the corresponding project. The respondents were primarily from India and few were from countries outside India. The qualitative data collected through exploration was analyzed using ‘constant comparison method’ (Glaser, 1965), which led to the creation of a risk factor categorization for DAD projects. A risk factor can be defined as a condition that can present a serious threat to the successful completion of a software development project (Schmidt, Lyytinen, Keil & Cule, 2001). The categorization comprises of eighty-one risk factors grouped to form higher level risk areas and further broad five core risk categories. The risk categories were mapped to the components of Leavitt’s model of change in order to cover all the aspects of an organization which may sustain the occurrence of risks in DAD projects.

In order to identify the most important risk factors and the most frequently used risk management methods, we conducted a descriptive quantitative study. While conducting the pilot testing of the questionnaire to be used for this study, we identified another risk category which deals with risks occurring due to the misalignments in the business vision of the customer and the project objectives of the DAD team. This risk category was named as ‘Business Objectives and Goals’.

While conducting descriptive study, we obtained ranking for risk categories (level 1), risk areas under each risk category (level 2) and risk factors under each risk area (level 3) along with the project constraints (time/cost/quality) which is being impacted most significantly. We also obtained the rating for risk management methods corresponding to each risk factor to find out which method is considered to be the most frequently used for controlling a particular risk factor in DAD projects. Non-parametric tests including Kendall’s test for Concordance and Chi-Square test of Independence were applied to analyze the quantitative data collected.

This analysis helped us to create the risk management framework for distributed agile projects. The framework presents the most important risk categories, most important risk areas in each category and the most important risk factors in each risk area which impact the performance of
DAD projects. An in-depth analysis of each risk factor was done to identify the contradicting agile principles and DSD properties, which are causing the occurrence of the risks. Risk management methods that are most frequently used in practice to reduce the impact of the identified risk factors have also been determined. The study also finds the effect of specific risk factor on the project constraints of time, cost and quality.

The risk management framework was then validated partially by implementing it in three DAD projects being executed in three different multinational IT consultancy companies. We provided the ‘Software Development Life Cycle’ risk category to one company and ‘Project Management’ risk category to two other companies for implementation. The validation results shows that the framework was useful and helped the teams to anticipate the risk factors which may impact the projects and take preventive steps to reduce their impact. We now present in brief the research findings in following section.

1.4 Research Findings
This research study involved in-depth exploration followed by descriptive study of risks impacting DAD projects. It has led to the formation of a risk management framework which can be used by the practitioners to control the risks in DAD projects. The framework is composed of the following components:


- The risk categorization includes the cause of the risk factor, the originating software development methodology that leads to the risk factor and the corresponding risk management methods that can be used to control the risk. The study of the software development method that is primarily causing the risk to occur led to the bifurcation of the risk factors in two sets, namely, DAD Risk Factors and Non DAD Risk Factor. The factors that are caused due to combination of distributed development and agile methods were termed as DAD Risk Factors (forty-four), while the risk factors that are caused due
to other development approaches including distributed development, agile and traditional methods were termed as Non DAD methods (thirty-seven).

- Further, the risk categories were mapped to the components of an open system model of organizational change, namely, Leavitt’s model of Organizational Change. Leavitt’s model views system as a multivariate system composed of four basic components: task, structure, actor and technology. The key supposition of this model is that, these four components are interrelated and a change in one component will affect the other component. This model can easily be transmuted into the basic aspects for software development. Hence, this mapping of our categorization to the Leavitt’s model gave an organizational perspective to the risk management in DAD projects (Lyytinen, Mathiassen & Ropponen, 1998).

- Besides, the five risk categories, as described above, the framework also presents an important risk category namely, ‘Business Objectives and Goals’. This risk category contains the risk factors that occur when DAD teams are unable to or are not equipped with clarity of vision of the project and that of the customers’ business. The teams are not aligned with the objectives of the customer and hence build projects that are unable to give high business value to the customer. The framework also suggests some practices that can be utilized to reduce the impact of such risks.

- It was observed that the conflict between the DSD properties, namely, Spatial Distance, Temporal Distance, Language Barrier, Work/development Culture and Project Large Size and the basic principles of agile methods are causing various risks in DAD projects. Hence, for each risk factor, DSD property that is contradicting with the agile principles and practices were identified.

- The support from literature for the identified risk factors and the risk management methods was obtained. The risk factors that are having comparatively lesser visibility in the literature were separated from those which have been recognized by more number of studies. The identification of factors with weak support opens the scope for further research and investigation.

- Further, the quantitative descriptive study helped us identify the risk categories, risk areas within the risk categories and risk factors within the risk areas that are more important for DAD projects. It was found that ‘Group Awareness’ is perceived as the most important
risk category for DAD projects which was followed by ‘External Stakeholder Collaboration’. Next, in the order of importance of risk categories were ‘Software Development Life Cycle’, ‘Project Management’ and the least important risk category is ‘Technology Setup’.

- Further, the analysis of the DAD risk factors shows, that the ‘Spatial Distance’ between the project stakeholders is the cause of majority of the risk factors. The reason is that agile approaches foster enhanced communication and collaboration between the teams and customers which is very difficult to achieve in case of distributed development due to non-collocated members. Other important DSD properties that form the cause behind many DAD risk factors are ‘Temporal Distance’ and ‘Difference in ‘Work/Development Culture’ amongst the team members. Some of the DAD risk factors are caused due to ‘Language Differences’ of the team members and few of them due to ‘Large Scope of the Projects.’

- The framework also suggests a set of most frequently used risk management approaches which can be used for controlling the risks. Along with this, the risk factors which are being impacted by ‘Time’ as a constraint, ‘Cost’ as a constraint and ‘Quality’ as a constraint were also been separated. This will help the practitioners focus on the risk factors which are impacting a particular project constraint, which is most critical for the success of the project.

- Overall this framework can be used to foresee the risk factors which may impact DAD projects and the appropriate measures that can be taken by the practitioners to overcome their impact. The findings from the validation study shows that impact of approximately 6% to 50% of risk factors can be reduced on the implementation of the partial framework. Further, it can be used by academicians and researchers as it opens new opportunities to carry research in the area of risk management in DAD projects.

1.5 Research Contribution

We created a Risk Management Framework for distributed agile projects which is based on the extensive exploratory followed by descriptive study and supported by the existing research studies. The framework created is comprehensive in nature as it encompasses all the important risks which occur in DAD projects. Although, there are studies which present the issues and
challenges in DAD projects, but they provide a situation based view only. This research study is able to give a holistic view of the risks in distributed agile by covering various aspects of software development including, the process of creation of software, management of solution development, collaboration of development team with other stakeholders and communication infrastructure and technology.

The extensive exploration helped us to create a detailed explanation of the risk factors, the cause of the risk factor, the understanding about the software development methodology which is leading to the risk factors and the methods that can be used to control the risk factor. This description can be used as a reference for handling various challenges in DAD projects by the team members and project managers.

Since, the focus of this thesis is on distributed agile development, the risks which are occurring due to combination of distributed development and agile (DAD risk factors) are analyzed in detail. For DAD risk factors, we looked at the DSD properties and the agile principles which are contradicting and hence becoming the reason for the occurrence of the risks The Non DAD risks, although are not included for ‘Discussion’ section of the thesis (Chapter 5), but are significantly important and are explained in detail in our findings (Chapter 4).

The risk categorization created was mapped to the Leavitt’s model of organization change, which is used extensively in the literature of Information Systems. The major elements of our categorization could be related to all the components of the Leavitt’s model. This helped us to relate our findings with all the aspects of risk management in an organization in order to allow easy implementation of the framework.

The exploration provides a comprehensive view of the risks occurring in DAD projects. The quantitative study is used to segregate those risk factors which are comparatively more important for DAD projects. The more frequently used risk management methods for the identified risks are also presented, which will help the practitioners to do effective risk management. The other strength of the work is that, the practitioners can consider the risk factors selectively, based on the project constraint which is most critical for the project.
The framework was validated and empirically tested by DAD teams in three different companies. We obtained results showing that the research findings were useful as it helped the teams to identify important risks which could have impacted their projects. The suggestions provided were also useful and helped them to reduce the impact of the risks. Hence, we state that the framework will be valuable for the practitioners who develop solutions by DAD approach. Even the researchers can get an exhaustive list of risk factors occurring in DAD projects. They can look at those factors which are more prominent in the current studies and even those which need more exploration. The new emerging risks in the industry which have not yet been explored with the required amount of rigor, forms the basis for further research and exploration.

1.6 Organization of Report
This dissertation is organized as follows: Chapter 2 reviews the extant literature in the areas of traditional software development, agile software development methods, distributed software development and distributed agile development. We also discuss about the benefits obtained by using distributed agile and the challenges the organizations face when using this approach. Further, the concept related to risk management and the theoretical framework which forms the basis of this research study has been discussed. Chapter 3 describes the research method utilized for this study which was divided in two phases i.e. Qualitative Exploratory Study and Quantitative Descriptive Study. The detailed description of both these phases is given in this chapter. Chapter 4 presents the data analysis and results of qualitative exploratory study followed by that of quantitative descriptive study. Along with that we also present the data analysis and results of the validation of the risk management framework. Chapter 5 presents the discussion of the exploratory results, the quantitative study findings and the findings of the validation. We present the limitations and the challenges faced while carrying the work in Chapter 6 along with the recommendations for the future scope in this area of research and finally the work is concluded in Chapter 7.