Effectiveness of Requirements Validation Techniques within GSD w.r.t. Communication and Coordination

By

Asma Munir
(126 – FAS/MSSE/F06)

Supervised By

Dr. Naveed Ikram

Department of Computer Science
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad

2009

بسم الله الرحمن الرحيم

In The Name Of Allah The Most Beneficent The Most Merciful

Department of Computer Science
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad

Date: 25th May, 2009

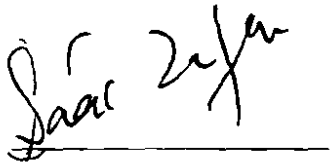
Final Approval

It is certified that we have read the thesis titled "Effectiveness of Requirements Validation Techniques within GSD w.r.t. Communication and Coordination" submitted by Miss Asma Munir, Registration No. 126 – FAS/MSSE/F06. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for MS degree in Software Engineering.

Committee

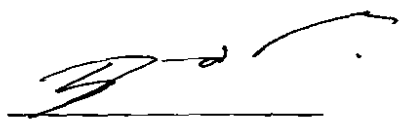
External Examiner:

Dr. Saad Naeem Zafar
Dean, Faculty of Computing
Riphah International University.



Internal Examiner:

Miss Salma Imtiaz
Lecturer, Department of Computer Science
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad.



Supervisor:

Dr. Naveed Ikram
Associate Professor, Department of Computer Science
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad.

Now I know, a refuge never grows,
From a chin in the hand and a thoughtful pose
Gotta tend the earth if you want a rose.

~Indigo Girls

To My Mother

You are my friend and my teacher.

You are my endless inspiration.

Thank you for your unwavering faith in me.

Thank you for your steadfast support in all my endeavors.

I am where I am only because of you.

I love you.

A dissertation submitted to the

Department of Computer Science,

International Islamic University, Islamabad

as a partial fulfillment of the requirements

for the award of the degree of

Master of Science in Software Engineering

DECLARATION

I hereby declare that this thesis neither as a whole nor as a part has been copied out from any source. It is further declared that I have written this dissertation entirely on the basis of my personal efforts made under the sincere guidance of my supervisor Dr. Naveed Ikram. If any part of this dissertation is proven to be copied out from any source or found to be reproduced from someone else's work, I shall stand by the consequences.

No portion of work presented in this dissertation has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Asma Munir
126 – FAS/MSSE/F06

ACKNOWLEDGMENT

In the writing of this thesis, I am exceptionally grateful to Allah for granting me the strength, patience and capability for undertaking and accomplishing a task that at many times seemed terrifyingly daunting, excruciatingly difficult and utterly unachievable. I am grateful to Allah for making me strong enough to adequately deal with all the hurdles and hardships I encountered on the way and for helping me keep sane throughout the process.

I want to express my gratitude to my supervisor Dr. Naveed Ikram for his invaluable guidance and steadfast support throughout the thesis time. His timely feedback and rational comments helped me achieve goals that otherwise would have been quite impossible. I thank Allah repeatedly for blessing me with a teacher who sets a great example of what a person should be, who encourages you to develop an internal strength, a strong trust in your own abilities and an unwavering faith in Allah to guide you at all times.

I am exceedingly thankful to my beloved parents; I could not have achieved anything without their endless faith in my abilities and their constant moral support. Thank you for being there for me always, thank you for letting me lean on you always, thank you for all those reassurances, thank you for making me believe in myself, thank you for making me the person that I am today, thank you for everything!

I want to thank my best friend Amna. You have been there with me at every step of the process, I don't think I can thank Allah enough for blessing me with such a sweet, special, kind and supportive person. You understand me without my having to utter a single word. I would probably have been reduced to pulling my hair out of exasperation and frustration if it were not for you. Thank you for being the lovely person that you are, thank you for being my friend.

I would also like to thank Sarah, my little sister. You tolerated me through all those grumpy and moody times. Thank you for being so understanding, thank you for all those fun and relaxing times, those midnight snacks accompanied with endless movies and TV shows, thank you for your sweet company. I miss you and all those cool times endlessly.

I want to express my thanks to my cousin, my little brother Hassan as well. You are actually one of the very few persons who truly understood the hard time I was going through. You were there for me with your kind and understanding words. Thank you so much for that. Also, thank you for your generous help with providing me the resources for research papers. I would not be exaggerating if I said that nothing would have been possible without your help in this regard.

I would also like to express my gratitude to all my MS class fellows for their support, understanding and encouragement.

Last but not the least; I would like to thank my husband Faraz. Thank you for being the funny, kind and sweet yet stern person that you are. I am grateful to Allah for having a person like you beside me through all the times to come.

ABSTRACT

Requirements validation activities ensure that the developers are working with a set of requirements that are a true representative of users' intents. Various techniques are employed to accomplish requirements validation. These techniques face challenges when employed in the context of global software development, especially by the horrendous issues of communication and coordination in geographically distributed software development environments. This thesis firstly proposes a framework for evaluating requirements validation techniques after an in-depth study of literature pertaining to communication and coordination problems in GSD, communication and coordination requirements for a particular technique and factors that enable and influence communication and coordination in a geographically distributed environment. Based upon this framework, a comparative analysis of these techniques is performed to identify their strengths and weaknesses in a global context and provide recommendations to increase their effectiveness in a distributed setup. A solution for best dealing with the requirements validation techniques and communication and coordination problems encountered by them in global software development is presented. Finally, the proposed solution is validated by means of a case study.

TABLE OF CONTENTS

CHAPTER 1

1 INTRODUCTION

The trend towards development of software systems in a globally distributed environment has been growing rapidly for the past few decades [1]. Management, development and maintenance of software systems have evolved from being concentrated at a single site to being distributed over multiple sites over the globe [8]. Various forces have contributed towards this situation including the need to capitalize on global resource pools, concerns for costs, proximity to market, improving time to market by using time zone differences in round-the-clock development and mergers and acquisitions [1, 2].

In global software development (GSD) context, requirements have to be specified across cultural, language and time zone boundaries [3]. Findings indicate major problems in communication and coordination (C&C) between these sites that have a negative impact on the requirements phase [2, 3].

Software development in global environments has a profound effect on Requirements Engineering (RE) activities [11]. Problems in requirements engineering phase are the major cause of software project failures [4]. 40-60% of all defects found in a system can be traced back to incomplete or inconsistent requirements [5]. Requirements Validation (RV) is the ongoing process of RE that aims to identify and correct all these errors early in the project life cycle [6]. Various activities are performed in this process to ensure correctness, completeness, consistency, non-ambiguity and feasibility of the requirements [6]. These requirements validation activities are affected to various levels by various constraining factors of a globally distributed software development environment.

The techniques widely used for validating requirements include prototyping, animations, inspections and natural language validation. The focus of this thesis is to evaluate these techniques in a globally distributed environment and analyze how they are affected by various issues of communication and coordination in GSD. A framework is proposed for evaluation and comparison of these techniques. Drawing upon this comparison, communication architecture is proposed to best deal with the communication and coordination issues affecting requirements validation in GSD.

1.1 PROBLEM DOMAIN

Requirements Validation is a communication-intensive activity and requires interactions between a diverse group of people including analysts, customers of the intended system and users in the problem domain [6]. These people differ on grounds of varying background,

skills and knowledge level; an increased amount of communication effort is required to bridge the semantic gap between them [7].

Carrying out this communication and coordination between the relevant stakeholders is a key challenge of Global Software Development (GSD) [3]. Communication and coordination are encumbered by various factors in GSD like geographical distance (making travel difficult), temporal distance (different time zones) and socio-culture distance (language and culture) [2].

Many of the challenges faced by GSD are attributed to inadequate communication between remote sites [9, 10]. As the team members are physically separated, frequency of communication between them drops off sharply [2, 8]. Insufficient and impeded communication makes it difficult to timely disseminate project knowledge, familiarize individuals with each other's work practices and to promote shared understanding between team members [8]. Infrequent and less effective communication also poses a challenge towards the process of coordination of task, activities and efforts by distributed teams [2].

There are no requirements validation techniques designed specifically to cater to the needs of GSD context. Traditional techniques for collocated development are employed, but no evaluation of these techniques exists to determine how they fare in the GSD environment specifically with respect to communication and coordination. Given the significance of Requirements Validation, it is important to analyze how its techniques are affected by communication and coordination problems in GSD and find a solution to make the process more effective in geographically distributed software development settings.

1.2 RESEARCH QUESTIONS

Requirements validation activities ensure that the developers are working with a set of requirements that are a true representative of users' intents [6]. Geographical distribution poses various challenges with respect to communication and coordination when these techniques are applied in GSD.

The aim of this research is to understand the communication and coordination problems encountered when requirements validation techniques for traditional software development are employed in GSD context and suggest a possible solution. Following are the questions that will be addressed by this research:

- How effective are traditional Requirements Validation techniques, with respect to communication and coordination, when applied in a globally distributed software development environment?
- What possible measures can be taken to improve the effectiveness of these techniques in GSD with respect to communication and coordination?

Firstly, a communication model is proposed to define communication paths that are essentially required between relevant project stakeholders and secondly, a guideline is provided towards use of appropriate combination of mode and media for enabling communication between the paths identified in the model.

To review, the major contributions of this thesis include: firstly a proposed framework for evaluating individual requirements validation techniques and analyze the extent to which they are affected by communication and coordination issues of GSD; secondly a comparative analysis of requirements validation techniques on basis of the proposed framework to determine the techniques that would fare best in a globally distributed software development environment; thirdly a proposed solution for tackling with GSD communication and coordination problems in employing the most appropriate requirements validation technique.

1.5 THESIS OUTLINE

The remainder of this thesis is structured as follows:

Chapter 2: The second chapter gives an introduction of the problem domain, which is Global Software Development. It discusses the motivations behind the practice of developing software by geographically distributed teams. It reports the various challenges that hinder the effectiveness of software development activities as found in the literature. Special focus is on communication and coordination issues of GSD, which are deemed as the major source of problems in GSD.

Chapter 3: The third chapter gives a brief introduction to widely employed requirements validation techniques. Moreover, it gives a broad overview of requirement engineering activities and how they are impacted by the issues of GSD.

Chapter 4: The fourth chapter discusses the framework proposed for evaluating requirements validation techniques. It performs evaluation of each individual technique. These techniques include prototyping, animation, inspections and natural language validation. A comparative analysis of these techniques is performed to gauge the strengths and weaknesses of each technique and find out the technique that fares best with respect to GSD communication and coordination issues.

CHAPTER 2

2 GLOBAL SOFTWARE DEVELOPMENT

Global software development (GSD) is increasingly becoming a common practice in the software industry [2, 12]. The management, development and maintenance of software have evolved from being concentrated at a single site to being distributed across multiple sites over the globe [8]. GSD is characterized by stakeholders from different national and organizational cultures, separated by geographical and time zone distance, collaborating by means of information and communication technologies to develop a software system [12, 13].

Significant advances in technology, especially the growth of internet, have made communication and collaboration between remote sites a practical option and have aided this practice of software development surpassing national boundaries [8]. Research reports a large number of software projects developed between USA and India as well as other continents like Asia and Europe [14]. According to [8], a study conducted in 2000 revealed that 70% of US firms have outsourced some kind of business process and that 203 of US fortune 500 companies engage in offshore outsourcing.

There are a number of business reasons and benefits that motivate companies towards global software development. These are discussed in section 2.1. Despite the advantages offered, geographically distributed software development poses a number of challenges. These challenges are discussed in section 2.2.

2.1 MOTIVATIONS FOR GLOBAL SOFTWARE DEVELOPMENT

Many technological, organizational, and economic factors have led to the increased globalization of development projects. Although this trend has been growing for the past few decades, now it has become more of a norm than an exception [15]. One of the most important reasons for organizations to opt for global software development is its potential to reduce software development costs [1, 8, 12]. The organizations make huge savings by moving parts of the development work to low-wage countries where the same work can be done for a fraction of the cost [12].

In global software development, developers are located in different time zones that allow organizations to increase the number of working hours in a day thus decreasing the software development cycle time and improving the time-to-market [1, 12]. This is known as the follow-the-sun [12] or round-the-clock [1] development. Through time zone effectiveness, organizations achieve longer working hours on the development projects. When one site finishes working on a project for the day, developers at the other site in a different time zone start working on it thus potentially achieving a 24-hour workday [12].

Limited pool of trained workforce is another one of the reasons that motivates organizations to hire talent in other countries [9]. GSD has the potential to provide access to a large pool of highly skilled workers [1, 8]. The scalability available to organizations as a result of larger labor pool with specialized skills enables them to increase the size of their development effort without bringing dramatic changes to the organization itself [12].

In global software development, various actors come from different backgrounds and organizations and thus have varying practices and procedures [12]. It is possible to get different views from different people since they come from different parts of the world and have their own ways of accomplishing tasks. These factors lead to increased innovation and shared best practices [12, 13].

Organizations form subsidiaries in other countries to achieve the business advantage of proximity to their customers [9]. This enables them to benefit from locating closer to the target market [12]. Additionally, local employees are culturally and linguistically closer to the customer and have better knowledge of local business conditions [1, 12]. The global demand for software products and services has led to many mergers and acquisitions across organizations as IT firms strive to enter new markets and complement their product lines [8]. The opportunity to capitalize on these mergers and acquisitions is another factor leading towards global software development [1].

2.2 CHALLENGES OF GLOBAL SOFTWARE DEVELOPMENT

Physical separation among project members poses serious challenges to software project completion [1]. These challenges if not tackled properly can render the effort of collaboratively developing software over multiple sites ineffective. Discussed below are the major issues and challenges of global software development.

Once the sites that will be participating in global development of software have been determined, deciding how to divide the work among them is extremely significant but quite difficult [1, 8]. This division is constrained by various factors including resources available at the sites, their level of expertise in various technologies and the infrastructure [1]. Distribution of work across sites has to be carefully executed to avoid increased need for communication and coordination between the multiple sites. The best approach would be to achieve low coupling between the distributed tasks so that the development teams can work as independently as possible [1, 9, 16]. If the architecture of the system does not support relatively independent modules, there will be coordination difficulties among the sites; the sites may hold conflicting assumptions that persist for much longer times than in collocated development [9].

To reap the benefits of GSD, effective knowledge and information sharing mechanisms are required [1] however, global context makes it difficult to seek out and integrate the necessary knowledge [2]. Insufficient knowledge and information management can lead to lost

opportunities of reuse that could otherwise have saved valuable time and cost [1, 8]. Change notification is critical for success of a software project [18]. However, communication gap between the geographically separated team members often results in delayed communication of issues and information, which has a negative effect on success of the project.

Process non-uniformities and mismatches among the development sites cause communication and coordination problems [9]. Varying processes for doing work and sharing project information can create confusion and misunderstandings [2]. Additionally, variation in definitions of important/crucial terms can cause conflict and mismatched expectations [1, 9]. Process differences also lead to problems in synchronization and system integration [1, 8].

The major cause of issues encountered in global software development, however, stem from communication and coordination problems between the software development teams in distributed settings. Since the thesis concentrates specifically on communication and coordination issues in GSD, they are described in detail in the following section.

2.3 COMMUNICATION AND COORDINATION ISSUES

The largest source of problems in GSD is issues related to communication and coordination across distributed sites [9]. Communication is identified as one of the root causes of challenges faced by software engineers [10]. This complexity of communication is better handled in collocated development; within a geographically distributed team, various factors like varying organizational and national cultures add layers of complexity to the process [10]. Difficulties in communication are an impediment to the process of coordination required for aligning efforts and successfully accomplishing project tasks [2].

Described below are the major causes of communication and coordination problems in GSD, identified from literature. Some are the direct cause of problem whereas others their consequence.

2.3.1 PHYSICAL DISTANCE

Geographic distance among team members profoundly affects the ability to collaborate successfully [2]. The primary reason for these problems is that many of the coordination mechanisms that are ordinarily used in a collocated setting are either disrupted or entirely absent in GSD [2]. Distance exacerbates the fundamental RE problems such as poor communication among stakeholders because it is difficult to enable ongoing and effective communication between development sites [14].

Spontaneous communication and coordination declines tremendously because of increase in physical distance between the sites [2]. Additionally, it is not always possible to arrange face-to-face meetings, which are the richest form of communication, because of travel time, airfares and busy schedules [14]. Because of lack of face-to-face and synchronous

backgrounds [9]. Messages may seem strange or even rude and are less likely to be immediately responded to [9]. Even if the people at the sites speak a common language, subtle differences in meanings introduced by differences in culture may still lead to problems [3, 8].

Different technical and domain vocabularies among the stakeholders separated by distance and different national and organizational cultures also cause problems [2]. Distance prevents a thorough understanding of these cultural differences by the various stakeholders [3].

Cultural diversity hinders shared understanding of both system requirements and project related issues [11]. Different cultures often have different values, beliefs and approaches to communication and problem solving and may result in developers having disparate interpretations of the same requirements without even realizing it [11]. These differences result in requirements being expressed using diverse terminologies and level of detail making it difficult to analyze them for consistency, conflicts and redundancies [3]. Cultural boundaries make it difficult for the requirement engineers, developers and users to form a consistent understanding of requirements, thus having a significant effect on the development of a shared mental model in the distributed team [3, 11].

Distance among the development sites increases the likelihood of diversity in their corporate cultures [3, 8]. Organizations can have different hierarchies and associated protocols for communication [8]. Both sites can differ in work practices and procedures followed and might need adjustment [17]. Moreover, the cultural setup usually results in different communication styles [1], which need to be discussed in advance to avoid any problems later in the project [17]. Distance however prevents the thorough understanding of these cultural differences between the distributed stakeholders [3].

Additionally, [8] reports that culturally divided teams are not as cohesive as local ones and this leads to poor trust among team members, poor cooperation and frequent conflicts.

2.3.4 SHARED UNDERSTANDING

Shared understanding is defined as a common knowledge structure held by members of a team that enables them to form accurate explanations and expectations for a task [11]. Shared understanding is a necessity for coordinating and aligning actions to achieve a common goal [11]. To operate efficiently and effectively, all members of a software team must have a shared understanding of project requirements [11].

Cultural differences, both functional and national, also pose a formidable challenge for achieving shared understanding among the various stakeholders in a distributed team [2, 3, 11]. Inherent difficulty of understanding requirements is exacerbated in GSD by both loss of context and loss of communication bandwidth [2].

Challenge of achieving a shared understanding is also attributed to added delays in communication and loss of informal communication between teams separated by geographical distance [11].

2.3.5 LACK OF TRUST

Lack of trust among distributed sites results in unwillingness towards open communication and sharing of expertise and project information [9]. Outsourcing arrangements can lead to mistrust of the other site because of fear of loss of intellectual property and the worry about one's job security [1, 9]. Lack of trust is exacerbated by infrequent interactions between the distributed team members [3] and the differences in organizational culture between sites [18].

2.3.6 LACK OF INFORMAL COMMUNICATION

Informal communication is very important for coordination of work and for learning the culture of an organization [3]. Developers not located together have very little informal and spontaneous communication [1]. Loss of informal communication or 'water fountain talk' challenges the goal of forming a shared mental model of requirements between the distributed stakeholders [11].

Lack of informal communication does not encourage unplanned contact between participants for sharing information even though it is a better means of communicating requirements change information since formal mechanisms like specification documents do not ensure a quick reaction by the appropriate party [9]. Moreover, documentation is not always up to date in organizations and cannot be relied upon for understanding the status of a project [18].

Absence of this informal communication hinders establishment of a good stakeholder relationship. Lack of personal relationships contributes towards lack of trust among remote stakeholders, which results in impeded communication [3].

Expertise identification is also a problem attributed to lack of frequent and informal communication [9]. Participants have limited knowledge of who is working at the other sites on what modules [3] and often problems lay dormant because it takes time to identify who exactly to initiate contact with to clarify a simple question [1, 9].

2.3.7 IT INFRASTRUCTURE REQUIREMENTS

Significant pitfalls arise from difference in infrastructure including network connectivity, development environment, test and build labs and change and version management systems, in the multiple development locations [9]. Since the communication among the teams mostly has to be carried out electronically, issues related to network connectivity and faulty internet can seriously hamper the communication and coordination which is indispensable in carrying

out software development activities. Other common issues include problems related to mismatched data formats, incompatible environments for instance different versions of same tools, varying platforms etc [1, 2].

CHAPTER 3

3 REQUIREMENTS ENGINEERING

Requirements Engineering (RE) plays an important role in the process of software development. It is defined as *"the systematic process of developing requirements through an iterative co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats and checking the accuracy of the understanding gained"* [6].

Requirements engineering is an activity that attempts to discover, capture and document needs and requirements of a set of different stakeholders for a software system. The output of this process is a set of requirements that are formally agreed upon by all the relevant stakeholders, including the customers, users and the developers. RE is one of the most crucial activities of software development. It is highly interactive and the communication and collaboration between various stakeholders determines the quality of the final product [6].

[6] describes a set of sub-processes involved in requirements engineering phase. These are requirements elicitation, requirements specification and requirements validation. These processes cannot be isolated; their relationship is depicted in the figure below.

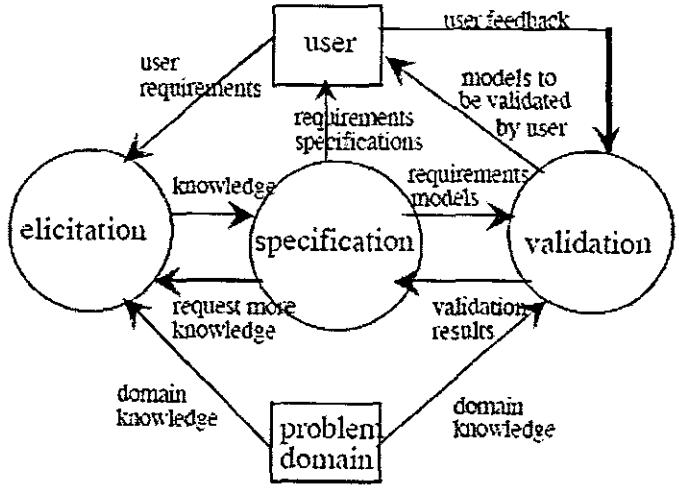


Figure 1: Requirements Engineering Process

3.1 REQUIREMENTS VALIDATION

According to [6], requirements validation is defined as *"the process of certifying the requirements model for correctness against the user's intention"*. It additionally defines requirements validation as a set of techniques which *"establish and justify our (and the user's) belief that the requirements model specify a software solution which is in conformance to the customer and user's intents"*.

Requirements Validation is an extremely important and ever-present activity that is carried out in conjunction with other requirements engineering activities including requirements elicitation and requirements specification [6]. The aim is to make sure that the software development team is working with a set of requirements that are a correct representation of what the customers and users actually require.

40-60% of all defects found in a system can be traced back to incomplete, inconsistent or ambiguous requirements [5, 6]. According to [5], cost of fixing a defect during system testing is about 90 times higher than fixing it during requirements analysis. Therefore, Requirements Validation is carried out with the aim to identify and correct these errors early on rather than later when the software will be designed and coded [6].

Requirements validation activities are performed to ensure the following desired properties:

- **Correctness:** The formalized requirements should reflect user intentions accurately, resulting in software system that behaves as expected [5].
- **Consistency:** There should be no internal contradictions in the requirements specification [5, 6, 21].
- **Completeness:** It should be ensured that the requirements model does not omit any essential information about the problem domain. [6, 21]
- **Non-ambiguity:** Requirements should be clear in the sense that they cannot be interpreted in more than one way [6].
- **Feasibility:** It is important to validate requirements to make sure they can be implemented in the given available budget and technology [21].

Depending upon feedback and validation results, the requirements model is modified until it corresponds to the user's expectations [6]. Since requirements have to validated against user and customer intents, their active participation is of utmost importance for success of the process [5, 6].

3.2 REQUIREMENTS VALIDATION TECHNIQUES

Various techniques are employed to validate requirements for different properties. Different techniques for validation may appear to be more suitable for some kinds of applications than for others. The widely employed techniques for validation are [6]:

- Prototyping
- Animations
- Inspections
- Natural Language

3.2.1 PROTOTYPING

A prototype is an operational model of the application system [22]. It is used to explore the essential features of a proposed system through practical experimentation before its actual implementation [26, 28]. It implements certain aspects of the future system thus only demonstrating selected parts of the desired system [22, 26]. Prototypes are categorized into throwaway and evolutionary and can be horizontal or vertical depending upon the nature of requirements that have to be implemented.

It is impossible to have a complete, consistent and fixed requirements document as a starting point for the technical construction of a software [22, 24]. Requirements change because with time individuals get a better understanding of what they actually want from the software system and thus are better able to articulate these intents [24, 27]. Prototyping assists in the discovery and validation process of these requirements [22, 25]. You can never assume that the requirements are valid. It is imperative that they be inspected by lots of eyes and from various different perspectives [25]. Prototypes are concrete, tangible and illustrative in nature and provide a basis for evaluating the written specification thus facilitating in bridging the gap between the requirements specification and the actual user' intents, needs and wishes [6, 22, 23, 25].

Prototyping assists in [6, 22, 26]:

- Clarifying any relevant specification or development problem.
- Evaluating accuracy of problem formulations.
- Serves as a basis for discussion between the developers, users and customers and aids in decision-making.
- Since it is an executable model of the written specification, it supplements it and provides a basis for bridging the gap between the specification and the actual user intent, needs and requirements.

With prototypes you do not need to rely solely on specification documents for validating requirements. The users are asked to validate something that is closer to their real experience

and thus they are more capable of detecting anomalies [6, 22]. Prototypes provide a tangible idea of the problem solution as opposed to a static requirements specification [22, 25]. Thus, it provides the developers, users and customers with a concrete basis for [22, 23]:

- Discussing requirements
- Clarifying problems
- Reaching decisions.

3.2.2 ANIMATIONS

Animation of a specification is *"the process of providing an indication of the dynamic behavior of the system by walking through a specification fragment in order to follow some scenario"* [29]. In its simplest form animation is a sequence of steps including gathering, processing, pictorial rendering, analyzing and interpreting of data [35].

Animation provides an interactive and user-friendly environment for validating requirements [32] and establishes a cooperative atmosphere between software engineers and customers for requirements understanding [34]. It bridges the communication gap between various stakeholders by providing complex information and concepts in a form that is tangible and easily comprehendible [30, 31, 33]. It promotes improved comprehension and shared understanding by representing the requirements in a more direct and engaging manner, permitting stakeholders to see requirements, gain awareness and assist them in making requirements related decisions [35].

A key factor for success of requirements validation is improved communication and shared understanding between various stakeholders [29]. Shared understanding is impeded by the need to browse through disjoint textual requirements and accompanying models [35]. However, animation facilitates this between the customers and the developers by providing information effectively via graphical representations that provide a closer match to the mental model of users than textual models and improve understanding [31, 33, 35].

With the help of visual validation, customers are able to identify shortcomings in specifications and suggest changes more easily as compared to textual information [34]. These visual artifacts can be as simple as hand-drawn sketches and as elaborate as advanced interactive experiences [35]. However, it is important to find a suitable graphical representation keeping in mind the type of information and the target audience, since untrained customers cannot be presented with abstract representations that might bewilder them and take them time to comprehend [33].

3.2.3 INSPECTIONS

Inspection is a highly collaborative technique used to uncover defects and find errors in software artifacts like requirements specifications [36, 37]. Inspections involve meetings that

benefit from the synergistic effect of a group of people working together and moving towards a common goal [36]. Inspections involve a formally defined process in which inspectors examine and record defects individually and then in the form of a group [37]. Inspections usually involve the following sequence of steps [37, 38, 39]:

- Planning: It involves identifying the target material, selection of appropriate participants (assigning roles) and making a schedule for the meeting.
- Overview: It involves providing the inspection team a brief introduction of the document under inspection.
- Preparation: It involves individual review of the document in which inspectors create their fault lists.
- Inspection: A group meeting, generally face to face, is conducted to discuss and find additional defects. An action list is developed for necessary activities to be performed.
- Rework: Documented defects are addressed by the author of the document.
- Follow-up: The moderator checks that all reported defects have been dealt with and that no new defects have been introduced. A re-inspection is scheduled if necessary.

Thus, the inspection process involves fault collection and discussion. It involves individual review of documents not constrained by place or time [38]. The discussion however is traditionally a same time and same place meeting [39].

3.2.4 NATURAL LANGUAGE

Natural language is a widely practiced technique for validating requirements [6, 40]. Analysis of problems usually starts from interviews with the customer or from available user documentation and both are heavily based on the use of natural language [40]. Even unorthodox techniques like observation of user's working environment or capturing users' work with a video camera require translation into natural language [40].

It is important for the customer to perform an informed validation of the requirements, which is only possible if he is able to understand the representations the analyst has employed for the requirements model [40, 41]. Usually the users have limited or no knowledge of these formal models. Thus, to establish a shared understanding NL has to be used [6, 40, 42].

The use of natural language to state requirements has several benefits [42, 43]:

- Natural language is universal; any type of requirement in any kind of application domain can be described.
- It is flexible; requirements can be described rather abstract or quite detailed.
- Finally, it is wide-spread; everyone can read and write such requirements.

3.3 GLOBAL SOFTWARE DEVELOPMENT AND REQUIREMENTS ENGINEERING

Requirements engineering is a task that is difficult enough when done in a traditional collocated environment but it becomes even more difficult when stakeholders have to specify requirements across cultural, language and time zone boundaries [14, 20].

Various issues of global software development negatively affect the process of requirements engineering. Lack of understanding, lack of awareness of local working context, reduced level of trust among stakeholders, impeded ability to share work artifacts relevant to development of requirements and cultural differences all challenge the effective collaboration of distributed teams [14, 20]. A model of how various issues of GSD impact requirements engineering is presented in [14] and depicted in figure 2.

Requirements engineering is a communication intensive task [19]. The main challenge of global software development on requirements engineering phase is the difficulty to communicate and coordinate [14, 19, 20]. RE requires coordination among application domains, interdependent modules, tasks and people [11]. Team members coordinate their tasks by communicating and developing a common understanding of the work, organization and its goals and knowledge about other players in the project; these activities however are more difficult to achieve in a distributed team than in a collocated one [11].

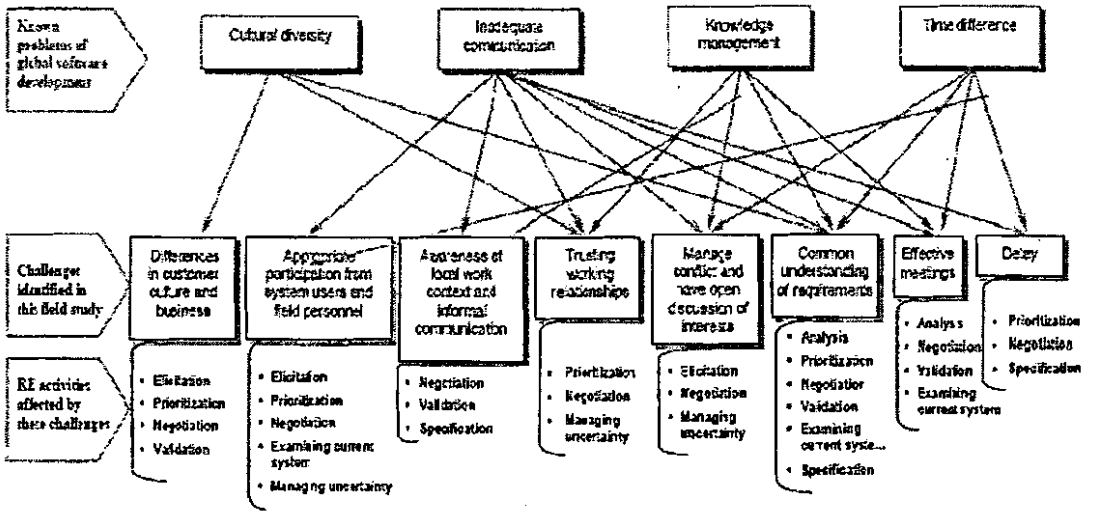


Figure 2: Effect of challenges of GSD on RE activities

CHAPTER 4

4 REQUIREMENTS VALIDATION TECHNIQUES IN GSD

RV is an extremely communication-intensive activity and requires interactions between a diverse group of stakeholders [6]. Relevant project stakeholders need to review the requirements to ensure accuracy, completeness and optimal level of detail to deliver software that truly meets the business needs. Given the differences in their backgrounds, skills and knowledge, increased communication is required to bridge the gap between them [7].

Improper communication is reported to be one of the main causes of defects in requirements [5]. These communication problems are exacerbated in GSD [20] and pose a challenge to the process of carrying out requirements validation activities. As mentioned earlier, no RV techniques exist that specifically cater to the needs of a GSD context. Traditional RV techniques are employed but there is a need to evaluate their effectiveness in carrying out validation tasks and find out how well they deal with the communication and coordination issues of GSD. For this purpose a framework is proposed to evaluate these individual techniques and then compare them to determine their levels of effectiveness. This framework is discussed in the next section.

4.1 FRAMEWORK FOR EVALUATING COMMUNICATION AND COORDINATION ISSUES IN RV TECHNIQUES IN GSD

We propose a four-dimensional framework, adapted from [7], to evaluate and compare various widely used collaborative RV techniques. The choice of dimensions for the framework is based on a review of the literature pertaining to communication and coordination problems in GSD, communication and coordination requirements for a particular technique and factors that enable and influence communication and coordination in a geographically distributed environment. This framework is presented in Fig. 3.

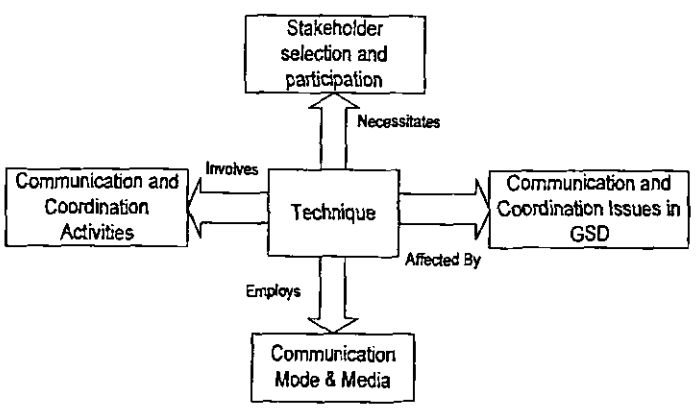


Figure 3: Four-dimensional view on technique (adapted from [7])

The following subsections elaborate on each of these framework dimensions, providing a brief description along with the rationale for including it in the framework.

4.1.1 STAKEHOLDER SELECTION AND PARTICIPATION

Over the years, many information technology systems have failed to deliver up to the required expectations of the users. Inadequate involvement of the users is cited as the major factor contributing to this shortfall between expectation and reality [44].

The structured design methods used for developing software systems require active user participation. Different users at different levels in the organization play different roles in the RE process. Each of these roles has a varied set of responsibilities that must be recognized in advance to mitigate the act of assigning people based purely on availability and convenience [44].

Selection of appropriate participants for validation purpose is a prerequisite to the success of the process. Communication will be rendered ineffective by the presence of inappropriate people whose contribution to the process will be limited [7]. Relevant users from appropriate levels have to be selected depending upon the type of requirements that are to be validated by a particular technique [44].

Some requirements might need a technical review for which end users might not possess the necessary skill; such a case would require expert judgments. There might be artifacts that require IT specialists and users both. Similarly, signing off requirements and making critical decisions would require the presence of managers with seniority and authority in the meeting. Therefore, to ensure valid assessment of requirements, right set of stakeholders from appropriate levels of hierarchy must be selected [44].

The different roles identified in [44] for users are:

- Top management role
- Middle management role
- User representatives
- End users

For effective requirements validation, contacting the appropriate roles in the remote stakeholder group is extremely significant [48].

Selection and participation of appropriate stakeholders is critical for the success of requirements validation. This identification is difficult to achieve in collocated software development and is exacerbated in geographical context. This is because of lack of face-to-face (f2f) meetings that help getting acquainted with different people who have varying expertise and skill sets at the distant sites and provides a basis for initiating contact with the relevant people at appropriate times.

4.1.2 COMMUNICATION & COORDINATION ACTIVITIES

Communication activities are those that involve formal or informal information exchange between all stakeholders in a project [7]. Identifying the C&C activities involved in a technique would assist in understanding the particular nature of these activities, the degree of collaboration it necessitates between the various stakeholders and the sort of mechanism that can be employed to best support these activities in a global environment.

4.1.3 COMMUNICATION & COORDINATION ISSUES IN GSD

The fundamental problem of GSD is that many of the mechanisms employed for communication and coordination in a collocated setting are disrupted [2, 17]. Communication problems exist in traditional forms of software development as well [45], but they are exacerbated by global distribution [2, 17]. Following are the factors that affect communication in a global development context:

4.1.3.1 PHYSICAL DISTANCE

Physical distance between sites profoundly affects the ability to collaborate [2]. There are fewer chances to contact people at the distant site and thus communication is much less frequent and much less effective [2]. Additionally, traveling difficulties, including traveling time and airfares, reduce the chances of face-to-face meetings that are the richest form of communication [2, 46].

4.1.3.2 LANGUAGE AND CULTURE

Socio-cultural distance (language and culture) also poses a challenge to communication and coordination activities in GSD [2, 17, 46]. When communicating in a non-native language there is a higher possibility of sending unintentional messages [46]. Language barriers affect the transfer of knowledge of requirements to system analysts [3, 47]. In addition, terms and concepts vary across different countries and organizations; therefore, special attention has to be paid on how things are expressed [46]. Cultural differences, both national and organizational, further exacerbate the communication problem in global context by increasing the chances of misinterpretations [2, 17, 46].

4.1.3.3 TIME ZONE DIFFERENCES

Different time zones place a burden on communicative relationships between sites [17]. Communication between the sites has to be carefully planned since the time zone difference reduces the window of synchronous communication between the sites [17, 46]. Direct communication, which is the means for quickest feedback, becomes an extra effort since it

has to be carried out very early in the morning or late at night [3, 17], thus straining the relationship between the two sites [46]. Time zone differences therefore delay the communicative process [17].

4.1.3.4 LACK OF INFORMAL COMMUNICATION

Informal communication plays an important role in trust relationship building and additionally, plays a significant role in resolving ambiguity [3, 46]. Lack of informal communication in global context is a cause of many problems including lack of context, unresolved issues, misalignment and rework [1, 3]. Formal communication is required for important decision-making meetings and requires a clear interface for sharing critical information [1, 46]. Network connection problems can cause delays and reduce the effectiveness of the meeting [46].

4.1.3.5 SHARED UNDERSTANDING

Shared understanding of requirements is hindered by various factors in GSD. These include infrequent or nonexistent f2f meetings [46], differences in national and organization cultures [2, 3], lack of context [3] and ineffective remote communication across sites [3].

4.1.3.6 TRUST

Collocated teams build trust through informal communication [3, 46]. Distance however is an impediment to building trust relationships since it makes informal communication impossible [2]. Lack of trust is constrains and inhibits effective communication since it results in irregular flow of information across sites.

4.1.3.7 IT INFRASTRUCTURE

In global development, communication channels are often supported electronically [17, 46] and therefore, incompatible IT infrastructure is an impediment to effective communication. Different operational platforms, incompatible requirements and development tools, application version problems and network connection and reliability issues obstruct the process of effective and timely communication [1, 17, 46].

4.1.4 COMMUNICATION MODE AND MEDIA

4.1.4.1 COMMUNICATION MODE - SYNCHRONOUS/ASYNCHRONOUS

Synchronous and asynchronous communication each comes with its own set of pros and cons. Synchronous communication ensures faster feedback and improved understanding [17]. However, language difference can create a problem in this mode of communication since it requires instantaneous response and does not allow time for reflection on one's response to a query. Asynchronous communication on the other hand allows time to think on a response and be careful in language and expression [47] but usually results in delayed feedback which can result in unresolved issues [17, 46]. Asynchronous communication also causes the problem of information overload [18], which can further delay response.

4.1.4.2 COMMUNICATION MEDIA

Selection of communication media for carrying out requirements engineering tasks has an impact on the effectiveness of the process [2, 48]. Different communication media is appropriate for carrying out different kinds of interactions [49, 50]. It is important to choose a media for communication depending upon the task, its complexity and its requirements for media richness [49].

There are various theories regarding media choice and its effectiveness. Social presence theory [49] predicts the media that individuals will employ for particular kinds of interactions. According to media-richness theory [49], task performance improves with the selection of a medium with appropriate richness. Media richness is a medium's information carrying capacity in terms of feedback, channel, source and language.

The types of information and communication technology include auditory, visual and textual media [50]. According to [50], a single medium is not enough for carrying out a particular task. Using complimentary modalities helps people best accomplish status, information and problem-solving tasks. Successive Information and communication technology i.e. mixed media is required for efficient and effective completion of tasks [50, 53].

4.2 EVALUATION OF RV TECHNIQUES BASED ON FRAMEWORK

The following sections present the evaluation of the widely used requirements validation techniques based on the proposed framework to gauge their effectiveness with respect to communication and coordination in global software development.

4.3 EVALUATION OF PROTOTYPING TECHNIQUE

The following sections provide an evaluation of the prototyping technique for requirements validation according to the framework dimensions.

4.3.1 STAKEHOLDER SELECTION AND PARTICIPATION

- For effective requirements validation, contacting the appropriate stakeholders at the remote site is consequential [48].
- Usually there is a multiple layer of stakeholders and the client communication might be routed through client's IT department and business community [48]. There is a high chance of introducing misinterpretations and misunderstandings in stakeholders needs at each communication level [3, 48]. Therefore, direct communication with the end users, who will be actually using the final software product, is necessary to ensure the adequateness of the prototype and thus the effectiveness of requirements validation [3, 24, 27, 48].

4.3.2 COMMUNICATION AND COORDINATION ACTIVITIES

- Requires Negotiation between the client and the developers [27] to prioritize set of elicited requirements to ensure that there is no gap between what the client wants and what the prototype can provide [28].
- It is important to pre-decide the level of detail and sophistication of the prototype and the tool that is going to be used for its development [58].
- The purpose of the prototype has to be made clear to the end-users right from the beginning to avoid any future misunderstandings [23, 24].
    - It might give users the unrealistic expectation that there would a complete and working system in a short time [24].
    - Given too much access to the prototype, end-users may equate the incompleteness and imperfections in the prototype with shoddy design [24].
- Negotiation is required to deal with unrealistic expectations on the users' part [24].
- User interaction with the prototype has to be properly monitored and controlled [23, 24]
    - The prototype might let the users visualize increased functionality without considering the cost of added features [24, 27, 28].
    - A visual prototype might result in the user asking for frequent changes [24, 27].
- A controlled setting is required to keep the user expectations at a reasonable level [24].
- Developers might want to administer the demonstration of certain requirements, which are more ambiguous and more likely to cause a conflict, themselves [24].

4.3.3 COMMUNICATION & COORDINATION ISSUES IN GSD

4.3.3.1 PHYSICAL DISTANCE

- While validating requirements by means of a prototype, F2F meetings help gauge the level of involvement and engagement of the users in the process [57].
- To manage any misunderstanding or misconceptions on the users' part the developers need to administer the process of validation personally [17]. However, f2f meetings are infrequent or almost absent in most cases because of airfares and travel time [13, 46, 59] and it is impossible to read the social cues that help gauge the co-participator's reaction and enable to respond accordingly [56, 57].

4.3.3.2 LANGUAGE AND CULTURE

- In global context, it is difficult to get the correct requirements communicated because of cultural and language differences [17]. Language differences result in ambiguities and misunderstandings. Different national and organizational cultures have different communication styles and ways of sharing information.
- However, prototyping provides a concrete basis for getting better understanding of the requirements from both user and developers' point of view [22].
    - They form a basis of discussion and aid in decision-making [22].
    - They work as an interaction medium between the clients, users and the developers [27, 28].

4.3.3.3 DIFFERENT TIME ZONES

- Time-zone difference presents a problem in carrying out communication [55] and places a burden on communicative relationships [17]. Same time communication becomes an extra effort since it will have to be carried out either very early in the morning or late in the evening [58], thus straining the schedule of at least one of the parties involved in the process.
- Quick feedback and appropriate suggestions from the users are necessary to avoid delay and rework [24]. In global context, this is possible through synchronous communication but because of time-zone difference, this window of synchronous communication is reduced [17].

4.3.3.4 LACK OF INFORMAL COMMUNICATION

- Lack of informal communication in global teams negatively affects relationship building, which is important in requirements negotiation and hence in requirements

validation [48]. This relationship building is also important in providing stakeholders with the context needed for disambiguating the meaning of particular requests [48].

- Lack of well established relationship does not encourage impromptu conversations that can play a role in enhancing the shared understanding and improved communication [48].

4.3.3.5 SHARED UNDERSTANDING

- It is difficult to establish shared understanding among a globally distributed team because of difference in national and organization cultures and native languages [22, 54].
- Prototyping, however, helps in establishing a shared understanding of the requirements [22] since prototypes are tangible and illustrative and help removing misalignment of understanding and inconsistencies that otherwise might go undetected in specification documents [22].
- Validation through prototyping can enhance the communication and cooperative nature of working between users and designers [7].

4.3.3.6 TRUST

- Misunderstandings and misconceptions frequently occur because of geographical, temporal and cultural distances [59]. These might cause mistrust and discord between the distributed teams [48], which result in reduced cooperation, and irregular information flow between teams [8].
- It is important to make clear the intent and purpose of the prototype. Any miscomprehension on the user's part on the size, sophistication or purpose of the prototype might lead to damaged trust and the associated problems [27].

4.3.3.7 IT INFRASTRUCTURE REQUIREMENTS

- Requirements validation through prototyping requires making a working prototype available to the client side over the distance. For this purpose IT infrastructure at the distributed sites has to be standardized [17] as use of different platforms at any levels (OS or development tools etc) can lead to incompatibilities [17, 55]. These incompatibilities are usually hard to find and time consuming to fix over the distance [17] and might cause additional costs [55].
- Reliability and usability of the network should be satisfactory [46].

4.3.4.3 COMMUNICATION MEDIA

Audio

- Requirements validation is a communication intensive activity and requires shared understanding between clients and developers.
- Noisy telephone lines with delays and echoes and possibly heavy accents can pose a serious challenge in communicating project critical information [17].
- Due to language differences and heavy accents, people prefer to avoid communicating important information over telephone conversations [17].
- If team members do not know the recipient very well they avoid using phone as a means of communication and prefer communicating through synchronous or asynchronous text messages [17].
- Through phone conversations, communication between the team members is neither recorded nor stored in a data repository, which might pose a challenge for future reference [18].

Video

- Videoconferencing can be used to accommodate for the loss of richness due to lack of f2f meeting, however, people are usually reluctant to use this tool [17].
- Many IT professionals shun videoconferencing because of various behavioral problems such as the awkwardness of interrupting or the inability to see all participants [60].
- However, videoconferencing has been found out to be effective for carrying out requirements validation in global context [48]. In addition, prototypes act as a means of enhancing the communication between the two parties [25, 27].

- Issues with network reliability and usability might present hurdles when demonstrating and validating the prototype by means of video conferencing [46, 55].
    - Slow network connection or frequent server failures cause not only frustration but increase the waiting time.
    - This requires ensuring presence of appropriate technology for carrying out the required communication at both sites.

Text

- Many offshore developers' written language skills often are much better than their spoken language skills [17].
- If instant feedback is required then chat can be used to communicate [17]. But with chat one loses some of the richness of tonal expressions from a phone call [17].
- If synchronous communication is difficult, email can be used to share the necessary information [17]. However, feedback via email might delay the project and slow down the development process [17].

- Asynchronous written communication (email and fax) automatically leaves a communication history [60], which can be referred to at later times.
- Too much asynchronous text communication can result in information overload and may result in dire consequences if a critical message is not responded to in time [18]. It can result in productivity loss and cause rework and delay [18].

4.4 EVALUATION OF ANIMATION TECHNIQUE

The following section evaluates the animation technique with respect to the framework.

4.4.1 STAKEHOLDER SELECTION AND PARTICIPATION

Customers who are funding the project, domain analysts and end users who will be actually using the software to carry out their job tasks all need to validate the visualizations to ensure that what they depict is in accordance with their needs and intents. They can report problems and detect opportunities for new features [62].

4.4.2 COMMUNICATION AND COORDINATION ACTIVITIES

- The visualized requirements have to be made available to the client for validation [32] so that they are able to identify if the developer's view of the system requirements coincides with their own [33].
- Exercising the requirements specification by means of an animation requires the users to provide various inputs and observe the resulting behavior [34]. Users might require developer's assistance in a few cases to understand certain system behaviors.
- Exercising the requirements enables the users to identify any shortcomings or describe missing or additional requirements [34]. Effective communication is required to comprehend these requirements to accommodate them.
- The visual artifacts developed as a result of this technique can be as crude as a hand-drawn sketch or as elaborate as interactive experiences [35]. The developers and the clients must decide upon the purpose of this process prior to incorporating it into the requirement engineering process.

4.4.3 COMMUNICATION AND COORDINATION ISSUES IN GSD

4.4.3.1 PHYSICAL DISTANCE

- Physical distance makes it impossible to carry out the demonstration of an animation in a f2f meeting. Temporal and geographic distances also create a problem and make communication less effective [59]. Narration describing the animation can be attached

to the animation or visualization [33]; it might not be a replacement for f2f communication but can provide with a good alternative as per needs of geographical distribution.

- Additionally, visualization technique in itself provides with an effective medium by means of which requirements of a system can be discussed, further explored and negotiated [30, 35].

4.4.3.2 LANGUAGE AND CULTURE

- Reliance on natural language for requirements specifications is one of the main sources of ambiguity; it is verbose in nature and thus enables incorporating irrelevant information [33]. Cultural and geographical issues exacerbate the communication issues related to natural language [33, 48]. However, visualization provides requirements information with graphics and animations along with textual explanations where required [33]. These pictorial representations are easier to understand [32, 33]. Additionally, improved understanding from simulations helps in clarifying requirement details [32] and provides a more tangible form for discussion as compared to purely textual documents that might contain details in ambiguous language promoting misinterpretation and misunderstandings [9].

4.4.3.3 TIME ZONE DIFFERENCES

- As is the case with other techniques, time zone difference makes it difficult to carry out synchronous validation of requirements through animation.
- Quick feedback from the users is required to ensure understanding and avoid rework and delay. However, with time-zone differences the window of synchronous communication is reduced [17].
- Even if the visualization is complimented by narrations at appropriate points [32] to enhance understanding and avoid assumptions, language difference, choice of words and terminology difference can create a problem [9]. It would be preferable to solve the issues as soon as they arise but that is not always possible because of temporal and geographical distance separating the user and developer sites [59].

4.4.3.4 LACK OF INFORMAL COMMUNICATION

- Informal communication plays an important role in building relationships, which play a role in understanding and disambiguating requests [48]. Improved relationship encourages open communication, which results in better requirements gathering, and relatively accurate requirement representation. Additionally, it develops patience among the team members and reduces the chances of misinterpreting and misconstruing comments on the artifacts while validating [9].

4.4.3.5 SHARED UNDERSTANDING

- A key obstacle to requirements validation is the lack of shared means of communication between the developer and the customer [34]. This communication gap exists because the customers and developers are usually ignorant of each other's respective domains and it is difficult to express and elaborate ideas that the other can understand [33].
- In global perspective, different cultures and native languages result in different perceptions that cause misunderstandings [9] and are a hurdle in the way of forming a shared understanding.
- By means of animation and visualization, information can be represented in a way that is easier to understand [33], that in turn enhances the communication between the various stakeholders [32, 33, 34]. Animation (complimented by narration) has a positive impact on the level of domain understanding attained by participants [32]. With reduced communication gap and improved domain understanding, it is possible to clarify understanding and promote shared agreement of the requirements [35].

4.4.3.6 TRUST

- Trust among stakeholders improves the requirements engineering process since it encourages the team members to openly discuss issues and listen to the other's point of view with patience [63]. It can encourage impromptu conversations that improve communication and help form shared understanding between the stakeholders [48]. Common understanding of requirements results in their accurate visual representations and thus reduces the chances of rework and delay.

4.4.3.7 IT INFRASTRUCTURE REQUIREMENTS

- The requirements visualizations and animations have to be made available to the client over the distance. Since the nature of animations varies from simple visuals to elaborately interactive experiences [35], it has to be made sure that they can be run over at the other site. Various tools are available that allow displaying an animation and interacting with it over the web [61].
- Even if MS-Office PowerPoint is used for creating the animation [32], version incompatibility can still cause a problem at the site. Hence, it is necessary to ensure that the other site has all the necessary requirements fulfilled for running the requirements model.

Text

- This technique makes use of pictorial representations for depicting system behavior [33]. Textual information is used at places where further clarification is required. However, according to [32] there is a higher level of understanding from narratives as compared to written text descriptions.

4.5 EVALUATION OF INSPECTION TECHNIQUE

The following sections provide an evaluation of inspection technique based on the proposed framework.

4.5.1 STAKEHOLDER SELECTION AND PARTICIPATION

- Defect detection rate through inspections is highly dependent on the personnel who take part in it; the more able the people who participate, the more effective the inspection [39].
- It is important to identify experts for inspection who are familiar with any notations or domain terminology etc used in the artifact under inspection [39]. This is important to make sure that the inspectors are able to contribute fully to the process. If they are unable to understand the conventions, they will have the additional task of getting acquainted with the application area and the notations to reach an acceptable level of fluency [39]. As a result, they will have less time to review the artifact.
- This requirement of specialized inspectors is better catered-for in asynchronous meetings. Since their domain expertise is called upon frequently, they can contribute in their own time [39]. This will also remove the blocking factor i.e. the delay caused in scheduling a meeting at a mutually suitable block of time for everyone to be able to attend [39].
- Moderator plans the meeting and identifies the inspectors for reviewing an artifact [39]. He has to have knowledge of the personnel available over distributed sites and a mechanism to identify the required expertise.

4.5.2 COMMUNICATION AND COORDINATION ACTIVITIES

- It requires assigning roles and responsibilities to the participants of the inspection [36, 37].
- Meeting place and time has to be scheduled and coordinated to ensure that all appropriate people can participate [36].
- It requires providing the participants of the inspection an overview of the artifact that is to be reviewed [64, 66].
- Reviewers individually studying the artifacts might require communication with the author/producer of the artifact to clarify some ambiguities or remove confusions [36, 64].

- Carefully moderated interaction is required to avoid any occurrences of interpersonal conflicts [64].
- It requires continual moderation to make sure discussion is not sidetracked [36, 64].
- The team must be able to coordinate their activities to ensure that goals of inspection are met and deadlines are adhered to [38, 39].
- Any modifications made or defects corrected have to be communicated back to the reviewers [39].
- If a re-inspection of the artifact is required, meetings again have to be scheduled [39].

4.5.3 COMMUNICATION AND COORDINATION ISSUES

4.5.3.1 PHYSICAL DISTANCE

- Inspections require simultaneous attendance and involvement of expensive human technical resources for reviewing an artifact [38, 64, 65].
- F2F meetings contribute more towards synergistic effect as compared to electronic media [36, 39]. However, these are not always possible in global context because of a number of reasons including high airfares and the travel time involved [13, 38, 46, 59] and the extensive task of coordinating and arranging a meeting that accommodates the schedules of all participants [38].
- However, inspections carried out through electronic media are less costly as compared to setting up f2f meetings [64, 66].
- Physical distance makes it difficult to better understand the justification behind a particular defect. Electronic media is not always effective in clarifying others' contributions [36].

4.5.3.2 LANGUAGE AND CULTURE

- Different organizational and national cultures have different communication styles, which might not be amenable across the multiple sites [18]. Different native languages hinder the process of effective communication giving rise to ambiguities and misunderstandings [9].
- Clarifying these ambiguities and confusions might take time when these meetings are carried out over distance [36].
    - The justification behind the defects might not be explained clearly [36].
    - Clarification provided might be equivocal, increasing chances of confusion and misunderstandings [36].

4.6.3.4 LACK OF INFORMAL COMMUNICATION

- Natural Language's inherent ambiguous nature calls for continuous communication between the clients and the developers to resolve issues in a timely manner to avoid rework and delay [48]. However, in GSD since informal communication is almost nonexistent [2, 3, 48, 75] and people find it difficult to identify people at the remote site and contact them for essential information [47].
- Informal communication is required to build a good working relationship; it ensures a better flow of information [47, 59] and plays a role in disambiguating particular user requests, providing with the context to better form an understanding of a requirement, its need and its priority [48, 78].

4.6.3.5 SHARED UNDERSTANDING

- Natural language is universal, flexible and widespread but inherently ambiguous [43, 70]. Words can have several meanings. With natural language does not always promote a shared understanding, stakeholders can have different interpretations of the same requirements [74] without even knowing [43]. Conflict in common understanding of requirements is also attributed to differences in both functional and national cultures [3]. This consequently results in a system that does not behave as intended by the users.
- The most recommended solution to ambiguity problem is the use of formal or semi formal requirements specification languages [43]. Translating natural language requirements into a formal representation might remove ambiguity by choosing one of the interpretations of that requirement but it does not mean that the chosen interpretation is the correct one [43].
- Reaching common ground and shared understanding requires uncertainty and equivocalilty reduction [79]. Uncertainty reduction is better handled by lean media while equivocality reduction is better handled by rich media [79].
- Hence, media also plays an important role in effectively getting across the intended meaning [50] and establishing shared understanding [79]. Therefore, great consideration has to be put in its selection depending upon the intent, content and urgency of the message.

4.6.3.6 TRUST

- In natural language, what might be considered common sense by one can be easily a surprise to a person coming from a different cultural background [40]. Often these inconsistencies and ambiguities go unrecognized causing delays, rework and often damaged stakeholder relationships [1, 18, 48].
- Any statement in natural language can be ambiguous; however, Requirement Engineering context helps in disambiguating it [43]. This context develops through

4.5.3.3 TIME ZONE DIFFERENCES

- There are specific areas of requirements that require extensive and timely communication to develop a common understanding [38]. However, time-zone differences hinder effective communication that is necessary to develop a shared and common understanding [17, 55].
- Resolving these issues asynchronously might take too long and there is a possibility that it might pass on equivocal messages [38].

4.5.3.4 LACK OF INFORMAL COMMUNICATION

- For effective inspection, it is imperative to have a team of members who have the expertise and domain knowledge required for inspecting the particular requirements artifact [39]. Although different social network tools are available, people usually make use of personal contact lists to recommend expertise in specific situations [13]. These contacts develop through informal communication [n]. Lack of informal communication therefore poses a challenge in identifying the required expertise for inspection purposes.
- Informal communication also plays a role in establishing good working relationship with other team members and improves interpersonal interaction [59]. This helps in understanding context of discussion, and thus reduces the chances of misinterpretations [9], and does not have a negative impact on the productivity of meeting sessions.
- Informal communication is also required to build trust among the members, which plays an important role in effective interaction [63].

4.5.3.5 SHARED UNDERSTANDING

- Shared understanding of the artifact under discussion is necessary to keep the inspection process productive. Any conflicts or confusions have to be taken care of so that rework can be avoided [46].
- It has to be made sure that the participants are all on the same page [13], however this is a challenge in distributed environment. Diversity in culture, education and communication styles can result in varied perception and understanding of the same problem [n, o], it is therefore imperative to ensure that everyone has a common understanding of a given concept.

4.5.3.6 TRUST

- Among unacquainted team members, initially there is a lack of trust [63]. Trust develops among members when they come to know each other [3] and when they

realize that they are working towards a common end [63]. Since there is a lack or absence of f2f meetings in a global context [13, 46], establishing trust presents an even bigger problem.

- Requirements validation is an extremely communication-intensive process [6], however lack of trust inhibits effective communication [63]. Inspection involves identifying defects and discussing them; it is important to convey the context of any comment as well lest they be misconstrued and give rise to bad feelings among the members [63].
- Lack of trust gives rise to uncharitable interpretation of other's behavior when disagreements arise [63]. This causes hard feelings and delay in clarification and conflict resolution [63].
- For effectiveness of the technique, it is important to encourage interpersonal relationship and trust building between the team members. This will provide them with patience when it comes to resolution of issues and hearing others out [63]. Informal communication plays a role in achieving this but its lack is one of the major challenges of global software development [9, 63].

4.5.3.7 IT INFRASTRUCTURE REQUIREMENTS

- Moderately fast internet connections are required to ensure timely communication among the members of the inspection team [38].
- Different platforms employed at the multiple sites might create a problem when it comes to sharing documents and other artifacts. Even if same applications are being used, version mismatch can still cause a problem [63].

4.5.4 COMMUNICATION MODE AND MEDIA

4.5.4.1 SYNCHRONOUS MODE

If a synchronous meeting is to be setup to carry out the inspection, then extensive communication and coordination between the sites is required to set up a time and schedule that is convenient to all the participants [38].

4.5.4.2 ASYNCHRONOUS MODE

- When using asynchronous media for inspection, it is not necessary to wait days before a meeting can be set up to accommodate the busy and conflicting schedules off all participants [38, 39].
- Asynchronous media minimizes some of the problems generally encountered in a f2f meeting for inspections:

- Parallel communication is possible by use of asynchronous media and there can be multiple threads of communication at any given time [36, 39].
- Simultaneous and parallel communication helps prevent production blocking [36].
- It reduces the potential for domination and enables a more even participation among the members. This increased group interaction results in increased group performance [36].
- The inspectors can explain themselves in whatever amount of detail that is required [38, 39].

- The electronic media used to carry out inspection in a distributed environment provides with the benefit of maintaining an automatic record of all communication and coordination [36, 39].
  - This material can be referred back to later when required [36, 38].
  - This also mitigates the process loss associated with failure to process contributions if they lose their focus during the meeting [36].
  - Documentation or commentary can be broadcasted throughout the group or directed to individuals as required [39].
- There are higher chances for information overload for the individuals participating in the inspection through asynchronous media [36].
  - This is because the asynchronous media allows them to participate simultaneously in multiple threads of discussions [36].
- In case of asynchronous communication, it is important to moderate the process vigilantly, to avoid digression and wastage of time in discussing irrelevancies [39, 64].
- At times response time of participants is slow which causes a delay in the overall process [38].
  - For this moderator has to issue deadline to keep the process on track [39].

4.5.4.3 COMMUNICATION MEDIA

Audio

- Inspection involves reviewing an artifact for its validity. Several problems occur when media that supports only audio conversation is used [63].
  - It is not possible to review a file or document collaboratively.
  - It is not possible to point to places in the document or scroll the other persons screen to a point of interest.
- Audio conversations prove to be less effective when between people who have different native languages [63].
  - It is usually hard to describe an important issue to another person in a second language.
  - These conversations relatively take a lot of time and effort on the participants' parts.

- Time zone differences can create a problem in establishing contact with the required person [38].
- Lack of visual contact leads to lowered level of awareness of presence and group behavior at remote sites [3]. Without a good facilitator, there can be problems in knowing who can be addressed regarding a particular issue and thus effective participation is reduced [3].

Video

- The necessary infrastructure required for videoconferencing sessions has an additional overhead. It is expensive to setup and maintain over remotes sites and its coordination over remote sites is often problematic [53].
- Even large companies usually have limited number of videoconferencing suites [67]. They have to be booked way before the meeting is scheduled. These meeting are also constrained by time since they require the simultaneous presence of participants in different time zones.

Text

- Individuals cannot monopolize airtime in asynchronous meetings as is possible in f2f meetings [64].
- It is easier to be diplomatic when employing asynchronous media for inspections, since comments do not have to be made instantaneously [64]. In addition, recorded data can be changed and reformatted as and when required [64].
- As defects are posted frequently at a single WWW location, inspectors might be tempted to indulge in premature discussion of the posted defects instead of pursuing their own individual inspections. This makes it necessary to circulate the defect lists only when all inspectors have posted their defect lists [39].
- When carried out by means of written communication, a database of all the messages provides an evidence for any formal assessment that would otherwise be solely dependent on opinion [39].
- With textual communication, it is possible to think and be careful in using language for the message and even research if necessary before responding [63]. It is also possible to attach corresponding material along with text messages [63].

4.6 EVALUATION OF NATURAL LANGUAGE TECHNIQUE

The following sections discuss evaluation of natural language technique based on the proposed framework.

informal interaction, trust and relationship building with other stakeholders that provide a person with an insight when interpreting another's statement [9, 63].

4.6.3.7 IT INFRASTRUCTURE REQUIREMENTS

- While sharing documents between sites, it is important to consider the applications being used to prepare them. Even if same applications are being used, version mismatch and incompatibility can still cause a problem [63, 67].

4.6.4 COMMUNICATION MODE AND MEDIA

4.6.4.1 SYNCHRONOUS MODE

- Although there are various internet technologies available to support synchronous communication, stakeholders still avoid it at times since lack of a personal relationship does not encourage impromptu conversations for resolving assumptions and conflicting requirements [48, 75].
- Electronic media makes it difficult to transmit unequivocal messages and the ambiguity in meaning causes multiple interpretations [3].
- For synchronous communication, the network has to be reliable and usable [46]. Frequent server failures or connections that are too slow can create frustration due increased waiting time. At times, the solution might be to switch to asynchronous mode [46].

4.6.4.2 ASYNCHRONOUS MODE

- NL is inherently ambiguous [43, 70]. For shared understanding of requirements in NL, constant communication between the clients and the developers is required to remove any ambiguity or inconsistency [3]. However, asynchronous communication reduces the possibility of a direct feedback [17, 46].
    - Time zone difference also adds a delay to the response time which can cause the feeling of 'being behind' or 'missing out' [47].
    - Non-native speakers might take long to respond to a query. They might take time to reflect on the question and make a thought-through response even though a quick feedback is preferable [47].
    - Issues can go unanswered or delayed because of information overload causing rework and thus damaged relationship [18, 48].

4.6.4.3 COMMUNICATION MEDIA

Audio

- Natural Language requirements do not provide a clear medium for discussion since NL is inherently ambiguous [42, 43]. The problems of oral communication will add to the already ambiguous requirements:
  - It is usually hard to describe an important issue to another person in a second language [63].
  - Different native language, hence accents that further create a problem [17].
  - Quality of the communication medium also plays a role in effective communication. Degradation of voice quality, noise and connection problems with delays and echo pose a serious challenge [17, 77].
  - Because response has to be made immediately, there is no opportunity to reflect and correctly phrase the reply in the other language, which might cause misunderstanding or misinterpretation [63].

Video

- Long meetings through videoconferencing are usually not possible. Companies have a limited number of video conferencing suites that need to be booked well in advance [67]. Therefore, these are limited to decision level meetings [67, 78].
- NL requirements require extensive communication for clarification; however, videoconferencing sessions are usually limited in time because of communication overhead. This precludes them from ad-hoc inter-site discussions [67].

Text

- When NL is used for requirements specification, it leaves much room for assumptions and uncertainties. Uncertainty is best removed by use of leaner media like email that focuses on factual -information rather than emotional cues [79].
- In written communication through chat or emails, it is possible to reflect and be careful in using language for the message and even research if necessary before responding [63].

4.7 COMPARITIVE ANALYSIS OF RV TECHNIQUES

The following sections perform a comparative analysis of the requirements validation techniques based on the dimensions of the proposed framework and discuss how each dimension affects the process of requirements validation in geographically distributed development environment using the particular RV technique.

4.7.1 STAKEHOLDER SELECTION AND PARTICIPATION

Selection and participation of right people is necessary to ensure effective communication in a process; however, a truly representative group of stakeholders is usually difficult to achieve because of several constraining factors that might exclude them from participation [7]. Varieties of roles have to be fulfilled by users at different grades in the organization [44]. Their responsibilities must be recognized in advance to mitigate the act of assigning people purely based on availability and convenience. Relevant users depending upon skills and functions must be selected [7, 44].

Prototyping: For validating a prototype, direct communication with the end-users is important [24, 27]. Indirect communication through multiple layers of stakeholders usually introduces misinterpretations and misunderstandings at each level [3, 18] and thus convolutes the communication.

Animations: Validation of requirements through animations would require the presence of stakeholders who have direct concern with the requirements set being depicted in the animation/visualization.

Inspections: Selection of participants for an inspection depends highly upon the type of artifact that has to be reviewed. The more able the people who participate, the more effective will be the inspection [39]. Therefore, the need is to identify personnel with the relevant domain expertise for reviewing the requirements artifact [39]. However, expertise identification is a much-reported problem of GSD.

Natural Language: Selection and participation of stakeholder when validating natural language requirements depends upon the technique being employed to read the requirements document i.e. ad-hoc reading or specific reading techniques [70, 71].

4.7.2 COMMUNICATION AND COORDINATION ACTIVITIES

Communication and coordination activities involve information exchange between the stakeholders [7]. Identifying the activities involved in a process would help in understanding their nature and hence assist in choosing the right communication mechanism to support them in a global environment.

Prototyping: Validating by means of a prototype requires demonstrating it to the relevant stakeholders. No matter how effective f2f meetings are for this technique, they are not always feasible in a global context. Depending upon synchronous or asynchronous means adopted, feedback has to be obtained from the users.

Animations: Animations like prototypes require making the visualizations available to the client for review [32] over the distance so that they can identify any shortcomings in the requirements or request additional requirements [34].

Inspections: Inspection is the most communication intensive technique of all requirements validation techniques. This communication presents additional problems when carried out in a distributed setting. It requires multiple participants for reviewing a requirement artifact. It involves identification and assignment of roles and responsibilities [36, 37], scheduling of meetings [36], communication between the reviewers and authors for clarification of ambiguities in an artifact [36, 64]. Additionally, communication between the participants has to be carefully moderated to prevent sidetracking and conflicts [36, 64].

Natural Language: Natural language is inherently ambiguous but cultural differences, national and organization, in a global environment lead to increased misunderstandings and misinterpretations. Hence, the need for communicating about these NL requirements increases because ambiguities have to be removed in a timely manner to develop a shared understanding and to come to an agreement.

4.7.3 COMMUNICATION & COORDINATION ISSUES IN GSD

4.7.3.1 PHYSICAL DISTANCE

Physical distance makes it hard to carry out f2f meetings. These meetings are either infrequent or almost absent in geographical contexts [13, 46, 59].

Prototyping: For prototyping, f2f meeting helps the developers to administer the process personally to handle any unrealistic expectations on the user's part [24]. This would help in managing the process and avoiding any misunderstanding that might lead to damaged trust and thus an impediment to open communication.

Animations: When using animations and visualizations, narrations can be attached at the appropriate places to describe all the important features [33]; this might not be a replacement of f2f meetings but provides with a good alternative in a distributed context.

Inspections: Inspections are carried out to take advantage of synergistic effect of people reviewing a requirements artifact together at a single place and time [36, 39]. Physical distance makes it almost impossible to carry out inspection in f2f meetings. Additionally, coordinating and arranging a meeting that accommodates the schedules of all participants is an extensive task [38]. Carrying out inspection electronically is less expensive than a f2f meeting but it is not always effective in clarifying others' contributions [36].

Natural Language: Requirements validation through natural language is the hardest in global context because with increasing physical distance, cultural differences, both national and organizational, also increase and there are greater chances of miscommunication and misinterpretation [46, 47]. Removing these ambiguities requires extensive communication between the site, however, when there is low proximity between participants the attempts at establishing communication and carrying out coordination activities diminish [76].

4.7.3.2 LANGUAGE AND CULTURE

In global context, it is difficult to get the correct requirements communicated because of culture and language [17]. Different cultures have different communication styles and ways of sharing information.

Prototyping: Prototyping provides with a concrete basis for better understanding requirements from both user and developers' point of view [22]. It works as an interaction medium [27, 28] and provides a basis for discussion and aids decision-making [22].

Animations: Visualizations provide requirements information with graphics and animations with textual explanations where required [33]. These pictorial representations are easy to understand and help in clarifying requirements details [32, 33]. They provide a more tangible form of discussion as compared to purely textual documents [32].

Inspections: During inspection meetings, different communication styles and different native languages can create a hindrance in effective communication. Justification behind defects might not be explained clearly and could promote equivocal messages, increasing the chances of confusion and misunderstandings [36]. This can cause loss of productivity of a meeting.

Natural Language: Misinterpreted requirements are the most common cause of errors in software [46]. In global context, misunderstanding of natural language requirements is exacerbated [3] because different backgrounds, perceptions and cultures result in multiple interpretations [18, 40, 41, 63]. Even if the involved parties speak the same language, different national culture and language nuances can result in different understanding of the same statement [3]. Corporate culture also poses a problem of expressing requirements in diverse terminologies (different technical and domain vocabularies) [2, 3, 47]. Different organizations might support different document structures and might employ varying levels of details adding to the overall difficulty of requirements communication in natural language [46]. Therefore, natural language requirements add a communication overhead.

4.7.3.3 TIME ZONE DIFFERENCES

Time zone differences make it difficult to carry out communication [55] and place a burden on communicative relationships [17].

Prototyping: For effective prototyping, quick feedback and appropriate suggestions from the users are necessary to avoid delay and rework [24]. However, time zone differences make it difficult because the window of synchronous communication is reduced [17].

Animations: Although pictorial representations through visualizations and animations assist in developing an understanding, it would be preferable to solve issues as soon as they are identified. This however is not possible because of temporal and geographical distance [59].

Inspections: When validating requirements through inspections, time zone differences make it hard to carry out a synchronous meeting. Extensive communication is required to set up a meeting that takes into consideration schedules of all participants [38]. Same time communication in these cases becomes an extra effort since it will have to be carried out very early in the morning or late in the evening [58], thus straining the schedule of at least one of the parties involved in the process.

Natural Language: Chances of ambiguities and misunderstanding are higher in natural language requirements validation [42] and thus increased communication between the two sites is required to establish a common understanding [11]. Temporal distance challenges this everyday communication and coordination [47].

4.7.3.4 LACK OF INFORMAL COMMUNICATION

Lack of informal communication in global teams negatively affects relationship building, which is important in requirements negotiation and hence in requirements validation [48]. This lack of well-established relationship does not encourage impromptu conversations that can play a role in enhancing the shared understanding and improved communication [48].

Prototyping: Prototyping technique for requirements validation provides a tangible and illustrative means for requirements validation. It has a positive impact on the level of understanding between the stakeholders. It enhances the communication and cooperative nature of working between users and designers.

Animations: Animations [32, 33, 34] like prototyping enhance user understanding and help in achieving a common understanding of requirements.

Inspections: Lack of informal communication in global context however poses several challenges for inspection technique. For effective inspection, team members with the required expertise and domain knowledge are required [39]. Although different social network tools are available, people usually make use of personal contact lists to recommend expertise in specific situations [13]. These contacts develop through informal communication [49]. Lack of informal communication poses a challenge in identifying the required expertise for inspection purposes in GSD. Informal communication also plays a role in building trust relationship between members, which in the collaborative nature of inspections is quite consequential. Additionally, it develops patience among the team members and reduces the chances of misinterpreting and misconstruing comments on the requirements artifacts while validating [9], thus preventing loss of productivity in meeting sessions. However, decreased or absent informal communication negatively impacts trust and relationship building in global environment.

Natural Language: Since natural language is inherently ambiguous, it requires extra communication between the clients and the developers to resolve issues in a timely manner and to avoid rework and delay [48]. However, lack of informal communication results in

limited knowledge about the relevant people to contact for disambiguating requirements. Additionally, lack of a relationship does not encourage spontaneous conversations between the two sites for resolving issues.

4.7.3.5 SHARED UNDERSTANDING

Diversity in culture, education and communication styles can result in varied perception and understanding of the same problem [49, 50]. It has to be made sure that all stakeholders are on the same page [13], however this is a challenge in distributed environment [48].

Prototyping: Prototypes are tangible and illustrative and help remove misalignment of understanding and inconsistencies that otherwise might go undetected in specification documents [22].

Animations: Animation and visualizations present information in a way that is easy to comprehend [33]. They have a positive impact on the level of understanding obtained by the participants [32] and thus it is possible to clarify understanding and promote shared agreement of the requirements [35].

Inspections: To keep inspection process productive it is important to have a shared understanding of the requirements artifact being reviewed to avoid delay and rework. Reaching common ground and shared understanding requires uncertainty and equivocality reduction [79]. Different language and varying communication and explanation styles can affect the development of shared understanding while carrying out inspection.

Natural Language: Natural language is inherently ambiguous, imprecise and incomplete [40, 42]. In global perspective, use of natural language for requirements communication exacerbates the problem of achieving a common understanding. Special attention has to be paid to the way things are expressed because when using a foreign language it is easy to send unintentional messages [46]. Differences in both functional and national cultures also cause a conflict in understanding [3].

4.7.3.6 TRUST

Requirements validation is an extremely communication-intensive process; however, lack of trust inhibits effective communication [63]. Misunderstandings and misconceptions frequently occur because of geographical, temporal and cultural distances [59]. These might cause mistrust and discord between the distributed teams [48], which result in reduced cooperation, and irregular information flow between teams [8].

Prototyping: For prototyping, it is important to make clear the intent and purpose of the prototype right from the beginning to take care of any unrealistic expectations [24]. Any miscomprehension on the user's part might lead to damaged trust [27]. Lack of f2f meetings

and informal communication in global context [9, 63] can make it harder to identify and mend these damaged stakeholder relationships over distance.

Animations: Trust among stakeholders can encourage impromptu conversations that improve communication between the stakeholders and thus a shared understanding [48]. Effective communication will play a role in the right depiction of requirements in visualizations and animations.

Inspections: Inspections involve identifying defects and discussing them; it is important to convey the context of any comment as well lest they be misconstrued and give rise to bad feelings among the members [63]. Lack of trust in a global context gives rise to uncharitable interpretation of other's behavior when disagreements arise [63]. This causes hard feelings and delay in clarification and conflict resolution [63]. For effectiveness of the technique, it is important to encourage interpersonal relationship and trust building between the team members. This will provide them with patience when it comes to resolution of issues and hearing others out [63]. Informal communication plays a role in achieving this but its lack is one of the major challenges of global software development [9, 63].

Natural Language: In natural language, what might be considered common sense by one can be easily a surprise to a person coming from a different cultural background, both national and organizational [40]. Often these inconsistencies and ambiguities go unrecognized causing delays, rework and often damaged stakeholder relationships [1, 18, 48]. For effectiveness of these requirements validation techniques in global environments, it is important to encourage interpersonal relationship and trust building between the team members. Trust among stakeholders will improve the requirements engineering process since it encourages the team members to openly discuss issues [63].

4.7.3.7 IT INFRASTRUCTURE

Standardized IT platforms are important to ensure successful sharing of requirements related artifacts [17].

Prototyping: Requirements validation through prototyping requires making a working prototype available to the client side over the distance. For this purpose IT infrastructure at the distributed sites has to be standardized [17] as use of different platforms at any levels (OS or development tools etc) can lead to incompatibilities [17, 55].

Animations: The nature of animations varies from simple visuals to elaborately interactive experiences [35] and it has to be ensured that the remote site has the necessary resources for reviewing the visual artifacts. It should be noticed that they are used as a complimentary technique to assist other forms of requirements validation [33].

Inspections: Inspection meeting carried out synchronously would require moderately fast internet connections to ensure timely communication among the members of the inspection team [38].

Natural Language: Different platforms employed at the multiple sites might create a problem when it comes to sharing documents and other artifacts. While sharing documents between sites for natural language validation of requirements, it is important to consider the applications being used to prepare them. Even if same applications are used, version mismatch and incompatibility can still cause a problem [63, 67].

4.7.4 COMMUNICATION MODE AND MEDIA

4.7.4.1 COMMUNICATION MODE - SYNCHRONOUS/ASYNCHRONOUS

Although there are various internet technologies available to support synchronous communication in GSD, stakeholders still avoid it at times since lack of a personal relationship does not encourage impromptu conversations for resolving assumptions and conflicting requirements [48, 75]. However, synchronous communication can assist in timely discovery of issues and problems as well as their timely resolution.

Prototyping: Validating prototype synchronously would help in being proactive and responding immediately to any perceived misconceptions or misunderstandings. This would ensure that problems do not become more convoluted because of the delay. However, for synchronous communication, the network must be reliable and usable [46]. Frequent server failures or connections that are too slow can create frustration due to increased waiting time. At times, the solution might be to switch to asynchronous mode [46]. Asynchronous communication would allow the users to evaluate the prototype in their own available time but would result in delayed feedbacks. Delayed feedback slows down the validation process and causes time and budget overrun [17, 24]. Additionally, delays in addressing problems can result in design problems that may be costly to repair later [24].

Animations: Nature of animations varies from simple visuals to elaborately interactive experiences [35]. Synchronous communication might be required at times to assist in enhancing users' understanding by providing useful explanations at the required times and obtaining feedback instantaneously. When requirements visualizations need to be validated asynchronously, they can be complimented by explanations of animation by the developer at appropriate moments and attached as a narration to the visual artifact [32]. These narrations can be paused, stopped and replayed as and when required and have been proven to play a role in promoting increased learning [32]. Moreover, through animation the user can be guided along to the appropriate part of the screen to consider it for evaluation and analysis. This verbal and visual information can make up for the loss of synchronous communication to some extent.

Inspections: Synchronous communication for inspection meetings in a global context poses the challenge of extensive communication and coordination between sites to set up a time and schedule that is convenient to all participants [38]. On the other hand, asynchronous communication for inspections has proved to be effective for global environment [38, 65, 66]. When using asynchronous media for inspection, it is not necessary to wait days before a meeting can be set up to accommodate the busy and conflicting schedules off all participants [38, 39]. Additionally, asynchronous communication minimizes some of the problems generally encountered in a f2f meeting for inspections. It makes parallel communication possible so that there can be multiple threads of communication at any given time [36, 39]. The inspectors can explain themselves in whatever amount of detail that is required [38, 39] without worrying about airtime as is the case in a synchronous meeting. The electronic media used to carry out inspection in a distributed environment provides with the benefit of maintaining an automatic record of all communication and coordination [36, 39]. However, because of increased opportunities for interaction and communication, there are higher chances for information overload for the individuals participating in the inspection through asynchronous means [36].

Natural Language: Natural language requirements do not provide a concrete means for discussion because of ambiguities introduced by language nuances, personal preferences towards use of language and cultural differences. Electronic media in addition makes it difficult to transmit unequivocal messages and the ambiguity in meaning causes multiple interpretations [3]. Synchronous communication would be helpful in obtaining quick feedback and responding immediately so that any inconsistencies in interpretation and understanding can be identified and dealt with in a timely fashion. On the other hand, asynchronous communication reduces the chances of a direct feedback [17, 46]. Misunderstandings and questions that would take just a moment to resolve can take days of back and forth communication when using asynchronous means to resolve an issue [17, 60].

4.7.4.2 COMMUNICATION MEDIA

Productivity and effectiveness of the communication process is affected by the type of information that has to be shared [7] and the media that is employed [2]. No one media can be sufficient to carry out all the requirements related communication for a technique. Employing a succession of media according to the particular context of communication helps in effectively catering to its various needs.

Prototyping: According to [53], videoconferencing is a viable option for validating requirements through prototyping in a global context. Videoconferencing can be used to accommodate for the loss of richness due to lack of f2f meetings [17]. However, issues with network reliability and usability might present hurdles [46, 55] when demonstrating and validating the prototype by means of video conferencing. Slow network connection or frequent server failures cause not only frustration but also increase the waiting time. This

requires ensuring presence of appropriate technology for carrying out the required communication at both sites.

Animations: Regarding the use of media for validating animations, it is possible to communicate important points of a static or dynamic visualization by recording the explanation and attaching it along with the graphical model [32] for validation by the user. Different narrations can elaborate the different points in the animation for assisting the user in understanding the model and can be transmitted with little additional cost [32]. According to [32] this provides with an engaging form of communication with the different stakeholders. Textual information can also be used at places where further clarification is required. However, according to [32] there is a higher level of understanding from narratives as compared to written text descriptions.

Inspections: According to literature, different media is effective in supporting different steps of an inspection. Use of asynchronous media followed by a synchronous inspection meeting has proved to be effective in global environment [53]. Asynchronous media like emails and chat can be used to discuss the requirements artifacts and clear any understanding issues. It is easier to be diplomatic when employing asynchronous media for inspections, since comments do not have to be made instantaneously [64]. Additionally, a database of all the messages provides an evidence for any formal assessment that would otherwise be solely dependent on opinion [39]. With textual communication, it is possible to think and be careful in using language for the message and even research if necessary before responding [63]. It is also possible to attach corresponding material along with text messages [63]. A synchronous meeting by means of videoconferencing or an online inspection tool can be used for discussing the defects identified in the artifact. Although videoconferencing is the richest media next to f2f meetings, the necessary infrastructure required for videoconferencing sessions results in an additional overhead. It is expensive to setup and maintain over remotes sites and its coordination over remote sites is often problematic [53].

Natural Language: When NL is used for requirements specification, it leaves much room for assumptions and uncertainties. According to [79], uncertainty is best removed by use of leaner media like email that focuses on factual information rather than emotional cues. In written communication through chat or emails, it is possible to reflect and be careful in using language for the message and even research if necessary before responding [63]. However, if asynchronous textual communication is used, even a small issue can take days of communication before the issue can be resolved. This would cause unnecessary delay and productivity loss. If issues are discussed by means of oral communication, quality of the communication medium would also play a role in effective communication. Degradation of voice quality, noise and connection problems with delays and echo can pose a serious challenge [17, 77]. Although videoconferencing is a rich media, it is used mostly for decision level meetings [67, 78]. This precludes them from ad-hoc inter-site discussions [67], which is necessary for removing the additional uncertainties and ambiguities in natural language requirements introduced by different native languages and other cultural differences [48].

4.8 SUMMARY OF EVALUATION OF RV TECHNIQUES

The following section provides a summary of the evaluation of requirements validation techniques in GSD based upon the dimensions of the proposed framework. Tables 6a and 6b summarize the results of technique evaluation. Table 5a and 5b describe the scale used for representing the level of affect various factors of GSD have on these techniques.

Scale	Represented by
Highly affected	HA
Moderately affected	MA
Mildly affected	MLA
Not Affected	NA

Table 1a: Scale for affect levels

Scale	Represented by
Necessary	N
Desired	D

Table 5b: Scale for media requirements

	REQUIREMENTS VALIDATION TECHNIQUES			
Dimensions	Prototyping	Animations	Inspections	Natural Language
Stakeholder Selection & Participation	HA	MA	HA	HA
Communication & Coordination Activities	MA	MLA	HA	HA
Communication & Coordination Issues in GSD				
Physical Distance	MA	MLA	HA	HA
Language & Culture	NA	NA	HA	HA
Time zone difference	MA	MLA	HA	HA
Lack of Informal communication	MLA	MLA	HA	HA
Shared Understanding	NA	NA	MA	HA
Trust	MLA	NA	HA	MA
IT infrastructure requirements	HA	MA	HA	MA

Table 6a: Evaluation of RV techniques with respect to dimensions relevant to communication and coordination

	REQUIREMENTS VALIDATION TECHNIQUES			
Communication Mode & Media	Prototyping	Animations	Inspections	Natural Language
Synchronous	N	D	N	N
Asynchronous	D	D	N	N

Table 2b: Evaluation of RV techniques with respect to dimensions relevant to communication and coordination

4.8.1 ANALYSIS OF EVALUTAION OF RV TECHNIQUES

The comparative analysis of RV techniques based on the proposed framework dimensions has attempted to offer an insight into how RV techniques are challenged by communication and coordination issues of GSD. Realization of these challenges can assist in identifying effective communication and coordination mechanisms, which can have a direct bearing on the success of the validation process.

Evaluation of the techniques done in previous section shows that techniques are influenced to varying degrees by the communication and coordination issues of GSD. It is observed that prototyping and animation techniques are affected to minimum extent by communication and coordination issues of GSD. The scale of affect for animation is less as compared to prototyping but as mentioned in section 4.3, it is used more as a complimentary validation technique than a primary one. Moreover, no real evidence of its use and experience in GSD was found in the literature.

Based on the evaluation, it is evident that natural language and inspection techniques are highly affected by GSD factors. Natural Language technique for requirements validation is affected to exorbitant extents. Different factors contribute towards this situation including the fact that natural language is inherently ambiguous and imprecise; chances of misunderstanding and misinterpretation are already high but they are exacerbated by GSD context of distributed teams communicating over national and cultural boundaries. Cultural differences manifest themselves in the forms of national and organizational cultures. These cultures differ on terms of native languages, communication mechanisms, communication styles, use of terminologies and media preferences. When using a non-native language for communication, it is easy to send unintentional messages. Given the differences in functional and national cultures, there is an increased chance of conflict in understanding and misconstruing meanings. Moreover, inconsistencies and ambiguities in NL can go unrecognized, lay dormant and have catastrophic repercussions. Lack of relationship, another challenge of GSD, does not encourage impromptu conversations among team members to resolve extensive conflicts in NL requirements. This can result in damaged stakeholder relationships and cause a further hindrance in effective communication. Additionally, electronic media increase the chances of transmitting equivocal messages. Because of its inherent ambiguous nature and need for equivocality reduction, NL necessitates excessive

communication for resolution of inconsistencies which is not always possible because of the time zone differences.

Inspections require a great degree of communication and collaboration for coordinating and setting up a meeting for multiple participants. Difficulty of identifying personnel with the required domain expertise, scheduling meetings that take into account and accommodate the busy work calendar of multiple people participating from different geographical places and thus different time zones, dissimilar and conflicting communication styles, varying terminology preferences and level of detail for communication all have an impact on the productivity of requirements validation through inspections. Asynchronous means of communication can help minimizing some of these problems by allowing participants of an inspection process to contribute in their own available time. However, this mode of communication is known to increase delays in response times. Moreover, extensive discussions on artifacts are likely to cause a communication overload, which can cause participants to lag behind in assessing and responding to comments and deliberations in an adequate timely manner.

Prototyping and animations require lesser communication and coordination activities as compared to inspections and NL. They provide a tangible and illustrative method for presentation and validation of software requirements, are closer to reality and thus easier to comprehend. They assist in achieving a common understanding of requirements, which for other techniques, like Inspections and NL, is a grave challenge given the influence of varying languages and cultures. However, prototyping requires extensive IT infrastructure support for making the prototype available at the remote site. Fast internet connection, video conferencing resources for demonstration and compatible tool support at the sites is required for successfully accomplishing the RV process. Videoconferencing resources however, are usually limited in number at a site and need to be booked well in advance. Additionally, these meetings are constrained by time and not every small issue can be discussed. Animations on the other hand might require varying degree of IT infrastructure support depending upon the sophistication of the visualizations. They could be developed in MS PowerPoint or an advanced tool specifically designed for creating animations. Here it would be important to point out the fact again that animations prove better as a secondary requirements validation technique given the nature, form and level of requirements they depict.

From the comparative evaluation of widely employed requirements validation techniques, it is concluded that prototyping proves to be the best choice for validating requirements in a GSD environment. It works as an interaction medium between the multiple stakeholders thus improving the quality of communication. It requires lesser communication and coordination activities as compared to inspections and NL that necessitate high interaction between the relevant stakeholders. However, it is of utmost importance to come up with an appropriate solution that addresses the issues and challenges that this technique faces in a GSD context so as to make the process fruitful and effective.

CHAPTER 5

5 PROPOSED SOLUTION

Communication and coordination issues are dependent on an intricate relationship of various factors. The form of communication between the distributed teams, direct or through an intermediary, and the media employed are important factors that influence the software development activities [82]. To deal with the communication and coordination problems encountered when using RV techniques in GSD, it is important to realize the significance of the above two factors and propose a solution that adequately deals with both of them. Therefore, the proposed solution consists of two parts: a communication model to define communication paths that are essentially required and secondly, a guideline towards use of appropriate combination of mode and media for enabling communication between the paths identified in the model.

5.1 COMMUNICATION MODEL FOR PROTOTYPING

Communication and coordination issues identified from literature, pertinent to use of prototyping as a requirements validation technique, highlight the need of a communication model that defines various stakeholder roles that are significant to the prototyping process and defines clear communication paths between them for effective collaboration. Existence plus increased awareness among the stakeholders about such communication model and what it necessitates would help in initiating contact and keeping relevant stakeholders adequately involved in the process for increased effectiveness.

[54] presents a model to define communication paths between various stakeholder roles for a prototyping environment. Depending upon different knowledge domains lead communication roles are defined to carry out the necessary collaboration with other roles. The model suggests a single individual as the client representative who has expertise over several knowledge domains. This, however, is not necessarily applicable to all different contexts of projects. Using this model would mean an individual representing the end users, domain analysts and the decision makers from the client side; he would have to filter information from these multiple roles and communicate them to the development team. There are high chances that the knowledge and information will be distorted while being communicated and transmitted over each layer of stakeholders. Since it is quite evident from literature that participation of relevant stakeholders is essential for success of requirements validation process, this model would not suffice in the context where direct involvement of stakeholder roles is irreplaceable.

Several other communication models, categorized as centralized, decentralized and hybrid, are discussed in [10]. Centralized communication models promote communication by routing information through specific individuals only i.e. there is limited or no direct communication between individuals; a single individual has the responsibility of communicating and routing the relevant information to other stakeholders. Decentralized communication models enable

communication paths between all the stakeholders resulting in a number of communication links and paths for information to flow. Hybrid models make use of both centralized and decentralized models for communication. For our context of communication model for successful prototyping, a centralized communication model would result in information being relayed via layers of stakeholders. This indirect communication, as already identified from literature, is one of the major reasons that prototyping efforts at times fail to cater to the actual user needs and requirements. Decentralized communication would result in everyone communicating with everyone else. Although this would ensure everyone's point of view being communicated, in distributed development it would be impractical to attempt to enable these links through electronic means of communication; they would generally result in communication overload and at times important information not being responded or catered to. The hybrid communication model fits best for a global software development; however, the model presented in this paper has been developed specifically to cater to the communication and coordination needs of geographically distributed development teams only. These teams collaborate specifically for sharing software related information and coordinating tasks pertinent to integration of software modules and other development activities. It does not identify paths for communication with other major stakeholder roles like domain experts, end users and decision makers from the client side. This model, therefore, cannot be adopted in the context of requirements validation by means of prototyping, where active participation of domain experts and end users plays a major role in success of the process.

5.2 PROPOSED COMMUNICATION MODEL FOR PROTOTYPING IN GSD

We propose a communication model adapted from [54] and enhance it to address the communication and coordination issues that exacerbate the problems of requirements validation through prototyping in GSD. This model is depicted in the figure 4 below followed by an explanation of the various components and rationale for including them.

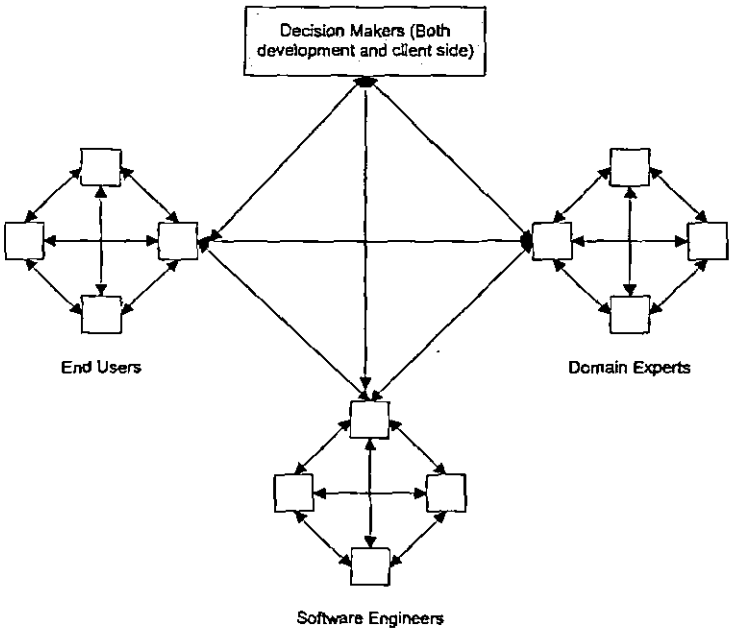


Figure 4: Communication Model for prototyping

The model is characterized by communication between various stakeholder roles through a lead role, which communicates with the general domain roles, filters information and relays it to the other stakeholder groups. The decision makers, managers from both the client and development side, need to spread the general awareness among the team members regarding importance of establishing these roles and enable communication between them for an overall successful project. Clear definition of these roles and their responsibilities will facilitate initiation of contact and continued communication over project information.

- This model stresses on fewer communication links so as to avoid miscommunication attributed to multiple conflicting opinions being transferred to other roles.
- Furthermore, by establishing communication links only with the representatives or lead-roles, difficulty of communication between large numbers of individuals across time zone differences is comparatively reduced.
- Decision makers from development and client side interact or communicate with the other stakeholder groups as and when necessary; their communication links exist when needed and not necessarily at all times.

Communication roles for the model have been identified through analysis of the literature on prototyping and determining its specific needs for stakeholder participation. We can only identify the broad stakeholder groups since it is impossible to define a useful, universally applicable set of stakeholder roles [81]. Further in-depth identification of stakeholder roles is contextual and varies from project to project. For prototyping, identification of stakeholders depends upon the content that characterizes the prototype, for instance information design, interaction design and navigation, visual design and system performance/behavior. However,

some studies provide a series of steps to follow in identification of roles like [80] and might prove useful in identifying stakeholder roles for a particular project.

A stakeholder is an individual who is materially affected by the outcome of the system or the project(s) producing the system [81]. These broad categories are given below. However, the actual list of stakeholder types for a project is context dependent and more concrete than this.

1. End-users: The actual users of the system.
2. Domain experts: Experts in a particular aspect of the problem or solution domain.
3. Developers: Designers, coders, technical writers, testers, and any other types of developer involved in the production and support of the system.
4. Decision-makers/Managers: The business managers, financiers, department heads, sellers, marketers, and other people who are investing in the production of the system. These stakeholders are only indirect users of the system or are affected only by the business outcomes that the system influences.

Each broad category consists of a lead-role representative who filters information and communicates with the representatives from the other roles. It is important to ensure that the broader stakeholder community is fairly represented i.e. the representative is aware of his responsibilities and importance of understanding and relaying the information from general role players to other stakeholder groups.

- The proposed model defines direct communication with end user and domain expert roles because direct communication and adequate representation of these roles have been identified as playing an important role in success of prototyping efforts.
- From literature, it has been identified that end users are not given a fair representation in validating the prototype. This model advocates a direct link with end users who will actually be using the software to carry out their daily job tasks and ensures that information and their points of view do not get miscommunicated over layers of other stakeholders that are employed to relay information.
- Periodic meetings between the identified roles can be used as a coordination mechanism for enabling collaboration among the distributed stakeholders for validation of the prototype. These can be prescheduled and included in the project plan or setup as and when required.

This proposed model does not specify the absolute communication links that have to exist for communication of RV through prototyping in GSD. It highlights the concept of establishing clear communication paths between the major stakeholders represented by a lead role. It can be enhanced to include communication paths between other stakeholder roles from knowledge domains like UI designers and usability specialists depending upon need and requirements of a particular project context.

5.3 COMMUNICATION MODE AND MEDIA

Another one of the issues identified for prototype validation is the need for direct communication between developers and the other stakeholders for purposes of:

- Personally administering user interaction with the prototype
- Obtaining quick feedback and responding immediately to any user miscomprehensions, misinterpretation or misunderstanding regarding the prototype concept or the requirements it represents
- To manage unrealistic expectations on the user's part and respond to any misguided expectations from the prototype

These problems and issues exist in traditional software development settings too but are exacerbated even more so in global development context. These issues call for f2f meetings to manage user miscomprehensions and manage their expectations. However, these are not always feasible in GSD because of high airfares, traveling difficulties and busy schedules. The need, therefore, is to decide on the mode that is next best to f2f meetings in dealing with these issues.

5.3.1 COMMUNICATION MODE

Keeping in view the above stated requirements, synchronous communication combined with an appropriate media would help in making up for loss of communication richness and effectiveness of f2f meetings to some extent.

- Ensuring synchronous communication would ensure timely feedback from the user and make it possible to give an adequately quick response to handle any miscomprehensions or obtain further clarifications.
- However, given the time zone differences, synchronous communication can place a burden on communicative relationships. It is of utmost important to realize in advance the impact and consequences of time zone difference and ensure that participants understand that one or both sites might be required to compromise occasionally on the meeting schedules to make synchronous communication possible for validation. This would help in avoiding hard feelings among the team members in the future.

5.3.2 COMMUNICATION MEDIA

According to [53], videoconferencing is a viable option for validating requirements through prototyping in a global context. Videoconferencing can be used to accommodate for the loss of richness due to lack of f2f meetings [17].

- It would help gain their level of involvement and engagement in the process (possible to point/guide attention to particular parts of the prototype visually).
- However, it is important to make sure that both sides have the required resources for carrying it out (issues with required technology, network reliability and usability might present hurdles).
- Personal preferences towards media usage should also be considered. Some people might find it difficult to communicate over videoconferencing; some other form/media of communication might be used as an icebreaker in the beginning.

Besides video-conferencing, other synchronous means like teleconferencing or live chat can also be used although they provide a level of communication that is less rich than videoconferencing. However, the difficulties in arranging these resources might make these media the next best option as compared to asynchronous validation over emails or through documents.

CHAPTER 6

6 VALIDATION OF PROPOSED SOLUTION

The proposed solution for facilitating communication and coordination, while employing prototyping as a requirements validation technique in GSD, is validated through a case study. The following sections give an introduction to the selected case, methods of data collection, the communication and coordination issues encountered in the project and discussion on how these issues could have been avoided by application of the proposed solution.

6.1 SELECTION OF THE CASE

The case study chosen is a project developed by a software development company that offers its services in areas of desktop and web programming, graphic and web design, customized database applications and outsourcing to a wide range of industries and business areas both in local and international market. The name of the company, however, is not mentioned here because of confidential reasons.

Selection of this particular software development project, as a unit of analysis, for the case study is because of its suitability to our context i.e. a software system developed by teams separated by physical, social and time zone differences making use of electronic media as the primary source of communicating system requirements and other project related information. The sources of requirements were the customers and clients for the system based in UK, whereas the development team was based entirely in Pakistan. The teams encountered various problems because of GSD, which are discussed in later sections.

6.2 DATA COLLECTION

Data for the case study was collected from various sources to back up interpretations from multiple sources as opposed to considering a single source. This triangulation i.e. drawing the same conclusion from multiple sources of information, adds credibility to the results and the conclusions. The data sources employed for this case study include interviews with the various stakeholders, study of their communication via emails and related documents.

6.2.1 INTERVIEWS

Interviews were used as a direct source for gathering the relevant information from stakeholders who were involved in this software development endeavor. The interviews were semi-structured and included both open and close ended questions. The questions were designed with the specific intent of extracting issues related to communication and coordination that the project teams encountered in the software development process, the effect of use of media for communication and the hardships of developing a software where

there is a lack of face-to-face communication for project meetings. The major findings from the interviews were summarized to the relevant stakeholders in the end to gain a feedback and avoid any misunderstandings regarding interpretation of comments and points of views.

6.2.2 DOCUMENTS

A number of documents were also analyzed to study patterns of communication between the distributed teams since documents were also at times used for providing feedback to the software development team and validating requirements. These documents were studied to gauge the level of effectiveness of this medium for communicating requirements related information between distributed teams.

6.2.3 EMAILS

Email was a widely used communication medium between the two distributed teams and therefore another source for our data collection. The requirements related communication between the distributed teams members was analyzed to study the type of information communicated via emails, its effectiveness and efficiency in successfully transferring the relevant knowledge to the receiver, promotion of shared understanding and the average time taken by the receiver to respond to the sent messages and information.

6.3 BRIEF HISTORY

The chosen project is a student assessment manager, referred to as product A in the remaining document, which a web-based system provided to clients in United Kingdom (UK), as a solution to their prevailing problem of managing student profiles and their assessments based on various factors in their local secondary school setup.

The project commenced in June 2005 with an estimated time of 6 months as the total completion time. Initially most of the requirements for the system were communicated to the development team by means of emails and documents. These sources, however, necessitated back and forth communication without promoting much common understanding of the requirements between the teams. The project manager for development of PRODUCT A at Pakistan site suggested a few sessions of chat as a proposed medium for eliciting and understanding initial requirements. The teams also held netmeeting sessions on a few Saturdays for discussing the prototype and obtaining feedback. However, use of electronic means in place of face-to-face communication for understanding and discussing requirements had an adverse affect; the project overran its time and cost budgets. Moreover, the prototype failed to satisfy user needs and requirements.

Face-to-face meetings at this point were decided as irreplaceable and a representative visited the development site for a week. The face-to-face meeting sessions were very effective in

resolving inconsistencies, clarifying outstanding issues and reaching a consensus on the communication procedures that would be followed in the future. The first version for the software was considered product A0.

A new software development team was established for the development of Product A1. New communication protocols were developed to effectively deal with the communication and coordination problems encountered in product A0. The developed communication protocols added by the experiences of the previous team helped in achieving a common understanding of requirements and thus in achieving the project objectives in a more timely and cost effective manner.

The subsequent sections discuss the major communication and coordination problems encountered, and how they hindered the process of achieving project objectives.

6.4 COMMUNICATION AND COORDINATION PROBLEMS

6.4.1 NATURAL LANGUAGE ISSUES

Use of natural language for communicating system requirements proved to be ambiguous and necessitated more communication between the distributed teams for gaining a common understanding. Phrasing of sentences and requirements often communicated multiple meanings and thus misinterpretations. These had to be restated and clarified with increased back and forth communication because of asynchronous means.

- The development team sent the user site a series of requirements related queries through an email on October 25, 2007. The queries were stated in natural language (English). Although the user site provided a timely response to the queries i.e. the same day via an email, the response was not entirely clear and another email had to be sent out to further elaborate on the requirements being discussed. The development site again had to send a detailed email to confirm that they have the right understanding of the requirements. This was confirmed the next day i.e. October 26. The user site again sent out another document on October 27 to correct some errors in its earlier email response. Natural Language requirements therefore have a high possibility of introducing ambiguities and causing multiple interpretations of the same statement. They necessitated back and forth communication to develop understanding and reaching a consensus.

6.4.2 IT INFRASTRUCTURE REQUIREMENTS

Various problems of IT infrastructure resulted in problems or delays in achieving project objectives. These are given below.

Requirements were communicated by means of emails and attached documents. Initially incompatibility of application versions used at the two sites created a problem. The user site sent requirements related documents to the development site on July 17, 2005. One of these documents was a Visio file, which was incompatible with the versions available at the development site. This problem was uncovered on 18th but an email was sent to the user site regarding the file format on 20th of July. The developer side had to request the document in a different format and later installed the version used at the client side. However, there was a hefty delay of 3 days in communication of these requirements caused by this incompatibility.

6.4.3 DIFFERENT DOMAIN TERMINOLOGIES

A communication gap existed between the stakeholders because of their varying fields of knowledge. The clients/domain experts used different domain terminologies and abbreviations for describing the requirements, which had to be clarified by the development site repeatedly. There was a huge need of extensive communication between the two sites to gain a common understanding of requirements and correctly get the intent and meaning of terms and requirements to the other team. These queries, which could be clarified instantly, were delayed since an asynchronous means of communication i.e. email was being used that took at least a day in being answered.

- Requirements related document was emailed to the user site on September 9, 2005. The document contained queries for clarification and confirmation of certain requirements including request for elaboration on certain domain specific terminologies (e.g. terms and abbreviations), that the development site did not understand. Because of a lack of response the user site had to be reminded again on September 12. The response was finally received on September 13. Because of delayed feedback and lack of response the inconsistencies and queries that could have been clarified within minutes took four days for complete understanding.

6.4.4 TIME ZONE DIFFERENCE

Time zone difference, according to an interview with the project manager at the development site, presented with both pros and cons. The development site was five hours ahead in time zone. On one hand, it enabled them to finish off the work in accordance with the feedback received from the user the previous day, and on the other hand it prevented them from contacting the users in case of ambiguities or further need of clarification. Time zone difference between the two sites made it difficult to hold synchronous sessions for discussion of system requirements. Several synchronous meetings, chat sessions and teleconferences, were held that had to be scheduled either late night for the development team or early morning for the client side. These meetings and sessions proved effective in better understanding of requirements and quick resolution of outstanding issues, however, because of time zone difference they were few in number.

- Given the difficulty in establishing a shared understanding between the users and the development team because of back and forth communication, the latter suggested a synchronous chat session for gaining a better understanding and forming a consensus on requirements. An email was sent to confirm timings on September 6, 2005. The meeting however had to be setup late night, at 11pm, for the development team because of the time difference between the two sites. Although these chat sessions were few in number they were reported by the stakeholders to be more effective than carrying out communication entirely through emails. They helped in resolving unclear requirements and obtaining instant feedback.

6.4.5 ASYNCHRONOUS COMMUNICATION AND FEEDBACK DELAY

Emails, an asynchronous means for communication, were used as the primary means for communicating requirements related information. It caused various problems, which are listed below:

6.4.5.1 DELAY IN FEEDBACK

The issues, unclearness or ambiguities, which could have been resolved in an instant through a phone call or other synchronous means, took at least a day for being resolved over an email. Delayed feedback also sometimes resulted in valuable time of developers going idle. Asynchronous communication via emails usually resulted in a delay since the emails were checked by the receiver in their own available time. Several such incidents were identified in the case study:

- An email regarding requirements was sent by the user site dated July 17. It was checked on July 18th and responded to on July 20th after the realization that the file format is incorrect. The issue that could have been solved instantly by synchronous means took three days to resolve.

- A working prototype for the software was made available to the users on November 3, 2005. Since the prototype was validated by the users in their own time, they provided their feedback a few days later. The first feedback was received via document attached with an email on November 6. Feedback from another user was received on November 11. These feedbacks were three and eight days late respectively, and contained conflicting comments and requirements for the system. The development team responded to these comments on November 14. Therefore, between the upload of the prototype, its validation and comments, and its response spanned total eleven days. This time could be saved to a tremendous extent via synchronous validation of the prototype that would have made it possible to respond to comments and conflicts in a timely manner with quick resolution of issues.

6.4.5.2 BACK AND FORTH COMMUNICATION

Natural language communication combined with an asynchronous mode i.e. email, necessitated increased communication between the remote teams for gaining a shared understanding of requirements. Clearing up terms and meanings required back and forth communication over emails that caused a delay plus a communication overhead.

- The development site emailed a set of queries to the user site on November 8, 2006. The feedback to it was provided through a document attached with an email on November 9. The development site reviewed the provided answers and feedback but some requirements still needed further clarification. Thus, an email was sent for further clarification on November 10. These queries were replied to on November 11. On November 14, the user site provided some further clarification on previous queries and asked for further elaboration regarding the previous set of requirements discussed. The queries related to requirements therefore, necessitated back and forth communication between the team members to enhance understanding and create a shared vision on requirements.
- In another event, the development site requested requirements related data from the user site by sending out an email on April 9, 2007. The email was responded to on April 12; however, it required further clarification, which was requested on April 15. This back and forth communication was necessitated by the asynchronous mode of communication that would otherwise be resolved in a few minutes via synchronous means.

6.4.5.3 LACK OF RESPONSE

Sometimes the emails were not responded to at all. Reminders had to be sent out to the relevant people for a quicker response. This usually caused a delay from one to three days.

- The development site sent the users a document containing queries regarding the requirements on September 9, 2005. There was no response from the user site for three days and they had to be reminded again on September 12. The queries were responded to on September 13 after this reminder. There was a gap of total four days in this requirements related communication because of asynchronous means.
- In another incident, a reminder email had to be sent to the user site on March1, 2006 to respond to an email sent earlier on February 27. There was a two day delay in response which resulted in wasted time for the development team.

6.4.5.4 CONFLICTING FEEDBACK

Relevant stakeholders validated the prototype asynchronously in their own time of convenience. Separate feedbacks were provided by the user side in emails with a gap of few days between them. They were both conflicting feedbacks; one of the users declaring the user interface appropriate while the other wanted it to be made simpler.

- Conflicting feedbacks to the prototype, made available to the users on November 3, 2005, were received on November 6 and November 11. Issues which could have been resolved if the prototype was validated synchronously with various stakeholders present at the same time took over 10 days for discovery and resolution. However, it would be important here to consider the fact that users might need time to gain a better understanding of the requirements and hence, changed or conflicting requirements with the passage of time; this is not necessarily solvable merely through an appropriate mode and media.

6.4.5.5 MIX-UP OF EMAILS

There were occurrences of mixed up emails where requirements related documents had to be resent thus causing a delay. Moreover, because of extensive requirements related communication carried out through the emails, certain replies and responses were mistakenly missed by a site, which caused a delay and need for reminding the other site.

- An email for requirements related information was sent to user site on February 27, 2006. The user site responded to the queries on March 1. However, there was a mix up of emails because of misunderstanding or extensive emails and the development site missed it; they reminded the user site for a response. The user site forwarded the

email reply on March 2. Because of mix-up of emails an unnecessary delay was caused.

6.4.5.6 INACCESSIBLE PROTOTYPE

The prototype was at times inaccessible to the user side (login issues). Asynchronous communication through emails for notification of such errors delayed the process of prototype validation.

- At one time, the users were unable to access the uploaded prototype at the link provided by the developers in an earlier email. An email was sent out to the developers regarding the issue on March 14, 2006. Synchronous communication as opposed to asynchronous means would have ensured a timely response to user's problem.

6.4.6 UNDEFINED STAKEHOLDER ROLES AND UNCLEAR COMMUNICATION PATHS

There was no explicit recognition of roles and responsibilities of various stakeholders for the project in the beginning. This resulted in unclear communication paths. Emails were used as the primary source of communication; the lack of defined roles at times caused relevant stakeholders to be missed out of the email loop causing a misunderstanding and an unnecessary delay in communication of requirements information.

- After a telephonic conversation on August 8, 2005, the user site was to send a requirements document. The project manager had not received this file by August 16 and emailed the users for the relevant information. On this reminder however, it was found out that the document had already been sent out but somehow the relevant stakeholder was missed out of the email loop. This caused an unnecessary delay and misunderstanding between the teams, since the information was not communicated to the project manager on time. This issue can also be attributed to unclear definition of stakeholder roles. Lack of recognition of these roles resulted in increased chances of missing out relevant stakeholders from the email loop. Recognition of roles of various stakeholders in advance would have ensured their inclusion in all relevant requirements related information plus it would have assisted in avoiding any such delay and misunderstanding.
- The development site sent a list of queries to the user site as an attached document with an email on September 9, 2005. A reminder had to be sent on September 12 to respond to the queries since they were causing a delay in the development process. There was a lack of recognition of user roles and knowledge as to who is to be contacted for what information and validation, the user had to route the request for response to another user who then made a reply on September 13. There was indirect

communication between the team members that caused a delay of four valuable days. Since there was not a clear definition of roles and communication paths, this delay could not be prevented.

- According to an interview with the project manager, for Product A0 there were multiple points of communication between the two distributed teams; multiple domain experts were contacting and communicating with the development team. There were, therefore, multiple points of views and conflicting requests for requirements from user side that had to be discussed and responded to; this took extra time and effort to resolve the inconsistencies and reach a mutual point or consensus on the multiple conflicting requirements. Moreover, since there were multiple people from user site communicating requirements, no one took responsibility when it came to dealing with the repercussions of changed and conflicting user requests.

6.4.7 INDIRECT COMMUNICATION

In the beginning, because of undefined stakeholder roles it was unclear as to who should contact whom for the relevant requirements information. Information was routed through layers of stakeholders to the relevant people. Requirements related information in the beginning was directed to the higher management of development side.

- An initial requirement document was sent by the user site to the higher management on development site on July 17, 2005. This document was forwarded to the project manager on July 18, which was checked by him on July 19. Due to version incompatibility, however, the project manager communicated back to the higher management regarding the issue on July 19 who then contacted the user site for a varying file format on July 20. In this indirect communication following problems were identified:
    - There were layers of communication between the two sites. Because of lack of knowledge regarding stakeholder roles, the user site and the project manager were indirectly communicating. This caused an unnecessary delay of 4 days in the resolution of the problem. Direct communication was established afterwards but the lost time resulted in waste of valuable time and lack of productivity.
    - Reliance on email, an asynchronous medium for communication, resulted in wasted time since the above emails were responded to the day next to sending date.
    - Indirect communication increased the communication overhead since it necessitated relaying of information between multiple layers. Additionally, it caused a delay in the requirements process.

6.4.8 UNREALISTIC USER EXPECTATIONS

The intent of prototype was not communicated to the clients right from the beginning. The users were anxious about the parts of the application that were inactive. Although, the reason was clarified then, they were still skeptical. According to an interview, initially the users were concerned about limited functionality of the prototype, which could have been avoided by making the users aware of the intent and purpose of the prototype right from the beginning.

6.5 VALIDATION OF PROPOSED SOLUTION

6.5.1 PROTOTYPING FOR REQUIREMENTS VALIDATION

Natural language is inherently ambiguous and imprecise. Although, initially requirements have to be obtained in natural language format, it is usually advisable to build a working prototype as soon in the development cycle as possible. As observed in the case study natural language requirements necessitated increased back and forth communication for resolution of issues. Increased reliance on this form of requirements would result in misinterpretations and an obstacle in reaching a shared understanding of requirements.

However, a working prototype was developed early in the life cycle, which helped in achieving the following objectives:

- Promoted shared understanding between the stakeholders.
- Provided a tangible means for discussing requirements as compared to relying solely on natural language for requirements specification.
- It helped in identifying missing and incomplete requirements which otherwise would have gone unnoticed in a natural language document.

6.5.2 NATURAL LANGUAGE ISSUES

Excessive dependence on email and natural language as a medium for communication in A1 necessitated increased need for back and forth explanations. This caused delay and a communication overhead.

- Communicating requirements in natural language even for a prototype is inevitable. For the product A, if these requirements were communicated asynchronously, as proposed by the solution, via phone call or even live chat, the requirements that took from October 25-October 27 to clarify could have been clarified and understood the same day. This would have saved 2 days spent in back and forth communication for confirming shared understanding of stated requirements.

6.5.3 IT INFRASTRUCTURE REQUIREMENTS

Difference in version numbers of softwares, as discussed for the case study, can cause unnecessary delays. Asynchronous communication added to this can cause further delay in resolving these issues of incompatibility.

- The requirements document was provided to the development site via email on July 17th. This was checked the next day. Even if there was an incompatibility in document because of varying version of the same software, if the problem was relayed to the user site via a synchronous means, for instance an instant message or voice mail, the three-day delay could have been avoided thus saving valuable time for both the teams.

6.5.4 DIFFERENT DOMAIN TERMINOLOGIES

A communication gap is likely to exist between the users and software engineers. Extensive communication is required to gain understanding of the varying domains. This communication carried out over asynchronous means, especially in natural language as was the case for product A, will create a communication overhead since it will require back and forth correspondence. It would be preferable to carry out this communication over a synchronous means, like chat or phone call, which would ensure a timely response and immediate understanding of any domain specific terms and abbreviations.

- In the case study discussed, if the queries were asked synchronously instead of through email, the delay caused because of lack of response could have been avoided. Moreover, better understanding of the requirements could have been obtained instantly.

6.5.5 ASYNCHRONOUS COMMUNICATION AND DELAYED FEEDBACK

Use of asynchronous mode for communicating requirements related information caused various problems including delayed feedback, increased need for back and forth communication for enhancing understanding and resolving issues, missed or mixed up emails and lack of response.

Various communication and coordination problems encountered by the distributed teams, listed in the previous section, would be better dealt with and could have been avoided at length by employing synchronous means for requirements validation.

6.5.5.1 DELAY IN FEEDBACK

By synchronous means, feedback from the users, regarding requirements, could have been obtained immediately. This would enable them to act on the feedback in a timely fashion rather than letting the time between a query and its response going idle. The synchronous sessions for requirements discussion were in few in number for the case study. However, even these few sessions helped in achieving a better understanding of requirements. They helped resolve outstanding requirements issues and in obtaining quick feedback and further clarification of any ambiguities.

- By use of synchronous means, the feedback requested on July 17th would have immediately responded to without the delay of three days that occurred in replying to the email.
- By use of synchronous means for validation of the prototype, the span of eleven days for receiving feedback from multiple users could have been avoided. Additionally, synchronous validation from multiple users at the same time would better enable the development side to tackle any conflicting opinions and comments. Furthermore, it would give them the opportunity to negotiate and reach a consensus in the same meeting instead of communicating back and forth via emails. Therefore, the hefty delay of eleven days could have been avoided.

6.5.5.2 BACK AND FORTH COMMUNICATION

Discussion of requirements in natural language over an asynchronous means i.e. emails resulted in ambiguity, which required increased communication between the distributed teams.

- By employing synchronous means for obtaining feedback from the user site on November 8, 2006, level of understanding could have been improved so that there would be a decreased need for communicating over the same requirement back and forth. Any further questions could be asked and responded to instantly thus saving the six days time and effort otherwise necessitated by emails. This would also have resulted in better managing the communication overload caused by extensive emailing.

6.5.5.3 LACK OF RESPONSE

- In the case discussed, emails went un-responded at times and a reminder had to be sent. This caused an unnecessary delay and wasted time.
- In the case discussed, the two instances of un-responded email caused a four and three days delay. Synchronous means of communication would remove the problem of

these unanswered emails since they would enable obtaining instant feedback and responses to the queries.

6.5.5.4 CONFLICTING FEEDBACK

Conflicting feedback from users caused a delay for the case discussed.

- Since validation was carried out asynchronously, the feedback from the multiple users came by a difference of five days time. The feedback was conflicting and took time to resolve. If the validation was carried out synchronously, then any problems that were encountered could have been discussed or negotiated at the same time to reach a solution/decision that satisfied all relevant stakeholders. This would have saved the time between November 3 and November 11, 2005.

6.5.5.5 MIX UP OF EMAILS

Asynchronous means usually result in a communication overhead so that emails get mixed up, unanswered or mistakenly unsent. This could also have been avoided by carrying out the necessary validation activities synchronously.

- By synchronous means, the email response sent on February 27' 2006 would not have been missed. A reminder would not be necessary and thus the two days delay could have been avoided.

6.5.6 DEFINING STAKEHOLDER ROLES AND COMMUNICATION PATHS

Misunderstanding and unnecessary delay was caused, in the case discussed, because the distributed teams proceeded without a clear recognition of stakeholder roles. There was a lack of awareness among the team members as to who is to contact whom for requirements information. Furthermore, lack of definition of roles and responsibilities resulted in extra time for relaying requirements related information via layers of stakeholders to the relevant individuals. Because of unclear roles, there were occurrences of relevant stakeholders being missed out of email loops by mistake that caused a delay and misunderstanding.

The above stated communication and coordination problems could have been avoided by application of the proposed model:

- If the roles and responsibilities for various stakeholder groups were recognized well in advance, it would have promoted clear communication paths between them. The significant requirements related email that was sent out to the higher management by the user site would have been directly sent to the project manager, without a layer in between. The removal of this additional layer would have prevented relevant

individuals from being missed out of the communication loop and the unnecessary delays and wasted time could have been avoided.

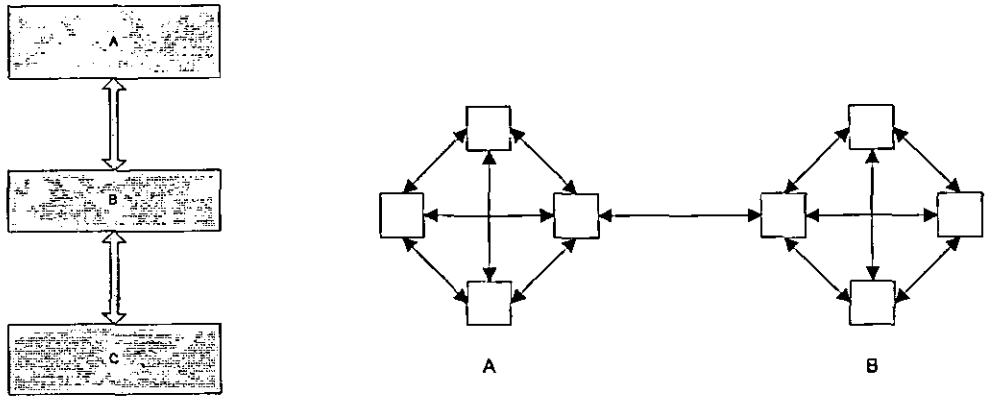


Figure 5: Communication between stakeholders before and after application of proposed communication model

- Establishment of lead representative role and direct communication links between them, as indicated by the proposed communication model, would have promoted the practice of communicating requirements related information directly between the stakeholders without any intermediate layers. By setting up the proposed communication model, the email sent on September 9, 2005 would have been sent directly to the relevant person. Prior identification of stakeholder roles and responsibilities and establishing clear communication paths between them would have helped in creating an awareness among the team regarding who is to be included in the requirements related communication; this would have helped in initiating contact directly with the relevant person and assist in determining who should be contacted for what sort of information. This would have prevented delay caused by expertise identification and ensured that relevant information is relayed to the relevant stakeholders. Hence, the four days delay caused because of a lack of lead role would have been avoided in this case by application of the proposed communication model.
- Creating lead representatives for various stakeholder roles would have discouraged multiple communication paths between the distributed teams. This would have ensured direct communication between the relevant individuals, and a single point of view being transmitted from one team to another. This way conflicting requirements and their repercussions could have been avoided; the lead role would be responsible for communicating the needs of his other stakeholder group members.

6.5.7 INDIRECT COMMUNICATION BETWEEN STAKEHOLDERS

Indirect communication between the stakeholders caused misunderstandings regarding timely communication of requirements a few times in the project life. Since in the beginning the requirements documents were emailed only to the management at the development side, they

were not timely relayed to the development team manager. Because of indirect communication via layers of stakeholders, an unnecessary delay was caused.

The above mentioned problems could have been avoided by application of the proposed model which suggests direct communication between the relevant stakeholders to discuss the system requirements and hence avoid any misunderstandings and miscomprehensions.

- This would have assisted in directing the requirements related information to the relevant people without introducing layers of stakeholders in between. Layers of stakeholders usually result in delayed communication of information to the relevant people. This would have promoted earlier recognition of problems and inconsistencies and therefore, their earlier resolution. The requirements related document emailed on July 17, 2005 would have been sent directly to the development team lead, with a direct communication between the two stakeholders regarding incompatibility of the document. The communication carried out via emails wasted four days, which would be resolved in a single day by the application of the proposed communication model. Moreover, this same communication carried out synchronously would have saved valuable time as well.

- If the proposed communication model was applied, the domain expert would have direct contact with the development team instead of directing requirements related information through the higher management. This would have prevented the misunderstanding that the user side is lagging in providing the requirements information and would have saved the valuable time wasted because of the unnecessary delay.

6.6 DISCUSSION AND ANALYSIS

Various communication and coordination issues, as reported in literature pertaining to software development in globally distributed environment, were identified in this case study. These problems were identified from interviews carried out with the project team members and by an analysis of their communication carried out via emails. As observed from the communication patterns in the email and reinforcement from interviews, it is established that the major pitfall in achieving project goals for Product A was the communication breakdown. The physical distance made it difficult for the two distributed teams to setup a face-to-face meeting to carry out all requirements related communication. Excessive reliance on asynchronous means for communication, in addition to lack of face-to-face communication, was a huge hindrance in establishment of a shared mental model for the project and its requirements. This eventually led to wasted time, cost and a system that did not satisfy user needs.

These communication and coordination problems, as discussed in the previous sections, are the problems related to communication of requirement in natural language, different domain

terminologies, IT infrastructure requirements, use of asynchronous communication media, lack of recognition of stakeholder roles and responsibilities and indirect communication between various stakeholders. By application of the proposed solution, for requirements validation through a prototype in GSD, most of these issues can be appropriately dealt with. Definition of stakeholder roles and responsibilities in initial stages of the project would help dealing with the issues of indirect communication and assist in establishing clear communication paths between the various stakeholders. This would also help in initiating contact with the right people for project related information thus saving both time and effort. Lead roles for stakeholder group would ensure limited communication paths and therefore, lesser chances of conflicting requirements and unsatisfactory results. Use of asynchronous means as discussed earlier would help in resolving the issues of delayed feedbacks, lack of response, mixed up emails and back and forth communication.

As discussed in the previous sections, communication breakdown in the project resulted in a system that failed to satisfy user needs. It necessitated face-to-face meetings between the users and the developers, which according to the software project manager helped in clarifying at least 70% of the requirements. From here on the requirements were better understood; moreover increased use of synchronous communication helped in validating the prototype on time and obtaining the feedback without an extensive delay. The project manager for Product A1 also insisted on limiting the points of communication between the two distributed teams. Through this single, reliable communication link, the requirements and feedback were communicated back and forth. This resulted in saving both time and effort required to solve conflicting issues and reach consensus on a set of requirements.

CHAPTER 7

7 CONCLUSION

This chapter discusses an overview of the research reported in this thesis, the contributions of the thesis and possible directions for future research.

7.1 OVERVIEW

Requirements validation activities ensure that the developers are working with a set of requirements that are a true representative of users' intents. It is a communication-intensive activity and requires interactions between a diverse group of people including analysts, customers of the intended system and users in the problem domain [6]. These people differ on grounds of varying background, skills and knowledge level; an increased amount of communication effort is required to bridge the semantic gap between them [7].

Carrying out this communication and coordination between the relevant stakeholders is a key challenge of Global Software Development (GSD) [3]. Communication and coordination are encumbered by various factors in GSD like geographical distance (making travel difficult), temporal distance (different time zones) and socio-culture distance (language and culture) [2].

Various techniques are employed for the purposes of requirements validation; widely applied techniques include prototyping, animation, inspections and natural language techniques. These techniques, however, face challenges when used in a global software development environment where teams are separated by geographical, social and cultural distance and collaborate over these national and cultural boundaries to meet software project needs and objectives.

The widespread use of geographically distributed resources for software development, the significance of carrying out requirements validation techniques and the need for appropriate communication and coordination mechanisms for dealing with the problems encountered is the basic motivation behind carrying out this research.

For our research purposes, we have performed an extensive literature review of global software development specifically in reference to communication and coordination issues of global software development. In light of these problems, we have evaluated the widely employed requirements validation techniques on basis of a proposed evaluation framework to find out how well they fare in global context, their strong and weak points in such a development environment, and their particular needs or requirements for improving their effectiveness. Based upon the results of evaluation, we propose a solution to best deal with the issues that are a hindrance towards achieving the maximum possible technique effectiveness.

The following section highlights the major contributions of this research.

7.2 CONTRIBUTIONS OF RESEARCH

The major contributions of this thesis include: firstly a proposed framework for evaluating individual requirements validation techniques and analyze the extent to which they are affected by communication and coordination issues of GSD, secondly a comparative analysis of requirements validation techniques on basis of the proposed framework to determine the techniques that would fare best in a globally distributed software development environment, and thirdly a proposed solution for tackling with GSD communication and coordination problems in employing the most appropriate requirements validation technique.

The framework for evaluating requirements validation techniques, with respect to communication and coordination in GSD, has been developed by reviewing the literature pertaining to communication and coordination problems in GSD, communication and coordination requirements for a particular technique and other factors that enable and influence communication and coordination in a geographically distributed environment. Development of this framework was necessitated by the fact that there are no RV techniques designed specifically to cater to the needs of GSD context. Traditional techniques for collocated development are employed but no evaluation of these techniques exists to determine how they fare in the GSD environment. Therefore, this framework assists in finding out the extent to which these techniques are affected by GSD and to discover their strengths and weaknesses in GSD relative to communication and coordination.

The comparative analysis of RV techniques based on proposed framework dimensions has attempted to offer an insight into how RV techniques are challenged by communication and coordination issues of GSD. Realization of these challenges can assist in identifying effective communication and coordination mechanisms, which can have a direct bearing on the success of the validation process. Evaluation of the techniques shows that they are influenced to varying degrees by the issues of GSD. From the evaluation, prototyping proves to be the best choice for validating requirements in a GSD environment. It works as an interaction medium between the multiple stakeholders thus improving the quality of communication. It requires lesser communication and coordination activities as compared to inspections and NL that necessitate high interaction between the relevant stakeholders.

A solution has been proposed in this thesis to enhance the effectiveness of prototyping efforts in GSD environment, specifically in terms of communication and coordination. Firstly, a communication model has been proposed to appropriately deal with the problems and issues identified from literature. This model defines communication paths between the relevant stakeholders that are essentially required. This model takes into account the major user groups for defining communication links between the stakeholders but depending upon need and context it can be enhanced to cater to the needs of that particular project and situation. Secondly, the proposed solution provides a guideline towards use of appropriate combination of mode and media for enabling communication between the paths identified in the model. This solution has been validated by means of a case study. A number of communication and

REFERENCES

8 REFERENCES

[1] James D. Herbsleb; Deependra Moitra; "Global software development"; Software, IEEE Volume 18, Issue 2, Mar/Apr 2001.

[2] James D. Herbsleb; "Global Software Engineering: The Future of Socio-technical Coordination"; Future of Software Engineering, 2007, Volume Issue, 23-25 May 2007.

[3] Daniela E. Damian, Didar Zowghi, "An Insight into the Interplay between Culture, Conflict and Distance in Globally Distributed Requirements Negotiations"; 36th Annual Hawaii International Conference on System Sciences (HICSS'03), 2003.

[4] Rafael Prikladnicki, Jorge Luis Nicolas Audy; "Requirements Engineering in Global Software Development: Preliminary Findings from a Case Study in a SW-CMM context"; Proc. V SIMPROS - Simpósio Internacional de Melhoria de Processo de Software, Pernambuco, 2003.

[5] Srihari Sukumaran, Ashok Sreenivas, R. Venkatesh; "A rigorous approach to requirements validation"; Fourth IEEE conference on Software Engineering and Formal Methods, 2006.

[6] Loucopoulos, P. and Karakostas, V. (eds), "Systems Requirements Engineering", McGraw Hill, 1995.

[7] Jane Coughlan, Robert D. Macredie; "Effective Communication in Requirements Elicitation: A Comparison of Methodologies", Requirement Engineering, 2002.

[8] Bikram Sengupta, Satish Chandra, Vibha Sinha; "A research agenda for distributed software development". ICSE 2006.

[9] Audris Mockus, James D. Herbsleb; "Challenges of Global Software Development"; IEEE METRICS 2001.

[10] Ban Al-Ani, Keith Edwards; "A Comparative Empirical Study of Communication in Distributed and Collocated Development Teams"; IEEE International Conference on Global Software Engineering, 2008.

[11] Yvonne Hsieh; "Culture and Shared Understanding in Distributed Requirements Engineering"; International Conference on Global Software Engineering, Oct. 2006.

[12] Eoin O Conchuir, Helena Holmstrom, Par J. Agerfalk, Brian Fitzgerald; "Exploring the Assumed Benefits of Global Software Development"; International Conference on Global Software Engineering (ICGSE'06), 2006.

[13] Gabriela Avram; "Of Deadlocks and Peopleware - Collaborative Work Practices in Global Software Development"; ICGSE, August 2007.

[14] Daniela E. Damian, Didar Zowghi; "Requirements Engineering Challenges in Multi-Site Software Development Organizations"; Requirements Engineering Journal, 8:149--160, 2003.

[15] Raghvinder Sangwan; Global Software Development Handbook; C R C Press LLC; 2007.

[16] Robert D. Battin, Ron Crocker, Joe Kreidler, K. Subramanian; "Leveraging Resources in Global Software Development," IEEE Software, vol. 18, no. 2, Mar./Apr. 2001.

[17] Henrik Munkebo Christiansen; "Meeting the Challenge of Communication in Offshore Software Development"; Software Engineering for Offshore and Outsourced Development, 2007.

[18] Daniela Damian, Luis Izquierdo, Janice Singer, Irwin Kwan; "Awareness in the Wild: Why Communication Breakdowns Occur"; ICGSE August 2007.

[19] Jo Hanisch, Brian J. Corbitt; "Requirements engineering during global software development: some impediments to the requirements engineering process - a case study"; Proceedings of the 13th European Conference on Information Systems, ECIS, 2004.

[20] Daniela Damian; "Requirements Engineering in Distributed Projects"; International Conference on Global Software Engineering, 2006.

[21] Ian Sommerville, "Software Engineering", Seventh Edition, Addison Wesley, 2004.

[22] Reinhard Budde, Heinz Ziillighoven ; "Prototyping Revisited", IEEE Computer Society , Proceedings of International Conference on Computer Systems and Software Engineering, May 1990.

[23] Andreas Homrighausen, Hans-Werner Six, Mario Winter; "Round trip prototyping for validation of requirements specification"; REFSQ'01, Interlaken, Switzerland, 2001.

[24] V. Scott Gordon, James M. Bieman; "Rapid Prototyping: Lessons Learned"; IEEE Software, Vol. 15, Jan. 1995.

[25] Stephen J. Andriole; "Fast, cheap requirements: Prototype, or else!", IEEE Computer Society, 1994.

[26] Wilhelm Hasselbring; "Programming Languages and Systems for Prototyping Concurrent Applications"; ACM Computing Surveys, Vol. 32, No. 1, March 2000.

[27] Michael Schrage; "Never go to a client meeting without a prototype"; IEEE Software, Volume 21, Issue 2, Mar-Apr 2004.

[28] Horst Lichter, Matthias Schneider-Hufschmidt, Heinz Züllighoven; "Prototyping in industrial software projects-bridging the gapbetween theory and practice"; IEEE Transactions on Software Engineering, Volume 20, Issue 11, Nov 1994.

[29] Vali Lalioti, Pericles Loucopoulos ; "Visualisation for Validation", 5th International Conference on Advanced Information Systems Engineering (CAiSE.93), Paris, June 1993.

[30] Mehmet Bülent Özcan, P. W. Parry, Ian C. Morrey, Jawed I. A. Siddiqi; "Visualisation of Executable Formal Specifications for User Validation"; Services and Visualization: Towards User-Friendly Design 1998: 142-157

[31] Alfredo Teyseyre, Marcelo Campo; "An Overview of 3D Software Visualization"; IEEE Transactions on Visualization and Computer Graphics; Volume 15, Issue 1, Jan.-Feb. 2009.

[32] Andrew Gemino; "Empirical Comparisons of Animation and Narration in Requirements Validation", Requirements Engineering Journal, Vol. 9, No. 3, 2004.

[33] Mehmet Bülent Özcan, P. W. Parry, Jawed I. A. Siddiqi; "The Application of Visualisation to Requirements Engineering"; In Proc. Conf. on Software Engineering and Its Applications, France, 1995.

[34] Pascal Fenkam, Harald Gall, Mehdi Jazayeri, "Visual Requirements Validation: Case Study in a Corba-Supported Environment," 10th Anniversary Joint IEEE International Requirements Engineering Conference (RE'02), 2002.

[35] Orlena C.Z. Gotel, Francis T. Marchese, Stephen J. Morris; "On Requirements Visualization"; Second International Workshop on Requirements Engineering Visualization, 2007.

[36] Craig K. Tyran, Joey F. George; "Improving software inspections with group process support"; Communications of the ACM, Volume 45 , Issue 9, September 2002.

[37] Marcus Ciolkowski, Oliver Laitenberger, Stefan Biffl; "Software Reviews: The State of the Practice," IEEE Software, vol. 20, no. 6, Nov/Dec, 2003.

[38] Michael Stein, John Riedl, Sören J. Harner, Vahid Mashayekhi; "A case study of distributed, asynchronous software inspection", Proc. of the 19th Int. Conf. on Software Engineering, Boston, MA, USA, 1997.

[39] Paul Murphy, James Miller; "A Process for Asynchronous Software Inspection"; 8th International Workshop on Software Technology and Engineering Practice (STEP '97), 1997.

[40] Vincenzo Ambriola, Vincenzo Gervasi; "Processing natural language requirements", 12th IEEE International Conference on Automated Software Engineering (ASE'97); 1997.

[41] Francis Chantree, Bashar Nuseibeh, Anne de Roeck, Alistair Willis; "Identifying Nocuous Ambiguities in Natural Language Requirements"; Requirements Engineering, 14th IEEE International Conference; 2006.

[42] Erik Kamsties, Barbara Paech; "Taming Ambiguity in Natural Language Requirements", in Proceedings of the Thirteenth International Conference on Software and Systems Engineering and Applications, Paris, France (2000).

[43] Erik Kamsties, Daniel M. Berry, Barbara Paech; "Detecting ambiguities in requirements documents using inspections"; In Proceedings of the First Workshop on Inspection in Software Engineering(WISE'01), 2001.

[44] Leela Damodaran; "User involvement in the system design process: a practical guide for users"; Behaviour & Information Technology, 1996.

[45] Amer Al-Rawas, Steve Easterbrook; "Communication problems in requirements engineering: a field study"; Proc. of the First Westminster Conf. on Professional Awareness in Software Engineering, Royal Society, London 1996.

[46] Seija Komi-Sirviö, Maarit Tihinen; "Lessons Learned by Participants of Distributed Software Development"; Knowledge and Process Management, vol. 12, 2005.

[47] Helena Holmstrom, Eoin O Conchuir; Par J Agerfalk; Brian Fitzgerald; "Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance"; International Conference on Global Software Engineering, 2006.

[48] Daniela Damian, "Stakeholders in Global Requirements Engineering: Lessons Learned from Practice," IEEE Software, vol. 24, no. 2, Mar/Apr, 2007.

[49] Daniela E. Herlea Damian, Armin Eberlein, Mildred L.G. Shaw, and Brian R. Gaines; "Using different communication media in requirements negotiation"; IEEE Software; Volume 17, Issue 3, May/Jun 2000.

[50] Keri K. Stephens; "The Successive Use of Information and Communication Technologies at Work"; Communication Theory, Volume 17, Number 4, November 2007.

[51] Gabriela N. Aranda, Aurora Vizcano, Alejandra Cechich, Mario Piattini; "Technology Selection to Improve Global Collaboration", International Conference on Global Software Engineering, 2006.

[52] Lai-Chung Lee, Whei-Jane Wei; "Case Study of Breakdown Analysis on Identification of Remote Team Communication Problems"; LNCS, 2006.

[53] Daniela Damian, Filippo Lanubile, Teresa Mallardo; "On the Need for Mixed Media in Distributed Requirements Negotiations"; IEEE Transactions on Software Engineering; Volume 34, Issue 1, Jan.-Feb. 2008.

[54] Grace Tai; "A communication architecture from rapid prototyping"; Human and Social Factors of Software Engineering (HSSE), St. Louis, Missouri, 2005.

[55] Nabil N. Kamel, Robert M. Davison; "Applying CSCW technology to overcome traditional barriers in group interactions"; Information & Management, Volume 34, Issue 4, 2 November 1998.

[56] Paola Di Blasio, Luca Milani; "Computer-mediated communication and persuasion: Peripheral vs. central route to opinion shift"; Computers in Human Behavior, Volume 24, Issue 3, May 2008.

[57] E. Vance Wilson, Ying Lu; "Communication goals and online persuasion: An empirical examination"; Computers in Human Behavior, Volume 24, Issue 6, September 2008.

[58] Jyoti M. Bhat, Mayank Gupta, Santhosh N. Murthy; "Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing"; IEEE Software, vol. 23, no. 5, pp. 38-44, Sep/Oct, 2006.

[59] Alexander Boden, Bernhard Nett, Volker Wulf; "Coordination Practices in Distributed Software Development of Small Enterprises"; Second IEEE International Conference on Global Software Engineering, August 2007.

[60] Erran Carmel, Ritu Agarwal; "Tactical Approaches for Alleviating Distance in Global Software Development"; IEEE Software, March/April 2001, Special Issue in Global Software Development.

[61] Robert Chatley, Sebastián Uchitel, Jeff Kramer, Jeff Magee; "Fluent-based web animation: exploring goals for requirements validation; ICSE 2005.

[62] Mehmet Bülent Özcan, P. W. Parry, Ian C. Morrey, Jawed I. A. Siddiqi; "Requirements Validation based on the Visualization of Executable Formal Specifications", 22nd International Computer Software and Application Conference, 1998.

[63] James D. Herbsleb and Rebecca E. Grinter; "Splitting the organization and integrating the code: Conway's law revisited"; Proceedings of International Conference on Software Engineering, 1999.

[64] Philip M. Johnson; "An instrumented approach to improving software quality through formal technical review"; 16th International Conference on Software Engineering, May 1994.

[65] De Andrea De Lucia, Fausto Fasano, Genoveffa Tortora, Giuseppe Scanniello; "Assessing the Effectiveness of a Distributed Method for Code Inspection: A Controlled Experiment"; Second IEEE International Conference on Global Software Engineering, August 2007.

[66] Vahid Mashayekhi , Janet M. Drake , Wei-Tek Tsai , John Riedl; "Distributed, collaborative software inspection"; IEEE Software, Volume 10, Issue 5, Sep 1993.

[67] Paul Layzell, O. Pearl Brereton, Andrew French; "Supporting collaboration in distributed software engineering teams"; Seventh Asia-Pacific Software Engineering Conference (APSEC'00), 2000.

[68] Vincenzo Gervasi, Bashar Nuseibeh; "Lightweight Validation of Natural Language Requirements: A Case Study"; Fourth International Conference on Requirements Engineering (ICRE'00), 2000.

[69] Kevin Ryan; "The role of natural language in requirements engineering"; Proceedings of IEEE International Symposium on Requirements Engineering, 1993.

[70] Stephen Boyd, Didar Zowghi, Alia Farroukh; "Measuring the Expressiveness of a Constrained Natural Language: An Empirical Study", Proceedings of the IEEE International Requirements Engineering Conference (RE05), Paris, France, August 2005.

[71] Thomas Thelin, Per Runeson, Claes Wohlin; "An Experimental Comparison of Usage-Based and Checklist-Based Reading"; IEEE Transactions on Software Engineering, August 2003.

[72] Leonid Kof; "Natural Language Processing for Requirements Engineering: Applicability to Large Requirements Documents"; Proceedings of the Workshops In Automated Software Engineering, 2004.

[73] Jorge H. Doorn, Marcela Ridao; "Completeness Concerns in Requirements Engineering"; Encyclopedia of Information Science and Technology 2nd Edition, September 2008.

[74] Fabrizio Fabbrini, Mario Fusani, Stefania Gnesi, Giuseppe Lami; "The Linguistic Approach to the Natural Language Requirements Quality: Benefit of the use of an Automatic Tool"; 26th Annual NASA Goddard Software Engineering Workshop, 2001.

[75] Rebecca E. Grinter, James D. Herbsleb, Dewayne E. Perry; "The geography of coordination: dealing with distance in R&D work"; GROUP 1999.

[76] Lucas D. Panjer, Daniela Damian, Margaret-Anne Storey; "Cooperation and coordination concerns in a distributed software development project"; Proceedings of the

International workshop on Cooperative and human aspects of software engineering, Leipzig, Germany, 2008.

[77] Marilyn Tremaine, Allen E. Milewski, Richard Egan, Suling Zhang; "A Tale of Two Teams: Success and Failure in Virtual Team Meetings"; Proceedings of the Human-Computer Interaction International Conference, Beijing, China, July 22-27, 2007.

[78] Gary C. David, Donald Chand, Sue Newell, João Resende-Santos; "Integrated Collaboration across distributed sites: The perils of process and the promise of practice" Journal of Information Technology,Volume 23, Number 1, March 2008.

[79] Teresa Mallardo , Filippo Lanubile , Daniela Damian; "The role of asynchronous discussions in increasing the effectiveness of remote synchronous requirements negotiations"; 28th International Conference on Software Engineering, May 2006.

[80] Helen Sharp, Anthony Finkelstein, Galal Galal; "Stakeholder identification in the requirements engineering process"; In Proceedings of 10th International Workshop on Database & Expert Systems Applications (DEXA). IEEE Computer Society Press, 1999.

[81] Kurt Bittner, Ian Spence; "Use Case Modeling"; Addison-Wesley Professional, August 20, 2002.

[82] Marcelo Cataldo, Matthew Bass, James D. Herbsleb, Len Bass; "On Coordination Mechanisms in Global Software Development"; Second IEEE International Conference on Global Software Engineering, August 2007.

[83] Per Runeson, Martin Höst; "Guidelines for conducting and reporting case study research in software engineering"; Empirical Software Engineering, Volume 14, Issue 2, April 2009.