Interscience Management Review

Volume 2 | Issue 1 Article 7

January 2009

A Framework for Risk Management in Globally Distributed Agile Software Development (Agile GSD)

Suprika Vasudeva Shrivastava
Symbiosis Centre for Information Technology (SCIT), suprika@scit.edu

Hema Date
National Institute of Industrial Engineering (NITIE), hema@faculty.nitie.edu

Recommended Citation

Shrivastava, Suprika Vasudeva and Date, Hema (2009) "A Framework for Risk Management in Globally Distributed Agile Software Development (Agile GSD)," *Interscience Management Review*: Vol. 2 : Iss. 1 , Article 7.
DOI: 10.47893/IMR.2009.1022
Available at: https://www.interscience.in/imr/vol2/iss1/7

A Framework for Risk Management in Globally Distributed Agile Software Development (Agile GSD)

Suprika Vasudeva Shrivastava
Symbiosis Centre for Information Technology (SCIT)
Pune, India
suprika@scit.edu

Hema Date
National Institute of Industrial Engineering (NITIE)
Mumbai, India
hema@faculty.nitie.edu

Abstract: **Global Software development (GSD) is gaining popularity as it helps in saving cost and reduces time to market. GSD faces various challenges like communication problems, time-zone differences and cultural differences. Agile principles are used as a means to increase production rate by making processes more responsive to change. Since success of using agile methodologies is dependent on communication and collaboration, combining agile with distributed development (Agile GSD) becomes a risky process. In the current literature, few research papers mention about the risk involved in globally distributed agile development process. This paper contributes by identifying some of the most relevant risk and suggesting risk mitigation strategies in the form of a framework for risk management for globally distributed agile development. The framework is created based on the literature available and the interviews of project managers and developers conducted in three software development organizations in Pune (India). Each component of the framework is discussed in the paper in detail. The research will be relevant for the project managers who are handling distribute agile projects.**

Keyword: Global Software Development. Agile methodologies, Scrum, eXtreme Programming, Risk Management

I. INTRODUCTION

Global Software Development (GSD) is a recent trend in the software development industry. GSD practitioners claim that it is possible for organizations to gain time-zone effectiveness, leverage a large skill pool, develop software closer to the customer, and exploit low labor cost in certain parts of the world [4]. There is a growing interest in applying agile practices in GSD projects. Agile practices may also exacerbate risk as agile practices are based on the philosophy of close, frequent and collocated collaboration [5]. In addition some GSD project contextual factors, for example collaboration modes, increased number of sites, a large number of project personnel, lack of tool support etc. may also impact on project communication and collaboration processes and restricts the use of agile practices in GSD [6]. Despite a number of risks when using agile GSD, a few instances of success are found in literature when some agile practices were used by global teams [20]. Hence risk management becomes a sensible activity in globally distributed agile development

Controlling risk improves essential software development features like product quality, planning precision and cost-efficiency. This is the reason why inclusion of risk management in software development is an important factor for project success [1]. Unfortunately many software development models, both traditional and agile ones, are not well aligned with the risk management processes [2, 3].

The current literature quotes some of the risk involved in executing agile GSD projects but, still does not gives a complete view of the same [6]. Moreover the risk list available is not generic to agile. Most of the papers concentrate on scrum as the base for their research.

In this paper we suggest a framework for risk management for distributed agile projects based on the literature survey done and the interviews conducted in the industry. The framework classifies risk involved in distributed agile development in three categories: Globally distributed development factors, agile factors and project management factors. It also suggests some risk mitigation steps for the same which can be used by the project managers to carry distributed agile projects.

II. CONCEPTUAL FOUNDATION

A. *Agile Software Development*

Agile software development refers to a group of software development methodologies aiming to more nimble and lighter development processes, making them more responsive to change. Among various agile methods, eXtreme Programming focuses on development practices and Scrum focuses on project management practices, are the most well known. Agility in short means to strip away as much of the heaviness, commonly associated with traditional software development methodologies, as possible, in order to promote quick response to changing environments, changes in user requirements and accelerate project deadlines [7]. Agile methods have proved to be ten times more productive than traditional development models and this has been achieved in the co-located teams [8].

Basically agile methodologies prefer software development over documentation. It believes in delivering many versions of the software in short iterations, and then updates the software according to the customer's feedback. This methodology helps in overcoming the problems like frequent changes and leads to faster development and greater user satisfaction.

B. *Global Software development*

Many organizations have started developing software remotely in order to achieve lower cost and access to skilled resources. Moreover large investments have enabled a move from local to global markets in the process of creating new competition and collaboration forms [9].Various factors are responsible for such a situation: pressure to improve time-to-market, create a pool of globally available skilled resources on reduced cost [10], minimized risk in case of natural catastrophes and other events [11]. Thus software development is now becoming muti-site, multicultural, globally distributed undertaking. Along with the benefits obtained through globally distributed development, there are many difficulties faced by various organizations. These problems are caused mainly due to distance, time zone difference, and cultural differences [4, 13].

C. *Need to extend Agile with GSD practices.*

Both agile and distributed software developments (and GSD) are growing trends as software business requires quicker quality production at a cheaper price. Distributed development is a fact of life for many agile teams. 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 globe. In the 2008 State of Agile Development survey, conducted by VersionOne, 57% of respondents stated that their teams were distributed. Further 41% of respondents state that they were currently using or plan to combine agile with outsourced development. 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 [14]. Thus there is a need to extend the agile practices to globally distributed software development.

D. *Benefits achieved by combining agile with Global Software Development*

Agile methods can be beneficial when combined with distributed development. There are studies which show that agile principles help in overcoming some challenges faced by distributed development [15, 19, 17, 18, 19]. Distribution of development (DSD/GSD) seems to cause decreased visibility of project status and agile process based on short continuous iterations make it easier to see the problems already on early stages of the project. Continuous integration of software code, which is a central part of agile methods, also helps to reduce configuration management issues. Use of agile principles seems to have a positive effect on communication between teams as development in cycles makes it easier for participants to see the short term goals [22]. Sprint reviews help the stakeholders to share information about the feature and requirement dependencies. Agile principles can even help create trust between different cultures involved in the process by constant communication and delivery of software [23]. Due to increased communication and collaboration there is an improvement in the software quality and increase in the team motivation [24]. Thus, agile in distributed teams has proved to be beneficial for project's quality and performance.

E. *Risk in Agile GSD*

Agile when combined with GSD also brings some new challenges and risks. Agile is more effective for co-located teams which is small in size. Distributed projects face the challenge of spatial, temporal and goal distribution due to large distance, time zone differences and difference in each development centre's goals [31]. Spatial distribution weakens the social relations, limits face-to-face interactions and makes it difficult for the project manager to track individual contribution to the project progress. Temporal distribution makes co-ordination difficult, causes unproductive delays due to time setting problems. Goal distribution occurs due to faulty information transfer or more concentration o own site's work. It leads to conflicts related to task interpretation, process principles and problem resolution approaches and ultimately low performance and site wars [31].

Other GSD factors like lack of communication, cultural differences, large no. of development centers, lack of tool support and infrastructure also impact the development heavily. Various Agile related factors like lack of documentation, customer not agile aligned, unclear user story status also adds to the risk involved in the distributed agile projects. Thus risk management becomes a critical project management activity which can proactively help in preventing problems in a continuous and concurrent manner [12].

I. RESEARCH METHOD

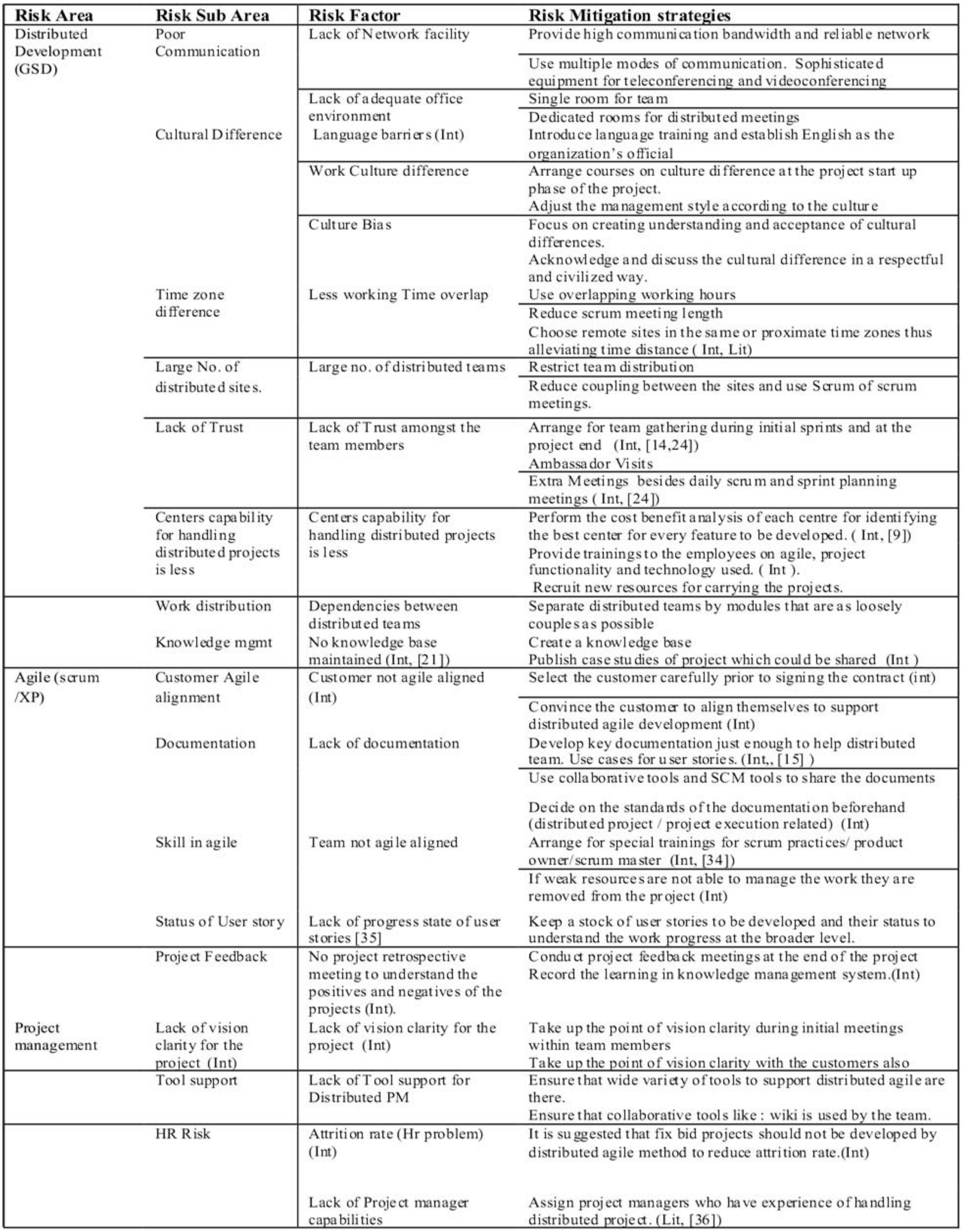
Various risk related to distributed agile development were studied from the research literature available. Overall there is a scarcity of literature on risk related areas of distributed agile development. Although some researchers do lists some risk involved when distributed agile projects are developed and some steps to mitigate those risks, but do not cover all the risk factors for distributed agile. Hence, this paper contributes by suggesting some more risk areas in distributed agile which are likely to be faced by project managers and also suggest various strategies which can be adopted to mitigate those risks.

Different risk factors were identified from the literature survey and used to generate an open ended questionnaire to understand the risk faced by the industry. We conducted 4 interviews in three organizations namely ABC (2 interviews), DEF (1 interview) and XYZ (1 interview) to understand the challenges faced by the practitioners while executing distributed agile projects. ABC typically uses XP and DEF, XYZ are into scrum for developing agile projects. All the three organizations are global IT consultancies located in Pune, India. They are having footprints all around the world with offices in the range of 12-25. Typically project managers and developers were interviewed because they could give us both project management and project execution views of distributed agile projects. Each interview was 1 to 1.5 hrs long. Two interviews out of the four were conducted by students (undergoing their post graduation in SCIT, Pune). The students were provided with relevant literature and were given inputs on concepts of distributed agile and risk management before they could go for collecting data. These students also have experience of working in the IT industry in India for 2-3 years as developers or consultants.

The risk parameters and risk mitigation steps were further updated based on the inputs we obtained from the interviews. The data collected from the literature and the interviews were consolidated in a framework. This framework will help project managers to keep a track of the risk which may impact their project negatively and can thus take measures to mitigate the risks.

Framework for Risk mitigation of Distributed Agile Development

The framework is composed of broad risk area, risk subareas, various risks under each subarea, various strategies which can be use to mitigate risk. The classification of risk areas and mitigation strategies have been formulated by using relevant literature and data collected from the interviews. The risk factors or the mitigation steps which are supported by interviews only are marked as (Int). The factors and mitigation steps which are supported by literature and interviews are marked as (Int, Lit). This shows that these factors are having heavy weightage and should be considered more seriously by the project managers. All the risk factors which are not marked are supported by literature available. Explanation of each risk factor and their mitigation strategies are explained later.



Risk Area	Risk Sub Area	Risk Factor	Risk Mitigation strategies
Distributed Development (GSD)	Poor Communication	Lack of Network facility	Provide high communication bandwidth and reliable network
			Use multiple modes of communication. Sophisticated equipment for teleconferencing and videoconferencing
		Lack of adequate office environment	Single room for team
			Dedicated rooms for distributed meetings
	Cultural Difference	Language barriers (Int)	Introduce language training and establish English as the organization's official
		Work Culture difference	Arrange courses on culture difference at the project start up phase of the project. Adjust the management style according to the culture
		Culture Bias	Focus on creating understanding and acceptance of cultural differences. Acknowledge and discuss the cultural difference in a respectful and civilized way.
	Time zone difference	Less working Time overlap	Use overlapping working hours
			Reduce scrum meeting length Choose remote sites in the same or proximate time zones thus alleviating time distance (Int, Lit)
	Large No. of distributed sites.	Large no. of distributed teams	Restrict team distribution
			Reduce coupling between the sites and use Scrum of scrum meetings.
	Lack of Trust	Lack of Trust amongst the team members	Arrange for team gathering during initial sprints and at the project end (Int, [14,24]) Ambassador Visits
			Extra Meetings besides daily scrum and sprint planning meetings (Int, [24])
	Centers capability for handling distributed projects is less	Centers capability for handling distributed projects is less	Perform the cost benefit analysis of each centre for identifying the best center for every feature to be developed. (Int, [9]) Provide trainings to the employees on agile, project functionality and technology used. (Int). Recruit new resources for carrying the projects.
	Work distribution	Dependencies between distributed teams	Separate distributed teams by modules that are as loosely couples as possible
	Knowledge mgmt	No knowledge base maintained (Int, [21])	Create a knowledge base Publish case studies of project which could be shared (Int)
Agile (scrum /XP)	Customer Agile alignment	Customer not agile aligned (Int)	Select the customer carefully prior to signing the contract (int)
			Convince the customer to align themselves to support distributed agile development (Int)
	Documentation	Lack of documentation	Develop key documentation just enough to help distributed team. Use cases for user stories. (Int,, [15])
			Use collaborative tools and SCM tools to share the documents Decide on the standards of the documentation beforehand (distributed project / project execution related) (Int)
	Skill in agile	Team not agile aligned	Arrange for special trainings for scrum practices/ product owner/scrum master (Int, [34])
			If weak resources are not able to manage the work they are removed from the project (Int)
	Status of User story	Lack of progress state of user stories [35]	Keep a stock of user stories to be developed and their status to understand the work progress at the broader level.
	Project Feedback	No project retrospective meeting to understand the positives and negatives of the projects (Int).	Conduct project feedback meetings at the end of the project Record the learning in knowledge management system.(Int)
Project management	Lack of vision clarity for the project (Int)	Lack of vision clarity for the project (Int)	Take up the point of vision clarity during initial meetings within team members Take up the point of vision clarity with the customers also
	Tool support	Lack of Tool support for Distributed PM	Ensure that wide variety of tools to support distributed agile are there. Ensure that collaborative tools like : wiki is used by the team.
	HR Risk	Attrition rate (Hr problem) (Int)	It is suggested that fix bid projects should not be developed by distributed agile method to reduce attrition rate.(Int)
		Lack of Project manager capabilities	Assign project managers who have experience of handling distributed project. (Lit, [36])

TABLE 1: Framework for Risk Management in Globally Distributed Agile Development

A. Distributed Development Factors:

Software project teams have become geographically distributed, which helps in cost reduction, improved quality and increased productivity. This leads to new challenges and risk related to communication problems, cultural issues and knowledge management.

1 Poor Communication: software development requires great deal of formal communication through vital communication channels. The complex infrastructure for GSD and great size of personnel network which changes over time, lead to decrease in communication frequency and quality, which directly affects productivity [21]. Reasons behind this risk may be lack of network facility or lack of collaborative office environment.

Lack of network facility: In order to support rich communication project managers need to provide high communication bandwidth and reliable network support throughout the development life cycle. Another practice which helps to reduce the impact of this risk is usage of multiple modes of communication [37]. Various communication tools like web camera, teleconference, video conference, net meeting, SMS, Internet relay chat can be used. Videoconferencing (preferable) or teleconferencing can be used for distributed session meetings [24].

Lack of adequate office environment: office infrastructure should be able to support distributed agile development. A common practice which ensures good communication is that the co-located team works in a single room. Dedicated rooms for scrum meetings with the required infrastructure also help to improve communication [24]. In some cases virtual conference room can also be used as a dedicated meeting room for scrum meetings [25].

2 Cultural Difference: when projects are distributed, a number of cultural problems like language barriers, work culture difference, cultural bias arise. These factors may have devastating impact on the project's execution. The project manger should try to improve the social integration by adopting certain strategies which helps in reducing the impact of culture difference.

Language barriers: lead to unconveyed information and misinterpretations [26]. This risk can be handled by introducing language training and establishing English as the official language of the organization [27]. All the organizations (ABC, DEF and XYZ) agreed to this risk as the one to be considered seriously.

Work culture difference: may arise due to difference in team behavior, perception of authority and hierarchy, planning, punctuality and organizational culture [26, 10]. Project manager can arrange for courses on cultural diversity during the project startup [26]. Adjusting the management style according to the culture e.g. participant's preferences for well-defined task or loosely defined task and self management.

Culture bias: occurs when project participants consider their norms and values as universal and neglect the others cultural norms [26, 28]. Focusing on creating understanding and acceptance of the cultural differences amongst the team members e.g.: letting each participant make a presentation on their individual culture, values, and expectations may help. Along with this discussing the cultural differences in a respectful and civilized way will reduce the gaps [28].

3 Time zone difference: distributed scrum meeting practices are found difficult when a GSD project involves a lack of overlapping working hours. Some strategies which can be followed are:

Overlapping work hours: working hours of the distributed teams can be adjusted so that the overlapping hours may increase. Other option like allowing scrum team members to attend meetings from home also supports synchronous communication [23, 19].

Reduce the scrum meeting length. Scrum teams can perform strict time-boxed meetings (thirty minutes Scrum planning meeting) in order to reduce the meeting length. Pre-preparation before attending the actual meeting also helps e.g.: posting three daily scrum questions, working on product backlog [18].

Choose remote sites in the same or proximate time zones. This will help in getting more number of overlapping hours for distributed teams; hence promote easy communication [29]. Organization ABC also stated that they prefer development centers whose time difference is aligned with the timing of their clients (E.g. for US client, a near shore centre like Brazil is preferred.

4 Large Number of distributed sites: as the number of sites increases, the management of the project becomes more difficult. The reason is project stakeholders' distribution over multiple sites, with different time zones may restrict the communication and collaboration amongst the team members [20].

This challenge can be handled by following the below mentioned steps:

Restrict the team distribution. A fully integrated team can be restricted to a limited number of sites. One method can be distributing project over multiple sites, but each Scrum team is distributed between only two sites [30]

Reducing coupling between the sites also helps. Reduce the dependency between the sites by allocating task with well defined interfaces and independent architectural interfaces [29]. In this kind of a scenario, one of the best practices recommended by Scrum Alliance is Scrum-of-scrum model. This model partitions work across isolated scrum teams with most of the dependencies eliminated. Scrum teams are linked by scrum-of-scrums where scrum masters meet regularly across locations [19].

5 Lack of Trust: due to geographical distance, there is always lack of trust and group awareness amongst the team members. This can be reduced by bringing the team together, particularly on pivotal points. Various steps which can be taken by the project manager:

Arrange for team gathering during initial sprints and at the project end. There is no substitute to face-to-face meetings to develop trust and shared identity faster [14, 24]. Thus team members should meet at the beginning of the project even before the distribution of work takes place. The members will get to know each other better and strengthen the group synergy. The team should also meet at the project end phase. This will help to summarize the experiences and find out what went well and what went wrong during the whole process of distributed development [31].

Sending ambassadors to distributed sites helps in facilitating communication amongst the team. Product owner, project manager and developers can be exchanged amongst the distributed sites. This helps in communicating both business and technical aspect of the project to the team [15]. This helps in increasing the group cohesion and trust building.

Extra Meetings besides daily scrum and sprint planning meetings also increases trust and collaboration. In order to clarify the doubts and get an update of the project progress Scrum team members may have some extra meetings also besides the daily scrum, distributed scrum of scrum and sprint planning meetings [24]. A senior project manager at ABC organization also suggested that having distributed meetings of project managers also helps to give a fair idea of the project progress to the project managers in all the distributed teams.

6 Centers capability for handling distributed Agile: the center(s) where the development will be carried out needs to be analyzed before allocation of the project features. A major risk is involved, if the centre is lacking in resources, experience in handling distributed projects or some other factor like time zone difference. Center capability factors like: center's experience in handling distributed projects, center's knowledge in the area, time to train the employees or obtain new employees, physical space and infrastructure, language barrier, time zone difference, attrition rate can be used to a cost benefit analysis of the centre. This will help in analyzing which center is better to develop which feature of the project [9]. If in case the project is allocated to a centre with less capability, new resources (human and infrastructure) needs to be arranged to fulfill the gap. Trainings on Agile (scrum master training, product owner training and agile methodologies) or trainings on technology to be used may be arranged for the employees. An interview at DEF also strongly supported this point.

7 Lack of use knowledge management: this challenge is faced by various organizations. They do not have a well developed and updated knowledge base or it is there but in a very nascent phase. The experience of the project manager, developers, scrum masters, product owners while carrying out the project, the methods used, the risk faced and the strategies used to mitigate them should be recorded. This helps the new team members to use the experience of his predecessors for working on his project. Distributed development must support knowledge sharing by maintaining a product/process repository focused on well-understood functionality by linking content from sources such as e-mail and online discussion, and sharing metadata information among several tools [21]. At ABC, knowledge management still needs to be developed, although they are maintaining some case studies of some very successful projects as published articles on their website.

8 Distribution of work: One of the challenges when the team distributes work is not to do so according to location. The architecture will begin to reflect the team's geographical distribution (according to Conway's Law). Different locations will become over specialized in particular components.

Much of the traditional thinking on the onshore/ offshore boundaries is based on the activity that people do. So analysis and design is done onshore, construction done offshore, and acceptance testing is done onshore. This obviously fits well with the waterfall model. On the contrary to this, matters improve when the offshore team handles as many activities as possible. When we split the effort with onshore and offshore development teams, we do this along functionality grounds rather than activities. We break the system into broad modules and let the offshore team tackle some of these modules. A corollary to this is to not divide teams horizontally (having one team do presentation and another do database). This means that it's important to separate distributed teams by modules that are as loosely coupled as possible [15].

B. *Agile Factors:*

1 Customer not agile aligned: One of the major risks faced by agile company is that, the customer is not agile aligned. The customer is either not well-equipped or not willing to co-operate the development team to carry the project development in the agile way. Organization ABC said that they prefer to work with those customers who are ready to work according to agile practices. Organization DEF said that they try to convince the customer to co-operate and support agile practices for project development. This is an important factor because agile practices require a heavy participation of the customer during the project development.

2 Lack of Documentation: Agile methods downplay documentation from the observation that a large part of documentation effort is wasted. Documentation, however, becomes more important with offshore development since the face to face communication is reduced. Poor documentation can cause ineffective collaborative development [15]. In GSD, however, in addition to documenting the various artifacts, updating and revising the documentation is equally important. To prevent assumptions and ambiguity and to support maintainability, documentation must be current and reflect what various teams are using and working on [10].

Thus, it is advised to create just enough documentation to support teams which are distributed and encourage knowledge sharing for e.g.: developing use cases for stories [15].

In order to support documentation, active collaboration tools: wikis, issue tracking tools are also used. Moreover, it is better to favor tools that impose less structure, that way the team can fit them better into how they want to work (one of the reasons that wikis can work so well [15]. Various tools like issue tracker (e.g. Jira), project management tool (e.g. Scrum works) also helps in maintaining documentation and good transparency [6]. Teams are always advised to check their documents into a version control system so people can easily get the most up to date material [15]. This is particularly important when you are doing remote work. Maintaining valuable documentation may also improve DSD team collaboration process while using agile practices [16, 6, 32, 33]. For example providing user stories with use case diagrams in globally accessible backlogs helps to reduce misunderstandings and improve team collaboration.

Organization DEF said that standards of documentation are also pre decided so that all teams can follow the same for distributed agile development.

3. Lack of skill in Agile methodology: one of the significant reasons for agile project failure is the lack of teams members who are skilled in agile methodologies [34]. According to Gartner Research, "Experienced staff should represent 20% to 30% of the team, with no more than 20% of their time given to supervising and mentoring junior members." And further from Gartner, "Organizations new to agility should invest in staff training on the selected agile method, and where the "agility skills gap" is large, they should bring in the necessary expertise. Organization DEF & XYZ stated that they provide trainings on agile for scrum masters, product owners or general agile methodologies. DEF also added that if the employees are still not able to show results, they are removed from the agile project.

4. Status of User story Development: usually burndown charts are created to understand the work in progress, but that provides the work done in terms of efforts expended. It will not provide the indication of the defects left per story or the quality of the story. This lack of overall picture of the work progress may lead to schedule overruns. Thus taking a stock of the user stories in an excel sheet format helps in giving a clear view of the stories that could not be completed due to technical issues or defects [35].

5. Project Feedback system: there is a meeting at the end of each sprint for reviewing the work done during the sprint and evaluate the progress of the project.

These meetings can be local within a scrum team and also done with the offshore teams. There is as such no meeting which gives feedback on the how the project was carried, what were the risk and challenges and how were they mitigated at the end of the whole project. Organization ABC stated that this kind of meeting would help them to perform evaluation and feedback which will be a tool for continuous improvement of the whole process of development. This meeting would concentrate on processes carried and decisions taken end-to-end during the project development starting from strategic decisions, allocation of project to centers, to how the project was carried to deployment. The risk identified and leanings obtained from such meeting would be recorded in a knowledge management system.

B. Project management Factors:

1. Lack of vision clarity: Sometimes a project starts, but not every detail is clear. That may happen because the beauty of Scrum is its adaptability to uncertainties. However, as time progresses and core structures are being built, the lack of a clear vision becomes more and more of a problem. The reason behind this is that the product owner, whose responsibility is to define the product goal and vision in cooperation with the client and communicate it to the team is himself not clear about the vision of the project. This can be a tedious process, especially if the client is unclear about his goals or has difficulty articulating them [38]. This risk needs to be covered by the product owner by taking this point seriously in meetings with the client as well as with the team members during the initial phase of the project itself. He should be is able to assist the client and take the lead when necessary to come up with a product vision that is comprehensive, thought-through and makes clear to the team where the project is heading. Organization ABC mentioned very strongly about this risk and recommended that initiatives from the product owner or the project manager should be taken in this direction else the project is very prone to failures.

2. Tool Support: Tool support is extremely essential for distributed agile teams for maintaining a natural flow of work. Various categories of tools for supporting team collaboration and communication within the teams, for supporting agile processes and distributed project management are used. Team members can use free head sets, web cams, instant messenger for synchronous communication and e-mail for asynchronous communication. Tools like wikis, blogs, whiteboards, electronic work space etc can be used for sharing common information amongst the distributed teams [15, 24]. These tools are mainly to connect geographically distributed teams and increase communication [14]. Another category of tools are the ones which support agile processes when teams are not co-located. Agile teams cannot rely on sticky notes, task board, burndown charts on the wall for project tracking. Various collaboration and project management tools like bug tracker, issue tracker, globally accessible backlog tool and burndown charts, backlog management tool while using scrum are used. Other tools for supporting distributed SCM processes are also used for maintain version controlling, maintaining repositories and status of the project. This helps in a sharing and controlled access of software configuration items amongst the distributed teams.

3. Human Resources related risk

Attrition Rate: Distributed agile projects face problem when the project is fixed bid with fixed time, budget and quality. Organization DEF stated that if a project is fixed bid then the project cannot have many people getting promoted since it affects the costing. This in turn affects the appraisals. Since many people cannot be promoted, this inculcates a sense of insecurity and increases attrition.

Lack of project manager’s capabilities: The traditional project manager (PM) who manages the triple constraints (scope, time and resources) through the use of a project plan will need to change his/her approach to managing the agile team. The successful agile PM must migrate from management to leadership, from monitoring compliance to enabling self-direction, and from acting as a foreman to becoming a facilitator of creativity and innovation. They have to keep their focus on delivering value to the customers which will be the ultimate measure of success of the project. They have to trust their team members and work along with them and keep them motivated instead of being only a task-manager [36].

In an interview with organization DEF, project managers in distributed agile projects need to keep their resources engaged and reduce attrition. A project manager should be handling only 9-10 members of agile teams in a location. One project manager should be working only one project or maximum two projects in the same domain.

4. Inappropriate estimates (cost / time/schedule): Estimates are an essential part of Agile Project Planning. Software estimation has always been problematic and people have proposed many different ways to do estimating. Different methods are on a spectrum from formal to informal and from supposedly objective to seemingly subjective. Some methods estimate size and derive effort while others estimate effort directly. A project manager at DEF stated that estimates are made for each sprint. Once the sprint starts they come to know if the estimates were

correct or not. Accordingly re-estimation of user stories should be done if required. If the project manager feels that the estimates are not correct he should make re-estimates on the user stories. Estimates are given using function-points, but estimates through experience tend to show more accuracy. Functional and technical expertise is very essential for right estimates.

I. CONCLUSION AND FUTURE SCOPE

Globally distributed agile development (Agile GSD) is a trend which is gaining popularity day by day. It is possible to tap into new global markets and make best use of globally available talent, while potentially reducing costs by using global distributed development (GSD). GSD faces key issue like lack of communication, cultural differences, lack of trust and collaboration amongst the team, hence reduced software quality. Along with this, usage of agile with GSD is gaining strength.

There are many benefits of using agile methods with distributed software development. It helps in evaluating and measuring the progress of the project and problems of the project are more easily noticed at the early stage. It also handles the problems related to communication challenges in GSD such as difficulties in initiations and maintenance of communication. There are new challenges and risks introduced when agile is combined with distributed software development, which need to be taken care off. Some researchers have worked in this direction and suggested some risk factors and mitigation strategies. We have contributed by adding more risk and mitigation steps to the current body of knowledge through the literature survey and interviews conducted IT organizations located in Pune (India).

A framework is created which divides the risk for distributed agile in three major areas: risk related to global distributed development, risk related to agile development methodology and risk emerging due to project management issues. These areas are further divided in certain sub-areas and risk factors under each sub area are listed. Corresponding to each risk factor some risk mitigation steps are suggested.

The framework suggested here is the output of a preliminary research work done in this direction. In the near future we intend to conduct more such interviews and explore more risk parameters and their mitigation steps. Thus, the framework will be further modified and upgraded. We are sure that this framework will be useful for the project managers for dealing with projects which are developed using distributed agile methodology.

II. ACKNOWLEDGEMENT

Suprika V Shrivastava thanks Nipun Mohanty, Joel John Saldanha and Mahendra Kumar Dodke for their support and data collection for this research.

III. REFERENCES

[1] Englund H., "A Case Study to Explore Risk Management Models", Master's Thesis, Royal Institute of Technology, Sweden, 1997

[2] Boehm et al., "Spiral Development of Software-Intensive Systems of Systems". Proc. of International Conference on Software Engineering, USA, 2005

[3] Sliger M., "Relating PMBoK Practices to Agile Practices". URL: http://www.stickyminds.com/sitewide.asp

[4] E.Caramel, Global software teams: collaborating across borders and time zones: Prentice-Hall, 1999.

[5] P. Abrahamsson, O. Salo, J. Ronkainen, J. Warsta, Agile software development methods - Review and analysis, VTT Electronics ed.: VTT Publications, 2002.

[6] Emam Hossain, Muhammad Ali Babar, Hyw-young Paik, "Risk Identification and Mitigation for Using Scrum in Global Software Development: A Conceptual Framework", 16th Asia-Pacific Software Engineering Conference, 2009.

[7] J. Erickson, K. Lytinen and K. Siau, "Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research.", Journal of Database Management, 16(4), 2005,p. 88-100.

[8] R. Phalnikar, V.S Deshpande, S.D. Joshi, "Applying Agile Principles for Distributed Software Development", International Conference on Advanced Computer Control, p.535-539 ,IEEE, 2009

[9] R. Priklandnicki, J. Audy Nicolas, R. Evarito, "A Reference Model for Global Software Development: Findings from a Case Study.", IEEE International Conference on Global Software Engineering (ICGSE '06), 2006.

[10] J.D Herbsleb and D. Moitra (2001), "Global Software Development, IEEE Software, March/April, USA, p. 16-20, http://conway.isri.cmu.edu/~jdh/collaboratory/research_papers/IEEE_SW_editorial_final.pdf (2010)

[11] Ismo Lehtonen, "Communication Challenges in Agile Global Software Development", University of Helsinki, Department of Computer Science, Faculty of Science., 2009.

[12] R. Prikladnicki, M.H. Yamaguti "Risk Management in Global Software Development: A Position Paper." http://gsd2004.cs.uvic.ca/camera/prikladnicki.pdf., 2004. (2010)

[13] S Krishna, S. Sahay, and G. Walshman, "Merging cross cultural issues in global software outsourcing," Communications of the ACM, vol 47, no. 4, pp. 62-66, 2004.

[14] Ade Miller," Distributed Agile Development at Microsoft patterns and practices", Microsoft patterns and practices,

[15] M. Fowler, " Using an Agile Software Process with Offshore development", http://martinfowler.com/articles/agileOffshore.html, July 2006 (2010)

[16] H.Smits, "Implementing Scrum in a Distributed Software Development Organization", Agile 2007, p.371-375, IEEE.

[17] M.Kircher, P.Jain, A.Corsaro, D.Levin, "Distributed eXtreme Programming", Proceedings of the International Conference on eXtreme Programming and Agile Methods, p.147-154, 2004

[18] K.Sureshchandra, J.Shrinivasavadhani, " Adopting Agile in Distributed Development", IEEE International Conference on Global Software Engineering , p.217-221, 2008.

[19] International Conference on System Sciences, p.274a-274a, IEEE, 2007.

[20] J. Sutherland, K. Schwaber, "The Scrum Papers: Nuts, Bolts, and Origin of an Agile Process" http://scrumtraininginstitute.com/home/stream_download/scrumpapers, Last accessed 25 March 2009.

[21] Miguel Jimeenez, Mario Piattini , Aurora Vizcanio, "Review article Challenges and Improvements in Distributed Software Development: A Systematic Review", Hindawi Publishing Corporation, Advances in Software Engineering, Volume 2009, Article ID 710971.

[22] M.Pikkarainen, J.Haikara, O.Salo, P.Abrahamson, J.Still, "The impact of agile practices on communication in software development", Empirical Software Engg,Volume 13, Issue 3, 13:303-337, 2008

[23] M.Paasivaara, C.Lassenius, "Could Global Software Development Benefit from Agile Methods?" International Conference on Global Software Engineering 2006, p.109-113

[24] M. Paasivaara, S. Durasiewicz, C. Lassenius, "Distributed Agile Development: Using Scrum in a Large Project," Software Process Improvement and Practice, Vol. 13, Issue 6, pp. 527-544, 2008.

[25] W. Williams, M. Stout, "Colossal, Scattered, and Chaotic (Planning with a Large Distributed Team)," in Proceedings of the Conference on Agile 2008, pp. 356-361, 2008

[26] L. Dube and G. Pare, "Global virtual teams," Communications of ACM, vol. 44, no. 12, pp. 71–73, 2001.

[27] C. Ebert and P. D. Neve, "Surviving global software development," IEEE Softw., vol. 18, no. 2, pp. 62–69, Mar./Apr. 2001.

[28] C. M. Solomon, "Global teams—The ultimate collaboration," Pers. J., vol. 74, no. 9, pp. 49–58, 1995

[29] E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," IEEE Softw., vol. 18, no. 2, pp. 22–29, Mar./Apr. 2001

[30] M. Vax, S. Michaud, "Distributed Agile: Growing a Practice Together" in Proceedings of the Conference on Agile 2008.pp.310, 2008.

[31] John Stouby Persson, "Managing Distributed Software Projects" PhD. thesis, http://vbn.aau.dk/files/33303765/Persson-Thesis.pdf, August 2009

[32] H. Holmstrom, B. Fitzgerald, P. J. Agerfalk, E. O. Conchuir, "Agile Practices Reduce Distance in Global Software Development" Information Systems Management, Summer, pp. 7- 26, 2006

[33] S. Berczuk, "Back to Basics: The Role of agile Principles in Success with a Distributed Scrum Team," in Proceedings of the Conference on AGILE 2007, pp. 382-388, 2007.

[34] "Agile in the Enterprise", Serena, 2007, http://www.serena.com/docs/repository/products/teamtrack/agile-in-the-enterprise.pdf (2010)

[35] Raja Bavani, "Critical Success Factors in Distributed agile for Oursources Product Development", Proceedings of CONSEG 09, International Conference in Software Engineering, Dec 2009.

[36] Nancy Nee "Successful Projects through Agile Project Management", http://www.projecttimes.com/agile/successful-projects-through-agile-project-management.html (2010)

[37] B. Drummond, J. F. Unson, "Yahoo! Distributed Agile: Notes from the World Over," in Proceedings of the Conference on Agile 2008, pp. 315-321, 2008. 463

[38] Nils Weisensee, "Five mistakes of scrum product owner that can make your project fail", July 27 2010, http://www.thenetcircle.com/2010/07/27/five-mistakes-of-a-scrum-product-owner-that-can-make-your-project-fail/ (2010)