

Spring 2013

Collaborative Requirements Engineering Notation for Planning Globally Distributed Projects

Paula Laurent

DePaul University, plaurent@cdm.depaul.edu

Follow this and additional works at: https://via.library.depaul.edu/cdm_etd



Part of the [Computer Sciences Commons](#)

Recommended Citation

Laurent, Paula, "Collaborative Requirements Engineering Notation for Planning Globally Distributed Projects" (2013). *College of Computing and Digital Media Dissertations*. 6.
https://via.library.depaul.edu/cdm_etd/6

This Dissertation is brought to you for free and open access by the College of Computing and Digital Media at Via Sapiientiae. It has been accepted for inclusion in College of Computing and Digital Media Dissertations by an authorized administrator of Via Sapiientiae. For more information, please contact digitalservices@depaul.edu.

DEPAUL UNIVERSITY

Collaborative Requirements Engineering Notation for Planning Globally Distributed Projects

by

Paula Laurent

Dissertation Thesis submitted in partial fulfillment for
the degree of Doctor of Philosophy

in the

School of Computing

College of Computing and Digital Media

Advisors:

Jane Cleland-Huang, PhD

Adam Steele, PhD

April 2013

Abstract

Collaborative Requirements Engineering Notation for Planning Globally Distributed Projects by Paula Laurent

Requirements engineering represents a critical phase of the software development lifecycle in which requirements describing the functional and non-functional behaviors of a system are elicited, modeled, analyzed, negotiated, agreed, and specified. In traditional software systems these tasks are typically performed in face-to-face meetings between requirements engineers and the project level stakeholders. However, in today's global software development environment, it is becoming increasingly commonplace for stakeholders to be dispersed across multiple geographical locations and time zones. Under these circumstances, face-to-face meetings become expensive, and often impossible to facilitate, and as a result the success of the requirements process relies, at least partially, on tools and processes that support distributed communication and collaboration.

To investigate the challenges and effective practices for performing requirements related activities in distributed environments, we conducted a series of in-depth interviews with project managers and business analysts who have worked on projects with non co-located stakeholders. Unfortunately, many project managers fail to plan and deploy the necessary infrastructures to support quality communication, and in practice requirements are often elicited and managed via email exchanges. To address these problems we introduced a visual modeling notation to help project managers proactively plan the collaboration infrastructures needed to support requirements-related activities in globally distributed projects. An underlying meta-model defines the elements of the modeling language, including site locations, stakeholder roles, communication flows, critical documents, and supporting tools and repositories.

During a follow-up study we observed several project managers using the visual modeling notation to plan their globally distributed projects. Results from the study showed that the modeling activity and the resulting diagrams helped the project managers to better understand the communication needs for the project, to identify potential communication and collaboration problems, and to proactively address the infrastructure and communication needs for the project.

The interview findings were then further analyzed to identify practices that either led to success or created significant challenges for the projects. The result of this analysis is a set of patterns and anti-patterns for globally distributed requirements engineering. The contributions of this research are meant to improve the practices and utility of distributed Requirements Engineering.

Acknowledgements

I want to express a most heartfelt thank you to all who supported me during this truly amazing undertaking.

Dr. Jane Cleland-Huang and Dr. Adam Steele thank you for being my advisors and for your invaluable contribution to this work. Your guidance contributed to my growth as a researcher and academic. For this and much more I am most grateful.

Jane you are a wonderful role model on being passionate about what you do both inside and outside of the university setting. Your enthusiasm during our project discussions was often contagious and I would return to work on my research tasks with renewed confidence.

To my dissertation committee Dr. Craig Miller, Dr. Xioaping Jia and Dr. Patrick Mader, thank you for your willingness to work with me, the feedback you shared and being available through the many false starts. Your time and consideration is greatly appreciated.

To my family; my mother who taught her children that they can accomplish great things through hard work and with God's presence in their lives; my father for reinstating my allowance when I took time off from work to focus on my research; and to my sister and brother for letting me believe that I was the smart one in the family, know that I could not have achieved this milestone without you. You are truly the wind beneath my wings. Finally to my husband John, by my side these many years, thank you for your continued love and support.

To every member of my support network of family, friends, classmates and colleagues, I say again thank you; your encouragement always inspired me to move forward.

Contents

Abstract	i
Acknowledgements	iii
List of Figures	viii
List of Tables	ix
Abbreviations	x
1 Introduction.....	1
1.1 Problem Statement and Motivation.....	2
1.2 Goal.....	3
1.3 Contribution	5
1.4 Published Work.....	6
1.5 Organization.....	7
2 Related Research.....	9
2.1 Requirements Engineering.....	9
2.1.1 Requirements Development.....	10
2.1.1.1 Elicitation.....	10
2.1.1.2 Analysis and Prioritization.....	11
2.1.1.3 Specification	13
2.1.1.4 Validation.....	13

2.1.2 Importance of Requirements Engineering	14
2.2 Distributed Requirements Engineering	16
2.3 Social Networks and Modeling.....	18
3 Methodology	22
3.1 Grounded Theory	23
3.2 Surveys and Questionnaires	26
3.3 Research Instrument.....	27
3.3.1 Open Source Software Projects Surveys.....	28
3.3.2 Information Technology Industry Interviews	29
3.3.3 Requirements Engineering Modeling Sessions.....	30
3.3.4 Visual Notation Icon Selections.....	30
4 Open Source Software Projects Data Analysis.....	32
4.1 Open Source Software and Surveys.....	32
4.2 Feature Requests	35
4.3 Research Findings	37
4.3.1 Creating Collaborations	37
4.3.2 Prioritizing Features.....	39
4.3.3 Engaging and Communicating.....	42
4.3.4 Managing Feature Requests	42
4.3.5 Identifying User Roles	43
4.4 Proposed Solutions.....	44
4.5 Conclusion	46
5 Distributed Requirements Engineering Interviews Data Analysis.....	48
5.1 Distributed Requirements Engineering Interviews	48
5.2 Meta-model and Visual Notation	52
5.2.1 Meta-model	52
5.2.1.1 Roles	52
5.2.1.2 Sites.....	53
5.2.1.3 Artifacts	53
5.2.1.4 Relationship between roles	54
5.2.2 Visual Notation	55
5.2.2.1 Basic Elements.....	56
5.2.2.2 Complexity Management.....	58

5.2.2.3 Examples.....	58
5.3 Research Findings.....	61
5.3.1 Pattern: Requirements Analyst Title.....	63
5.3.2 Pattern: Telephone – Requirements Analyst’s Preferred Communication Tool.....	69
5.3.3 Pattern: Location Spokesperson Role	73
5.3.4 Pattern: Requirements Engineering Tools and Technologies	77
5.3.5 Pattern: Face to Face Communication	90
5.3.6 Pattern: Requirements Specifications Shared only via Email.....	94
5.3.7 Pattern: Distributed Stakeholder Communications.....	98
5.3.8 Pattern: Multipurpose Requirements Specifications.....	101
5.3.9 Pattern: Travel Between Sites Unnecessary for Small Groups.....	104
5.4 Proposed Solutions and Conclusion.....	106
6 Requirements Engineering Modeling Sessions Data Analysis.....	108
6.1 Requirements Engineering Modeling Sessions.....	108
6.2 Research Findings	109
6.3 Proposed Solutions.....	110
6.4 Conclusion	114
7 Research Contributions.....	116
7.1 CGREN for creating RCGN Models.....	116
7.2 Patterns.....	117
7.3 GRETA	117
8 Future Research Opportunities	118
 A Data Collection Tools.....	 120
B RE Research Interviews Database Design	128
Bibliography	141

List of Figures

2.1 Requirements Development Process.....	14
4.1 Vendor Based Open Source Process for Entering and Managing Feature Requests	36
4.2 Methods Preferred by Users for Entering Feature Requests	39
4.3 User Satisfaction with the Requirements Management Process	40
4.4 Methods for Prioritizing Feature Requests	41
5.1 Meta-model Depicting Taxonomy for Distributed Requirements Engineering	55
5.2 CGREN Icons – Sites, Stakeholders and Relationships.....	57
5.3 CGREN Icons – Communication Media and Artifacts.....	58
5.4 General Structures of Requirements Gathering Collaborative Networks	59
5.5 Modeling Co-located Communication.....	61
5.6 Modeling Distributed Communication and Collaboration with LSP at Remote Site	61
5.7 Modeling Access to a Shared Artifact across Multiple Sites	61
5.8 Communication Flow between Project 3’s RA and Stakeholders	61
5.9 Communication Flow of Project 3 Modeled as a RGCN.....	88
5.10 Documentation Flow of Project 3	88
6.1 Requirements Engineering Sessions CGREN Glossary.....	112
6.2 Updated Meta-model.....	113
6.3 New Icons for Multiple Roles and Requirements Elicitation	113
6.4 Modeling a Joint Application Design session.....	113

List of Tables

4.1 Number of posts and responses of surveyed forums.....	34
4.2 Features observed in Opens Source Forums	35
5.1 Research study statistics.....	51
5.2 Research Analysts' Organizational Titles	64
5.3 Possible Telephone Pathways for Requirements Analyst	71
5.4 Projects with Location Spokes Person role	74
5.5 Tools and Technologies used by Requirements Analyst	78
5.6 Requirements development tasks and categories of tools	84
5.7 Types of collaboration tools.....	85
5.8 Face-to-Face Communications	92
5.9 No Face-to-Face Communications.....	96
5.10 Distributed Stakeholder Communications	99
5.11 Requirements Specification Utilization	103
5.12 No Face-to-Face Communications (small groups)	105
B.1 RE Interviews Project Metadata	128
B.2 RE Interviews Site-specific Metadata	128
B.3 RE Interviews Description of RE activities and RA quotes	129

Abbreviations

BA	B usiness A nalyst
CGREN	C ollaborative G lobal R equirements E ngineering N otation
GRETA	G lobal R equirements E ngineering T ool/ A pplication
IT	I nformation T echnology
JAD	J oint A pplication D esign
LSP	L ocation S pokes P erson
OSS	O pen S ource S oftware
OSSP	O pen S ource S oftware P roject
OTS	O ff the shelf
PM	P roject M anager
RA	R equirements A nalyst
RE	R equirements E ngineering
RCSN	R equirements- C entered S ocial N etwork
RGCN	R equirements G athering C ollaborative N etwork
SME	S ubject M atter E xpert
UML	U nified M odeling L anguage

*Chère Maman,
Merci pour tout.*

Chapter 1

Introduction

Requirements engineering (RE), a critical part of the software development lifecycle, is becoming increasingly difficult to manage as the scope of software projects continues to expand and project stakeholders are dispersed organizationally, geographically and temporally. This research focuses on the challenges and successes of conducting requirements elicitation, analysis and management with stakeholders at distributed heterogeneous locations.

Helping the stakeholders to discover and communicate their needs is a complex task [1, 2]. Oftentimes people have difficulty articulating what they want [3], therefore the requirements analyst (RA) must work closely with stakeholders to perform requirements elicitation activities to discover what they really need [4-6]. While working in a non-distributed software development project, stakeholders and RAs are co-located and therefore have the chance to engage in the requirements engineering process face-to-face on a regular basis. However, today's environment of globally deployed software projects requires stakeholders from dispersed locations to be included in the task of discovering and specifying requirements. With limited opportunity to spend time with stakeholders at distributed locations, requirements engineers, project managers (PMs) and business analysts (BAs) are forced to rely on technology to bridge the distance gap as they attempt to gather, analyze and manage all of the requirements information and artifacts. Face-to-face meetings become expensive, and often impossible, in such circumstances; instead the success of the requirements development phase of a distributed project relies to a large extent on the efficiency and performance of the chosen methods of communication and their supporting tools.

This research specifically contributes a new framework for analyzing distributed RE processes. The components of this new framework are:

- i) A modeling notation, Collaborative Global Requirements Engineering Notation, (CGREN) that facilitates the analysis and planning of distributed requirements gathering activities, including the flow of communication, the tools used for collaboration and stakeholder roles. The notation also provides practitioners with a technique to draw comparisons across projects.
- ii) Application of the new modeling notation to create a canonical set of requirements-gathering collaborative network (RGCN) models based on current industry practices.
- iii) Distributed requirements gathering patterns, based on the features of the RGCNs, which describe strategies for conducting successful RE processes and the pitfalls to avoid.

An additional contribution, the prototype of a web-based tool named GRETA that will assist practitioners in planning and executing distributed requirements activities, is also being developed.

1.1 Problem Statement and Motivation

The success of any software development project can be measured by how well it meets the stakeholders' needs and their system environment. Software requirements are the expression of these needs and RE is the process for determining these needs [7, 8]. RE entails translating the informal and incomplete needs, wants and desires of stakeholders into an exact description of what a software system must do.

The requirements phase is typically the first phase of any software engineering project. It is during this phase that the development team and stakeholders work together to define what the resultant new application or system enhancement will actually do. The primary output of this first phase is the requirements specification document, which is the blueprint for designing and constructing or updating a software system [4].

Since every software project has stakeholders who depend on it to help them work efficiently and to focus on their project goals, the time devoted to understanding their needs contributes greatly to project success [7]. Experts [6, 9] agree that the greatest investment in software development time is the time spent gathering, analyzing, documenting and managing the project requirements, because this first phase of a project is the easiest place to introduce defects into the to-be-developed software system. These defects can potentially account for up to half of the total system defects overall. Furthermore, studies have shown that finding and fixing these defects later in the project can cost as much as one-hundred times the cost after implementation than if they had been detected and corrected in the beginning, during the requirements phase [9-11].

The outcome and results of each development phase, i.e. requirements elicitation, specification, design and implementation, remain the same for distributed RE as in a co-located process, however the way in which stakeholders cooperate to achieve these goals differs considerably [12]. Numerous practitioners and researchers have conducted studies in requirements engineering for co-located industrial environments [1, 8], resulting in a variety of techniques that prescribe effective practices for a vast array of activities during the requirements phase of a project.

Prior research in the distributed requirements engineering domain has primarily concentrated on the communication and collaboration efforts of non-co-located members of the development team [12-16]. In contrast this research will focus on the distributed requirements relationships and activities that occur between the project leader responsible for gathering and analyzing the requirements and the project stakeholders who are the source of the requirements, to identify best practices and challenges.

During a typical non-distributed RE process, key project stakeholders are selected to participate in a series of face-to-face elicitation, analysis, and specification activities resulting in the requirements specification of a development project [2, 4]. However the stakeholders in globally distributed projects are often geographically separated across multiple sites, and typical RE activities such as face-to-face discussions and brainstorming sessions, which are relatively straightforward to facilitate when all participants are co-located, become significantly more challenging. Additional challenges discussed in prior research include: unsuccessful group collaboration due to process, language and cultural differences; ensuring stakeholder participation and communication; acquiring necessary requirements-related domain knowledge; managing stakeholder and requirements conflicts, and organizing and managing requirements documents and artifacts [12, 14, 15, 17]. Though in some cases it could be possible to assemble the stakeholders in one location for a series of centralized in-person meetings, in many projects this is not possible in terms of cost and effort.

1.2 Goal

The goals of this research was to explore the current practices and challenges of gathering requirements with stakeholders in a distributed environment; and to analyze and evaluate the communication paths and processes which have often been established in an ad-hoc manner; in order to identify patterns which either work or are problematic in industrial settings. The in-depth data analysis resulted in recommended improvements for the practices and utility of distributed RE. The outcome of the research is a framework

that project leaders can use to plan the distributed requirements development phase of a project in a systematic and goal-oriented way. This framework is based on a newly created visual notation that can be used to describe and categorize distributed RE activities and projects.

Established effective practices were modeled as patterns that can enhance traditional RE methodologies for large-scale distributed development projects. In essence these patterns will serve as guidelines that can help practitioners to plan their global requirements engineering projects and recognize some of the ineffective processes they may have in place. Descriptions of these patterns include a discussion of their strengths, weaknesses and applicability, in order to facilitate a systematic method for applying them in industrial projects.

To accomplish this goal, firstly, industry professionals who are responsible for gathering requirements from stakeholders at distributed locations were interviewed. These interviews were conducted using a specially created questionnaire to:

- i) Categorize types of communication, including synchronous methods such as person-to-person phone calls and teleconferencing, and asynchronous methods such as forums and emails.
- ii) Identify the various documents such as requirements specification, design drafts, scope of work, contract, etc., being shared between the RA and the distributed stakeholders.
- iii) Design RGCNs that represent the current industry processes for distributed requirements development activities such as gathering, documenting and managing requirements.

Initial analysis of the data gathered during the first round of interviewing yielded details about the distributed requirements gathering and managing processes, including descriptions of stakeholder roles, communication paths, tools usage; and the first set of real-world RGCNs. The questionnaire was then enhanced and two successive rounds of interviewing helped to validate the new taxonomy and meta-model designs, and recognize existing activity and organization patterns across projects.

We also conducted a participatory study in which industry RAs were asked to interactively use the refined visual notation, the Collaborative Global Requirements Engineering Notation, (CGREN) to plan and/or re-plan the roles, communication paths, and tooling needed to support the globally distributed requirements processes in their own projects.

1.3 Contribution

This research contributes to a more complete picture of distributed RE industrial practices. Through the analysis of requirement activities of global software engineering projects a clearer and more complete understanding of differing aspects of the whole process can be understood. Connecting the different aspects of the distributed requirements process allows easier identification of the different ways stakeholders participate, recognition of the challenges that stakeholders face, categorization of the tools used, and discovery of the processes that practitioners use to solve these interrelated issues.

As a result of studying real world projects a new meta-model and visual notation, CGREN, have been developed, that enables project managers to create RGCN models and analyze their distributed RE processes to identify and address potential problems early on in the software engineering process. CGREN also provides a common language to researchers for modeling distributed RE projects, thus facilitating analysis and comparisons across projects [18].

CGREN was needed because the existing visual notations in the RE and project management domains are unable to depict all of the necessary concepts for characterizing distributed RE projects. For example, UML Use Case diagrams [19, 20] primarily model the actual requirements and not the stakeholders and environment. Although the stakeholder onion model [21] identifies stakeholder roles, it does not describe how they interact with each other; and organizational charts mostly describe reporting structure, not person-to-person communication and collaboration activities [22, 23]. While Damian et al's [15] requirements-centered social networks (RCSNs) do provide a few of the necessary components for this research, i.e. sites, stakeholders, and communication paths; additional concepts such as artifact sharing and channels of communication are missing. For these reasons there exists no single existing solution to allow the practitioner to understand the holistic view of the distributed RE problem. Therefore a new approach, RGCN is presented as a solution to these needs. These RGCN models are used to describe the stakeholder roles, characteristics of the team structures, communication paths, collaboration artifacts, and tools in use, of distributed RE projects in industry.

Another contribution of this research is a set of distributed RE organizational and activity patterns, based on the characteristics of the RGCN models, that capture some of the best practices for addressing commonly occurring problems and challenges; and are designed to provide guidance to practitioners conducting geographically dispersed RE activities.

Furthermore, the new modeling language and associated functionality will be made available to practitioners and researchers, via GRETA, a web-based tool that is being developed. An early prototype of GRETA can be found at <http://golevka.cstcis.cti.depaul.edu/GlobalRETool/>.

1.4 Published Work

Findings for this research have been presented at several relevant international conferences, such as Requirements Engineering, Software Engineering and Global Software Engineering. Listed below is a complete list of the abstracts and presentations.

The research projects discussed in this thesis have been published in the following refereed conferences and workshops:

- **P. Laurent**, P. Mader, J. Cleland Huang, and A. Steele, "A Taxonomy and Visual Notation for Modeling Globally Distributed Requirements Engineering Projects," presented at 5th IEEE International Conference on Global Software Engineering, Princeton, USA, 2010 [24].
- **P. Laurent**, "Globally Distributed Requirements Engineering", presented at 5th IEEE International Conference on Global Software Engineering, Doctoral Symposium Princeton, USA, 2010 [25]
- **P. Laurent** and J. Cleland Huang, "Requirements-Gathering Collaborative Networks in Distributed Software Projects," presented at 17th IEEE International Conference on Requirements Engineering, Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills workshop, Atlanta, GA, 2009 [26].
- **P. Laurent** and J. Cleland Huang, "Lessons Learned from Open Source Projects for Facilitating online Requirements Processes," presented at 15th International Working Conference on Requirements Engineering: Foundation for Software Quality, Amsterdam, Holland, 2009 [27].
- **P. Laurent** and J. Cleland-Huang, "Requirements-Gathering Collaborative Networks," presented at 31st International Conference on Software Engineering, Socio-Technical Congruence workshop, Vancouver, Canada, 2009 [28].

The Open Source Software research project also resulted in a book chapter, **P. Laurent** and J. Cleland Huang, "Managing Requirements in Vendor-Based Open Source Software Development Projects," published in, *Open Source Software Developments*. Hauppauge: Nova Science Publishers, Inc., 2010 [29].

Other research presentation and publications include:

- **P. Laurent**, J. Cleland-Huang, and C. Duan, "Towards Automated Requirements Triage," presented at 15th IEEE International Requirements Engineering Conference (RE 2007), Delhi, India, 2007 [30], winner of the Best Paper award.
- **P. Laurent**, "Task Modeling and Analysis of Traceability Users in the Wild," presented at 15th IEEE International Requirements Engineering Conference (RE 2007) Doctoral Consortium, Delhi, India, 2007 [31].
- **P. Laurent**, "Traceability Task Modeling," presented at DePaul CTI Research Symposium (CTIRS 2007), Chicago, IL, 2007 [32].
- **P. Laurent**, "Trace Support for Requirements Prioritization," presented at ACM International Symposium on Grand Challenges of Traceability (GCT '07), 2007 [33].
- C. Duan, **P. Laurent**, J. Cleland Huang, and C. Kwiatkowski, "Towards automated requirements prioritization and triage," *Requirements Engineering*, vol. 14, pp. 73-89, 2009 [34].
- **P. Laurent** and A. Steele, "Using GOMS KLM to Support Cross Platform Prototyping," presented at Midwest Business Administration Association (MBAA 2005), Chicago, IL, 2005 [35].
- **P. Laurent** and A. Steele, "From Prototype to Application," presented at 2004 Midwest Software Engineering Conference, Chicago, IL, 2004 [36].
- **P. Laurent**, "Cross Platform Prototyping," presented at DePaul CTI Research Symposium (CTIRS 2001), Chicago, IL, 2001 [37].

1.5 Organization

This thesis is organized as follows: Literature reviews of requirements development, distributed requirements engineering and its importance and modeling techniques are presented in Chapter 2. Chapter 3 discusses grounded theory and the research instruments, namely surveys and questionnaires that were used in these studies.

In Chapter 4 we describe the Open Source Software Projects research study and results. These results helped to further motivate our research into the requirements development techniques and tools used in industry settings involving globally distributed project stakeholders.

Chapter 5 discusses the distributed RE study and results; describes the initial CGREN design and RGCN models. Next each of the nine patterns that were discovered during this research are presented; along with references to existing research and recommendations for practitioner use.

In Chapter 6 we describe a brief follow-up observational study we conducted to obtain feedback regarding the usefulness of CGREN. A refined meta-model and CGREN notation are the results of participants' feedback. Another outcome of this study is a step-by-step process for developing project specific RGCNs.

Finally in Chapters 7 and 8 we summarize the research contributions and applicability to RE in industry, that are detailed in previous chapters; and describe opportunities for future research, respectively.

Chapter 2

Related Research

This chapter details previous research that is related to the topic of distributed requirements development. In section 2.1 RE is defined and its importance as a research topic discussed. In the remaining sections the significance of developing software in a globally dispersed environment and by extension global RE, and background information about modeling techniques for capturing project activities and stakeholder interactions, is described.

2.1 Requirements Engineering

Requirements are the foundation of the software development process and project management activities. The goal of any development process is the construction of an appropriate solution to a specific problem. To achieve this goal in the software engineering domain, the stakeholders' needs have to be discovered, understood, recorded and managed [4]. RE entails translating informal and incomplete stakeholder needs into an exact description of what a software system must do. Easterbrook explains that the term *requirements engineering* refers to the “elicitation and formulation of requirements to produce a specification” [3].

The typical phases of any software engineering project are gathering the requirements; analysis/specification, design, implementation which includes coding, testing and implementing; maintenance, and eventually software retirement [1, 2, 6, 38]. This research centers on the requirements phase, also called the definition phase, which is the first formal mandatory phase of software development. Industry professionals acknowledge that this phase is indeed “both the most important and least costly project phase” [9]. This phase results in a detailed description of the software system to be developed. It is according to this requirements specification that the software product is tested near the end of the project to demonstrate that the required product has indeed been produced. The requirements specification answers the question *what* while attempting to avoid the question *how* [39].

The two major categories of RE activities are Requirements Development and Requirements Management [2, 4, 6, 40]. Requirements management includes all of the activities that continue throughout the project lifecycle – from when the first stakeholder need is elicited until the system is retired. Included activities

are software configuration management, requirements traceability, impact analyses, change and version control [2]. Requirements development encompasses all activities pertaining to the eliciting, gathering, evaluating and documenting of requirements for a software project [2]. Our research focuses on requirements development in industry - the current practices and challenges of eliciting and gathering requirements from stakeholders who are located at multiple distributed locations.

Though the requirements phase is acknowledged as the first phase of most software engineering life-cycles, this phase of a project should be considered an iterative process, if the project is to be successful. Stakeholder wants and needs will need to be gathered, clarified, refined and confirmed by the stakeholders themselves throughout the system development.

2.1.1 Requirements Development

Requirements Development is comprised of four high-level activities: elicitation, analysis, specification and validation. Refer to Figure 2.1 for an illustration of the entire requirements development process. The first three activities, elicitation, analysis and specification are the focal point of our research.

2.1.1.1 Elicitation

At the beginning of a project the development team member responsible for collecting the requirements, which we will henceforth refer to as the requirements analyst (RA), meets with the various stakeholders to discuss what they need and want from the system modifications or new system. The first activity, elicitation, is considered the project discovery process [2, 4, 6]. This is when the RA needs to identify the necessary stakeholders, discover their needs, gather information about the problem that the proposed new application will address; identify and negotiate conflicts and establish clear project scope and boundaries by proactively working with the stakeholders [2, 4]. During requirements elicitation knowledge is gathered about the stakeholders' needs by helping the stakeholders to understand and articulate their problems and describe their own vision of the to-be-developed system [2]. Information is iteratively collected, clarified and reformulated [4]. Though this is the initial step, requirements elicitation continues throughout the project lifecycle.

Furthermore it is during requirements elicitation that the development team representatives are learning about the problem domain and the stakeholders are learning about the relationship between their individual needs and those of the other stakeholders as well as the feasibility of the project as a whole [4]. The developers and stakeholders need to understand the problem and its domain, identify the relevant business objectives as relates to the project and the developers need to understand the needs and constraints of the stakeholders [2]. It is imperative at this point that the entire project team, that is both

the stakeholders and developers, concentrate on what the system should do as opposed to the how the system should do it.

Many techniques exist for eliciting and gathering user needs including collaborative sessions, i.e. workshops, brainstorming, and joint application design (JAD) sessions; one-on-one or group interviewing, which is one of the simplest and most effective techniques; ethnography when members of the development team observe how users interact with an existing application; questionnaires, user stories and scenarios, whereby the stakeholder identifies their main tasks and detail the sequence of events and conditions that comprise each task; modeling which is used by developers to communicate back to the stakeholders their understanding of the stakeholders' needs; prototyping, role-playing and using documentation from existing application such as user manuals, system output and problem reports [1, 2, 4, 41]. Conditions such as the type, scope and size of a project and stakeholder availability help to determine the appropriate techniques to use [41].

2.1.1.2 Analysis and Prioritization

Analysis, the second requirements development activity, pertains to understanding the problem and synthesizing a set of requirements that specify the best solution [4]. During analysis the requirements need to be clarified to ensure that all stakeholders understand them. The project team scrutinizes the gathered requirements for omissions and errors [6]. Requirements elicitation and analysis often happen simultaneously as this makes for a more efficient process. During analysis the development team acquires a deeper understanding of the system and its interaction, and identify stakeholder needs with global impact - farther reaching impact than just current project; in order to define the high-level architectural design, allocate requirements to architectural components, identify any additional conflicts as a result of the architectural considerations; and negotiate agreements between stakeholders [2].

The negotiation process is typically referred to as prioritization. Prioritizing the stakeholder needs usually happens in this phase [2, 4, 6]. For most projects there are many requests but finite resources including personnel, time and funding; and time to market deadlines may also exist. Due to these types of constraints stakeholders need to carefully prioritize and select, if need be, a subset of requirements for development. Oftentimes stakeholders have very different objectives, meaning that they will need to work collaboratively to identify conflicting requirements, negotiate solutions, and ultimately prioritize and triage requirements. Triage [42], a concept borrowed from the medical field, is the practice of quickly and systematically categorizing the victims of a disaster into one of three groups: those who can survive and go on to live normal lives only if they receive treatment; those who will recover even if they do not

receive treatment; and those with no hope of survival even if they receive treatment. In the requirements engineering domain, triage can be defined as the process of determining which requirements a product or release should satisfy given the availability of personnel, time and other resources [30, 42]. In large projects it can be hard for a development team to manually, fully organize and review all of the stakeholders' requests within typically demanding time-to-market deadlines. Furthermore good triage decisions require broad knowledge of technical, marketing, financial and sales issues; and therefore cannot be successfully performed by a single group of stakeholders [43].

The prioritization activities help the RA with resolving conflicts and scheduling product releases [6]. Agreeing on requirements' priorities helps the trade-off process since it is seldom possible to implement all high-priority requirements because the cost of implementation has to be balanced against priority [4]. Setting the priorities early in the project and reassessing them as necessary in response to changes in customer needs, market and business conditions, enables the development team to spend their time on high value activities.

There are many different prioritization techniques used in practice. Refer to Mead's [44] article for a discussion of several candidate prioritization methodologies that have been found useful in traditional requirements engineering. The summarized methodologies include Binary Search Tree (BST) [44], Numerical Assignment Technique [45, 46], Planning Game [47], 100-Point Method [48], Theory-W also known as "Win-Win" [49, 50], Requirements Triage [42], Wiegers' Method [51], Requirements Prioritization Framework [52, 53], and Analytic Hierarchy Process (AHP) [54-56]. Oftentimes stakeholders simply categorize their requirements as mandatory, desirable, or inessential [45]; quantitatively rank them [30]; or else employ the simplest of techniques, and rank their needs as high, medium or low. Project requirements can also be prioritized according to the availability of resources, i.e. time, personnel, costs; business objectives, functionality, need-by-dates, etc. The RA helps the stakeholders with the prioritization process in accordance with their organization's project management methodology.

Conflicts can also occur while prioritizing requirements. Two users can each present what they consider to be a number one priority. Potential conflicts must first be identified and brought to the attention of the project team. Then discussions of possible solutions that do not adversely impact the project schedule, cost and other high-priority deliverables must be had. Finally all involved must participate in selecting the option with the least amount of losses for every group [2, 38]. Cost and scheduling constraints usually cause the biggest conflicts. Scheduling functionally to be phased in over time is typically the solution

used in these instances. By the end of the analysis activities the stakeholder needs have been transformed into baseline requirements [4].

2.1.1.3 Specification

The specification activity entails documenting the requirements in an appropriately formal manner, i.e. text, flow diagrams, etc; that is accessible and reviewable by all project team members [1, 6, 38]. At the very least, the documentation should describe the problem or opportunity that exists and the functionality needed to address this problem [2, 6]. These documents can be authored solely by the RA or collaboratively by the RA and stakeholders [6]. The collection of documents, i.e. Systems Definition, Systems Requirements Specification capture the system requirements; and can be systematically reviewed, evaluated and approved [2, 4].

This activity concentrates on the production of the requirements specification. Easterbrook explains that the requirements specification serves as the contract between the stakeholders and the development team [3]. The specification is also the main channel of communication between the entire project team; as well as amongst the developers themselves.

Documenting the requirements should be started as the requirements are being gathered. These documents are then stored in an appropriate location, for example on a shared drive or in a requirements repository. Access to the storage location is governed by management policies; and on an ongoing basis the documents are updated and versioned as changes to the requirements occur.

2.1.1.4 Validation

During the validation activity the project team works to ensure that the requirements are correct, possess appropriate quality attributes, and will satisfy stakeholder needs [6]. Validation activities including formal and informal reviews [2] continue throughout all of the requirements development and management processes.. The validation activities are also described as the verification and validation (V&V) process; activities focused on ensuring the application will function as documented in the requirements documentation [2]. By this point in the requirements development process the stakeholder requests have been transformed into project requirements as a result of the collaborative efforts of the stakeholders and development team members.

Any changes introduced from this point forward in the software development lifecycle will be handled using Change and Version control procedures. After the requirements have been gathered and

The Standish Group reported that “clear Statement of Requirements” was among the top 3 reasons that project succeeded; “incomplete Requirements and Specifications” was among the top 3 reasons that projects were challenged; and “Lack of Requirements” was the number 1 reason that projects failed [57]. To the extent that these figures represent the overall industry, an estimated one third of a software development projects experienced serious problems related to requirements gathering, documenting and management [11].

Additional research data indicates that errors made during the requirements phase of a project account for 40-60 percent of software project defects [6]. Research studies performed independent of each other at companies including IBM, TRW, GTE and HP have measured and assigned costs to finding and fixing errors that occur at various phases of the software development lifecycle. If a unit cost of one equals the required effort to identify and fix an error that is discovered during the coding phase; then the cost to identify and fix an error detected during the requirements phase is five to ten times less. Whereas the cost to identify and fix an error during the maintenance phase is twenty times more [11]. Results of Boehm and Basil’s research indicate that finding a software problem and fixing it after implementation can be 100 times more expensive than finding and fixing it during the early requirements management and design phases [10].

Since requirement-related errors are among the most common and costly to fix, the success of software development projects is dependent on effective requirements management. In an Information Week analytics report about aligning application development with business goals, Roger Smith notes, “It’s not rocket science: obviously it will be less expensive to correct an error in a one-sentence requirements than after design diagrams, code modules, pages of user documentation, and dozens of test cases have been written to it [9].”

To manage software requirements, the support and cooperation of the entire project team is needed, because requirements development is a process, more than any other specific software development activity, that touches every project team member – the core team of developers as well as the extended team of customers and users [11], who more than likely are now situated at different locations around the globe.

2.2 Distributed Requirements Engineering

Cheng and Atlee describe a paradigm shift as “a revolutionary solution that introduces radically new ideas or technologies to tackle a new or existing problem [7].” Paradigm shifts are not everyday occurrences, but when they do happen, they impact a field tremendously. This type of shift usually begins with a novel approach for studying a particular problem. For example the World Wide Web has significantly altered how society communicates and how services are delivered to customers. A current paradigm shift is the shift toward global software development, and by extension, global requirements engineering and development [7]. Globalization is recognized as one of the major research challenges in the requirements engineering field [7] because RE is a people-intensive activity in which key project stakeholders, who are no longer co-located, are included in a series of elicitation, analysis, and specification activities [2, 4].

Prior studies in global development have focused primarily on the overall development effort of globally dispersed teams; for example Desouza, et al [13] researched how knowledge was managed and shared across global development teams. Their resultant strategies were based on case studies, surveys, interviews, etc. Stapel, et al explored the communication problems that distributed development teams encounter [58]. Damian et al investigated how to manage stakeholder interactions [14] and explored the concept of requirements-driven collaboration, and described the ways in which development teams coordinated their efforts when working on interrelated requirements. Their researched data was assembled from their case study’s project plans and questionnaire responses [15]. Managing task dependencies between remote locations was Herbsleb’s focus [12]; and Taweel et al studied distributed requirements gathering practices and project knowledge, during an ongoing software development project [16].

Some of these studies have identified specific issues related to activities and tasks in the requirements phase of a project. For example Herbsleb highlighted problems of impeded communication, incompatible support environments, cultural differences, and disparities in domain expertise across sites [12]. Taweel observed that communication and coordination challenges resulted in delayed projects, poorly-defined requirements, and repetition in the software development effort [16]. Other researchers have identified additional challenges in managing requirements when project stakeholders are distributed [14, 15, 17]. These challenges include difficulties in achieving effective stakeholder interactions in order to get to a shared understanding of the project; acquiring and managing knowledge through identifying and communicating with the people who know the most about the requirements; planning meetings with consideration to the different location time-zones; misunderstandings that can occur due to language and cultural differences; and implementing an effective change management process. An additional major

challenge of global RE is the lack of opportunity for informal communication including informal interactions such as impromptu face-to-face conversations at the copy machine or in the break room where information and ideas are often exchanged.

The vast majority of the studies on global development have focused on the overall software development life cycle, leaving numerous open questions related to how the requirements process can best be implemented in a globally distributed environment. In contrast to prior studies, our research focuses on the communication and collaboration between the members of the project team responsible for eliciting and analyzing requirements, and the stakeholders who are the source of these requirements, including subject matter experts, (SMEs) and end-users. We obtained our data by interviewing industry professionals who were responsible for developing and managing software project requirements from a diverse set of organizations.

As detailed in the previous section there are four core requirements development tasks that an RA needs to accomplish – Gather, Prioritize, Document and Manage. First the RA needs to elicit and gather the requests/requirements from the stakeholders, (Elicit). All of the elicitation techniques require the entire project team to communicate and collaborate with each other; which can prove more challenging when members of the project team are not situated in the same location and have limited opportunity for face-to-face interaction. Analysis, requirements negotiation and prioritization are the next tasks, (Prioritize), to be accomplished during the analysis phase; followed by documenting the requirements specifications, (Document). Finally there is the requirements management task, (Manage), specifically storing, tracking and accessing. When stakeholders are geographically dispersed, the RA may encounter communication and coordination challenges as they facilitate and lead the stakeholders in participating in each of these four core activities.

During a traditional software development project the stakeholders assemble together in the same location to discuss the project requirements. The PM utilizes industry standard processes as described earlier in this chapter, to engage the users in providing and prioritizing the requirements. The PM's responsibilities include identifying and assembling the stakeholders; organizing requirements gathering session and scheduling follow-up meetings. These meetings provide all project participants with opportunities for their voices to be heard and to receive immediate feedback. Developers use these sessions as opportunities to share their designs and prototypes in order to receive clarification and confirmation. As the requirements become solidified the stakeholders then meet to discuss and set priorities. This allows

everyone on the project to participate in prioritizing all of the projects requirements, not only their own, quantitatively and/or qualitatively [46, 47].

Today's environment of globally deployed software projects requires stakeholders from dispersed locations to be included in the task of discovering and specifying requirements. Though many challenges exist, the benefits of involving more stakeholders in the requirements elicitation process include the ability to capture a more complete set of requirements, explore options in greater depth, consider more perspectives, increase stakeholder buy-in to a project, and emerge tradeoffs and conflicts earlier in the software development lifecycle. Many organizations address this need by selecting a representative group of stakeholders and having them travel to a central location for a series of face-to-face brainstorming sessions. A facilitator, usually a PM, RA, or BA, works with the stakeholders to define and prioritize requirements. These face-to-face meetings provide stakeholders with the opportunity to explore, articulate, prioritize and negotiate requirements together in one setting. The success of the elicitation process is also made possible if the representative stakeholders chosen to participate possess subject matter expertise, the power to make decisions for their group, and the ability to communicate and collaborate with other stakeholders [2].

There is also an increasing trend for technology to be utilized to support remote communication and collaboration [26]. For example, forums, wikis [27, 59, 60] and online versions of commercially available requirements management tools are designed to facilitate collaboration between geographically distributed stakeholders without requiring in-person meetings. Media-rich tools, such as telephone conferences, email and chat, also allow distributed stakeholders to work synchronously or asynchronously and share information as necessary.

2.3 Social Networks and Modeling

A social network is a structure comprised of individuals or groups or organizations, referred to as *nodes*. These nodes are connected to each other by some type of relationship such as flows of communication, collaboration, trust, etc., or interdependency, usually depicted as *edges* or *lines*. Social network analysis, which focuses on the patterns of interactions between the nodes, is a useful technique for graphically explaining and analyzing stakeholder interactions and relationships [61, 62].

deSouza et al [63] studied the relationships that existed between software developers due to the technical dependencies of the components they worked on. deSouza's team created a tool that analyzes software

dependencies in order to facilitate the necessary communication and coordination of development activities. Their tool, Ariadne, generates sociograms which depict dependencies between developers in charge of different software components. When used in the context of social network analysis a sociogram represents a set of items i.e. coders and components, connected to one another by edges, the dependencies.

Damian, et al. [15] introduced the concept of a requirements-centered social network, (RCSN), and described it as a graph that illustrates the relationships and communication paths between project members working on an individual requirement. Each graph represents an individual requirement while each individual node depicts a development team member and each line depicts a communication path. Damian's research team used their RCSNs and social analysis concepts to study task awareness and collaboration patterns of developers working on the same or related requirements [26].

There are numerous notations that have been used to model people and their various interactions within an organization. For example, at the enterprise level, organizational models are commonly used to depict the flow of information between levels of management. They typically represent the reporting structure or hierarchy of an organization, department or project; and depict the extent to which roles, responsibilities and power are delegated and coordinated [22, 23]. However organizational models fail to display stakeholder interactions at a person-to-person level, to show how documents are shared between stakeholders, or how tools are used to support collaboration and communication. Organizational models are therefore insufficient for expressing the interactions and processes of a distributed requirements process.

An extensive analysis of other visual notations in the field of requirements engineering failed to uncover any method that provided all of the concepts needed to visualize the specifics of a globally distributed requirements project. For example, the stakeholder onion model [21] identifies stakeholder roles and associated stakeholders for a project but does not consider their interactions. Stapel et al's FLOW Mapping [58] is a proposed notation for modeling and depicting flows of information. This technique is meant as a communication planning and management tool that can also be used to incorporate informal channels of communication between developers and other stakeholders during distributed development projects. Though similar to our proposed notation in that the concepts of stakeholders, locations and artifacts can be modeled; the FLOW notation consists of only six generic elements. For instance there is no visual differentiation between different types of project stakeholders, as all stakeholders are represented as "fluid storage".

Use case diagrams from the Unified Modeling Language [19, 20] and other similar diagrams focus primarily on modeling the actual requirements as opposed to modeling stakeholders in the project environment. The User Requirements Notation (URN) which is a combination of Use Case Maps (UCMs) and the Goal-oriented Requirement Language (GRL), two complementary notations for representing scenarios, and goals and NFRs, respectively, is mainly for software and requirements engineers to use to create and review requirements specifications [64]. van Lamsweerde's [65] goal modeling framework incorporates the KAOS method and tools to construct goal and sub-goal models of complex systems, but not the project team interaction. Another goal-oriented modeling technique *i** is used for graphically describing a projects' actors and their goals [66, 67]. Rich picture diagrams [68] could be used to depict all of the elements of a distributed requirements engineering project, but they are generally developed in an ad-hoc way, on a project by project basis, in order to intuitively represent the users and their interactions with the system to be developed. Their ad-hoc nature makes it difficult to draw comparisons across projects. Furthermore, most literature on rich pictures depicts them as a tool for modeling requirements, and not for modeling the project environment.

Since it is clear that existing modeling techniques are therefore inadequate for expressing the kinds of interactions that occur across most distributed requirements engineering projects, we created our own modeling notation, CGREN to graphically depict distributed requirements engineering activities and to help document and analyze each of the projects that we discussed with industry professionals. We developed a meta-model and associated visual notation for modeling the stakeholder roles, locations, communication paths, shared documents, and tool usage.

While prior research [15, 17] has mostly concentrated on individual development team members; our models focus on the customers, clients, project specific stakeholder roles, and project team members responsible for gathering requirements. Our work expands on the concepts of social networks to model the process by which groups of stakeholders communicate during the requirements development process. The social organization of the stakeholders, their formal and informal collaborations, and adoption of relevant tools are all included in our model.

Though Damian's RCSNs provide several necessary elements for our research, such as stakeholders, sites, and communication paths, they lack additional concepts such as the use of shared resources or communication media that could provide useful contextual information about the project. Our notation builds upon concepts of RCSN, such as roles and activities to model the distributed requirements gathering process; and it also introduces a number of additional concepts. Because our models capture a

more general picture of project-level interactions, they are referred to as Requirements Gathering Collaborative Networks (RGCNs) [26, 28].

CGREN is used to construct RGCNs models that depict the current practices and identify potentially problematic techniques in each of the studied organizations; and as the basis of process patterns that can be used to guide future distributed development projects. A complete description of our meta-model, CGREN and RGCNs can be found in section 5.2. The taxonomy and visual notation presented in this thesis are both derived from findings of our research.

Chapter 3

Methodology

We conducted three qualitative real-world research studies to better understand how requirements are currently elicited, gathered and prioritized with stakeholders at multiple distributed locations. Beginning in the Fall of 2008 we conducted a study of open source software (OSS) projects. For this particular research project we visited web-based project forums to analyze the current processes, tools and general forum cultures; and surveyed the OSS project users and managers/administrators about the processes used to elicit and prioritize users' requests. In open source projects, software is used, built, and maintained by a community of users, and as a result product features, or system requirements, tend to evolve in response to specific requests from users. Due to the distributed nature of these projects, in-person meetings and discussions can be costly and impractical; therefore communication occurs primarily via forums and email exchanges. The process of gathering, eliciting, and prioritizing requirements is quite different in open source projects than in more traditional ones; where users tended to be co-located and therefore can converse and exchange ideas more readily. We conducted this study in order to understand the issues and challenges of managing feature requests in open forums, and the processes that are used to elicit and prioritize the stakeholders' requests - their needs, wants, and desires.

Specifically, our study explored and evaluated the forum-based requirements gathering and prioritization processes adopted by vendor-based open source software projects. The effectiveness of these various practices was evaluated, through observing how feature requests are managed in the forums, and also through a survey of vendor-based forum users and project managers. Our results highlight practices that could generally lead to more effective requirements processes in web-based requirements gathering and prioritization tools. We identified several strengths and weaknesses of using forums to support online elicitation and prioritization processes [29].

Next, in May of 2009 we began our second study which entails conducting and analyzing the findings from a series of in-depth interviews with industry professionals responsible for gathering and managing requirements from globally distributed stakeholders. Our goal was to examine the processes, tasks, and

strategies for eliciting, analyzing and specifying requirements in projects with geographically distributed stakeholders [25]. During traditional software development projects, stakeholders are usually co-located and participate primarily in face-to-face meetings to accomplish tasks of gathering, analyzing, specifying and prioritizing the requirements. However in today's global development environment, project stakeholders are often remotely distributed, causing these and other requirements development activities to be significantly more challenging to accomplish. Although it is possible to bring them together for a series of centralized face-to-face meetings, it is not always ideal in terms of cost and effort. Furthermore, the recent popularity of Web 2.0 technologies has significantly altered the ways in which society can communicate and collaborate to perform shared tasks [7, 14, 27, 69].

For this study we concentrated exclusively on the challenges and best practices of distributed requirements activities. Specifically we studied the commonly recurring stakeholder roles, communication paths and techniques, methods and tools used to document stakeholder requests and transform them into more formal requirements; adopted processes, organizational patterns, and techniques for conducting specific requirements activities in a distributed setting. From this we identified and documented challenges and successful strategies in the form of organizational and activity patterns [24].

Then beginning in the Fall of 2013 we conducted a follow up study in which we engaged industry professional in utilizing our CGREN technique to model their own distributed RE projects; to help evaluate and refine our novel framework.

3.1 Grounded Theory

We did not begin our research with any preconceived ideas. We were interested in what was happening in industry, so we chose to follow the grounded theory scientific approach in order to allow pertinent theories to emerge from the data itself. Theory derived from data is more likely to resemble reality, since it is based on real-world exploration and observation. Because a grounded theory is derived from data, it is likely to enhance our understanding, and offer insight, about a particular phenomenon; as well as imply possible guidelines and methodologies.

Dagenais, et al, describe grounded theory as a qualitative research approach that entails creating a theory in the context of a process that is grounded in the empirical data [70]. Easterbrook, et al note that grounded theory is a “technique for developing theory iteratively from qualitative data.” They continue, “In grounded theory, initial analysis of the data begins without any preconceived categories. As

interesting patterns emerge, the researcher repeatedly compares these with existing data, and collects more data to support or refute the emerging theory” [5].

In comparison to a quantitative approach, grounded theory is qualitative - it is more descriptive; which means that researchers collect information on the meaning that respondents attach to their experiences and opinions. As detailed later in this chapter, we incorporated open-ended questions in our surveys and questionnaires; a technique that allows us to collect details in the users’ own words; in order “to learn about people in their natural environment in order to identify” [71] how they accomplish certain requirement engineering tasks. The purpose of using qualitative surveys is to provide depth and individual meaning to the questions of interest. Our focus was the depth and uniqueness of each response.

The three major components of the qualitative research approach [72] are (1) data from a variety of sources such as interviews, documents, observations, etc; (2) procedures for interpreting and organizing the collected data, which for our research was our RGCNs and patterns; and (3) oral and written reports, i.e. conference and journal papers, and theses.

Grounded theory includes a series of steps to ensure that the study results are unprejudiced and repeatable. The steps include (i) identifying a problem through observation (ii) formulating a hypothesis and (iii) testing the hypothesis through data collection and analysis.

We started with preliminary investigation, observations and results of prior research; and refined our data collection methods as we focused in on one aspect of global software development. This research dealt with a particular practice. Specifically, we wanted to answer the following distributed requirements engineering question - *How are requirements elicited, gathered, documented and managed when the person responsible for gathering the requirements, the RA, is not co-located with the people who are the source of the requirements, the stakeholders?* By “how”, we mean, what processes, tools and techniques does the RA utilize to accomplish the aforementioned requirements development activities. Our intent was to identify the common practices that worked well and the common challenges that practitioners encountered.

Originally the questionnaire for our second study contained over 21 questions, based on the researchers’ experiences in industry and feedback from practitioners’ reviews. Five requirements-gathering collaborative network (RGCN) base models were also included to help elicit additional feedback from participants. Conforming to the grounded theory approach, our questionnaire evolved as the research

progressed. As we learned more about the general processes the RAs employed and the pitfalls they encountered, we enhanced the list of interview question with additional specific questions. To help ensure the quality of our data, transcripts of previously completed interviews included the new questions as follow-up questions for the study participants to answer.

The first ten interviews conducted helped to flush out the meta-model and concepts. For example the definition of the *site* component was update from *physical locale* to a more high level description -- *a place at which one or more project stakeholders are situated. A site could refer to a single building or a group of buildings in close proximity to one another* [24]. Also, during succeeding rounds of interviews, more why, when and what questions, have been discussed i.e. *when have you found technology most useful*. Refer to Chapter 5 of this thesis for a discussion of the meta-model and its components; and Appendix A for the complete list of interview questions.

The questionnaire developed for our final study, where we engaged industry professionals to evaluate the usefulness of our proposed modeling technique; was also refined after a practice session. Refer to Appendix A for the complete list of discussion and de-briefing questions.

The qualitative research approach is applicable to these research projects since we were trying to understand the meaning of RA experiences as they performed a specific set of tasks. To this end we reached out to industry professionals to find out what they were doing and thinking. We used qualitative methods to investigate a significant component of the global software development process about which little is known. The results of the data analysis and interpretation, these theories, are presented as descriptive models that can be used to explain and predict [72] a variety of requirements engineering activities; and applicable organization and activity patterns.

For our research quantitative data was also collected, such as the number of sites and stakeholders, along with the number of them performing certain activities, using particular tools and/or processes. Research participants were also asked questions about how long they had been in the SE profession and the number of years of experience they had with RE-related responsibilities. We also gathered counts of requirements generated for projects; and project size in terms of dollars and man-hours.

Two operations necessary for developing theory are asking questions and making comparisons, that is discovering what is same and what is different. In the next section we describe our survey instruments and beginning in Chapter 4 we discuss our data analysis.

3.2 Surveys and Questionnaires

Fink describes a survey as a “system for collecting information from or about people to describe, compare or explain their knowledge, attitudes, and behavior [71]”. A survey is a collection of questions, or a questionnaire, which is a useful research tool for collecting meaningful data [73]. According to Gillham “questionnaires are one of the tools of the population survey – a main research method. Surveys usually aim at a comparative and *representative* picture of a particular population. Social scientists use the term ‘population’ in the special sense of the group or list they are sampling from; they also speak of this list as a ‘sampling frame’ [73]”. In other words if there are large numbers of people that you want to study, a practical approach is to observe and interact with a sample of the appropriate groups.

We selected 2 types of survey instruments [71] for our studies:

- A Self-Administered Questionnaire that will be referred to as a survey from this point forward, which study participants can complete on their own. A survey can be mailed, completed online or on site.
- An Interview, using a prepared list of questions that requires participation from a minimum of two people: a person to ask and another to answer questions. This activity can occur face-to-face, via video or telephone conferencing.

Both surveys and interview questions include open, also referred to as open-ended and close questions. An open-ended question allows the participant to answer in their own words. These types of questions are best suited when the answers are in terms of opinions, beliefs or judgments, since a small list of available choices are less likely to be representative of all possible responses. Open questions are extremely useful because a researcher can garner more information if participants are given the opportunity to respond as they wish.

When using a close question, which entails providing a list of possible answers, “researchers commonly find themselves resorting to techniques that force responses into predetermined categories [73]”. The element of discovery tends to be reduced if all possible questions and answers are decided on beforehand. Close questions are best suited for gathering specific types of data, i.e. male or female, system user or system administrator, age, number of years with the company, job title, etc. Analysis of closed questions responses tends to be straightforward; while open-ended questions require more content analysis, for instance organizing similar ideas into meaningful categories.

Surveys are effective tools for gathering information from a large number of people. Distributing surveys are less expensive than organizing and conducting interviews since a survey allows a researcher to contact more people, and quickly collect more information. Other advantages [73] to using a survey are that research participants can complete surveys as their time permits and can be guaranteed anonymity, since they do not have to include their name. There is also a lack of interviewer bias when gathering data via a survey instead of conducting interviews, as research suggests that a different interviewer may get different answers if the activity of interviewing is not properly planned and organized.

There are some challenges in conducting research using surveys and questionnaires, including overcoming the typically low rate of response. This means that researchers must find a way to make the survey instrument interesting and worthwhile enough for the participants to complete, a way to motivate potential participants. It has been observed that participants will expend effort if the survey is relevant to them and/or related to something they deem important. Note that we are bombarded by questionnaires and surveys every day, i.e. rate customer service, job reviews, etc., and must decide if completing them is worth our time. Often times, too, participants worry about how the collected data will be used and if their responses can be traced back to them.

3.3 Research Instruments

In developing our online surveys and interview questions, we used a cross sectional descriptive design [71] as we are interested in gathering information from currently existing groups of people at a set point in time. Our research entails qualitative data-gathering. The surveys and interview script contain open-ended questions, in order to get details in the respondent's own words. For all of the studies we had access to small samples of our domain populations, i.e. professionals and users with similar responsibilities. Fink notes that qualitative surveys are useful "to provide depth and individual meaning to the questions of interest [71]".

To mitigate the challenges noted above, we carefully developed each online survey and interview questionnaire, in order to acquire quality data. Our survey instruments were double-checked by members of our targeted communities, real world participants who are current users of the products and processes we are examining.

To help insure the reliability of our survey instruments, we had industry professionals, similar to our target audience evaluate the wording and appropriateness of each question. Prior to using each of our

instruments, we conducted a cognitive pretest [71], meaning that we sent a draft to these professionals and asked them to review each survey question and consider the following:

- a) What does this question mean to you?
- b) Is there a better way of asking this question?
- c) What do the response choices mean to you??
- d) Given the choice of two formats, which is better?

Pilot tests were conducted as well.

A usability test was performed for the online surveys, to ensure that participants would be able to access the surveys and select appropriate responses. The online surveys allowed respondents to participate at a time and place of their choosing. The list of interview questions was refined and augmented as a result of early response analysis. All interview respondents were asked all questions, since we were able to include new questions as follow up items when sending transcripts of our interview session to them, for their review and approval. By distributing the interview questions well in advance of the scheduled interviews, the RAs had the opportunity to review them and prepare for our discussion, and to arrange an interview time most convenient for their schedules. Our goal in doing this was to help the respondents feel more relaxed and less pressured. Introductory discussions and information explicitly described the purpose of each study, and helped to alleviate the participant's anxiety about our research and use of collected information. We describe each of our instruments in greater detail in the sections below.

3.3.1 Open Source Software Projects Surveys

We used a survey to collect information directly from OSS project users. The objective of our online surveys was to explore and evaluate the forum-based processes used to elicit, gather and prioritize software enhancement requirements of vendor-based OSS projects. These OSS processes are quite similar to the requirements gathering forums that can be used for traditional projects to help facilitate requirements engineering in similar circumstances.

Two online surveys were created, one each for OSS project administrators and users, which contained questions that would help us gain a better understanding of the administrators' requirements elicitation, prioritization and organization methodologies, and the real activities that the users perform when posting their requests/requirements. As described earlier, before distribution and use, both surveys were reviewed by researchers with prior industry experience. The surveys were then evaluated by several software engineering professionals. These industry professionals were similar to our targeted audience of OSS users, and so were able to provide feedback regarding the clarity and suitability of the questions.

Participation from the OSS project administrators was solicited via email. We requested user participation by posting a link to the survey on the project forums [27]. A copy of each survey can be found in Appendix A.

For the online surveys only Internet Protocol, (IP) addresses were collected. An IP address is a numerical identification assigned to devices i.e. computers, printers, etc., that are part of a computer network [74]. Participation anonymity was maintained since an IP address is not tied to an individual. With the IP address we attempted to guard against a single participant completing multiple surveys.

In order to include a diverse group of opens sources software projects in our study, we asked IT professionals and software engineering students to suggest the open source projects in which they most frequently participated. The inclusion criteria for our exploratory surveys were vendor based open source projects that included over 3000 forum postings. From the initial fifteen candidate projects, eight were selected for further analysis. These eight projects represent a variety of open source software domains, including groupware, system management and gaming. The description and results of our OSS projects study are presented in Chapter 4.

3.3.2 Information Technology Industry Interviews

We used interview questions to collect information directly from requirements analysts in a variety of industries. The objectives were to find out: how RAs elicited and gathered requirements from stakeholders at multiple distributed locations; how they worked with stakeholders to prioritize requirements; how the requirements specifications were documented, and subsequently managed. In essence we are interested in how the RAs performed the requirements development tasks. The majority of interview questions were open-ended. Our discussion topics are straightforward questions in order to extract accurate information from industry professionals.

The interviews were conducted via telephone for the most part. Each RA was asked to think about a specific recent project in which they were responsible for eliciting and gathering requirements from geographically distributed stakeholders. Every participant answered the list of over 21 questions [26]. At least one hour of the RA's time was requested for each interview, which was recorded and later transcribed for in-depth analysis. Researchers requested study participation from their industry contacts; and also solicited the help of their academic and professional, colleagues and associates in extending the call for participation.

Although the interviewee's name was collected, participants were assured that their own and company names would be stored separately from all gathered data and would not be included in any research publications. A complete list of interview questions can be found in Appendix A. The description and results of the IT industry study are presented in Chapter 5.

3.3.3 Requirements Engineering Modeling Sessions

This research study entailed our observation of industry professionals utilizing and evaluating our CGREN as they modeled distributed requirements engineering activities associated with their respective projects. At the beginning of each session we discussed the requirements phase of a sample project with the participant and then demonstrated how to model the project using our framework at the whiteboard. Participants were then invited to model the RE activities for one of their own projects. Each research session was conducted in office space equipped with a whiteboard and markers for creating the RGCN models. At the end of each session researchers solicited feedback from the participants regarding the proposed framework and its usefulness in modeling activities in their real-world projects, i.e. planning distributed RE activities, early detection of potential issues, etc. Each session was audio recorded and transcribed for further analysis.

Again the participant's name and company name were collected and maintained separately from the other data gathered during the research session. The de-briefing questions can be found in Appendix A. The description and results of this follow up study are presented in Chapter 6.

3.3.4 Visual Notation Icon Selections

During the Information Technology Industry Interview study the questionnaire included base models that depicted two generic project stakeholder roles, that of the requirements analyst and one or more stakeholders. As part of our analysis, a project description including models of each of the requirements activities discussed was created to share with the study participants for their review and verification. These reports were created using standard word processing and graphical presentation tools.

Since creating the models using multiple standard word processing and graphical presentation tools proved to be time-intensive; the researchers began designing and developing a web-based, global requirements engineering tool / application, (GRETA), that could be used to produce the distributed requirements activity models in a fraction of the time. During the design phase researchers realized that once developed this application would also enable practitioners and other researchers the ability to

develop their own distributed requirements activity models. It also became apparent that a proprietary set of visual notation icons would be necessary.

A graphic artist tasked with designing an initial set of icons representing entities and communication in the meta-model was added to the research team. The prototyped icons were first evaluated through a series of informal user tests by approximately ten research lab associates. The icons then progressed through an iterative re-design and online survey evaluation process that occurred between November 2010 and February 2012.

In total fifty graduate students and IT professionals of culturally diverse backgrounds reviewed and provided feedback on over 60 icon designs that resulted in the nine stakeholder, four communication media and 5 artifact icons included as components of the visual notation. Refer to Chapter 5.2 of this thesis for a discussion of the meta-model and its components.

Chapter 4

Open Source Software Projects Data Analysis

The data described and analyzed in this chapter is the outcome of the OSS projects surveys. This was a two-part study in that we first accessed each of the OSS project forums to examine and evaluate the tools and techniques that were available. Our second step entailed surveying the forum users. Refer to Chapter 3 for a full discussion of our data-gathering methodology. Details and results of this study are also published as a chapter in Open Source Software Developments in 2010 [29].

We recognize that the limited response rate is a possible threat to the validity of our study. We used the “number of posts” information that was listed on the OSS project forums to gauge each project’s level of activity, as our criteria was to include only open source projects with a minimum of 3000 posts. Unfortunately though, statistics regarding the number of registered users was not provided to forum participants, so we were unable to determine the percentage of user participation. OSS project administrators, who did not want their user community involved in the study, removed the postings requesting participation and the survey links. Still we believe this research is relevant and applicable since it represents real-world requirements engineering practices that are currently taking place in the OSS domain.

4.1 Open Source Software and Surveys

OSS development represents a collaborative community-based effort to develop software in which the users participate in deciding what features to build, and a subgroup of developers participate in designing the solution, writing code, and deploying and maintaining the system [75]. As a result of this development process, requirements and functionality usually evolve in direct response to specific user requests. In the OSS domain, applications and the application users tend to be geographically separated, making it difficult and costly to engage in face-to-face requirements gathering sessions. Because of the distributed nature of this development environment, OSS projects rely on wikis and online forums to communicate with their communities of stakeholders [69, 76].

There are currently two common OSS development models. The first is the user-based model in which software is developed collaboratively by the users, and in which integration of new features is governed by an executive body. The second model represents a vendor-led approach in which a specific vendor controls the development and integration of new features. Although source code is released to the users to develop additional features, the primary responsibility for development is carried by the vendor. This research concentrated on vendor-managed projects.

Even though OSS projects are quite different from more traditional software development projects, it is still crucial for OSS vendors to understand and meet their stakeholders' product-related needs. However, many of the techniques used for traditional requirements development practices are unviable or difficult to implement in an OSS development environment for several reasons. First, unlike other projects which may have more defined phases, OSS projects tend to be highly iterative with strong expectations for ongoing and continual improvements to the product. Secondly, instead of eliciting and gathering requirements from a carefully selected and representative group of stakeholders, OSS projects generally support a very open elicitation process in which any and all stakeholders are welcome to participate. In addition, it is difficult to facilitate face-to-face meetings in OSS projects, and so elicitation, prioritization, negotiation, and other requirements related activities are generally conducted using web-based tools such as wikis and forums.

Two different online surveys were used to gather information from open source project administrators and from forum users, respectively. For purposes of this research, an administrator was defined as an OSS vendor staff member; and a user as a current or prospective OSS user or provider. The survey questions were designed to gain an understanding of the actual activities that users performed when contributing feature requests; and the administrator's methodology for eliciting, prioritizing, and managing these requests. The survey was reviewed by researchers with industry experience. It was then appraised by several software engineering industry professionals who provided feedback regarding the clarity and appropriateness of the questions. These professionals were similar to our target audience of OSS users. We solicited participation by contacting the project administrators via email. A request for user participation and a survey link were posted in the OSS user forums. A copy of each survey, the solicitation messages and forum postings can be found in Appendix A.

In order to include a diverse set of open source software projects in our study, we polled software engineering students and IT professionals for suggestions about the open source projects in which they participated; and also browsed through the forums in Source Forge. Only vendor based OSS projects with

at least 3000 postings were selected for the study. Initially fifteen candidate open source projects were identified and after further investigation and evaluation, the analyses and results from eight of these projects; representing a variety of software domains, including games, groupware and systems management, were selected to be included in this study. Additional OSS projects were included, but one was dropped when it was discovered to actually be a community based forum, and the others were dropped because we received fewer than 4 responses to the survey. In the remaining forums, we received three responses from administrators and 107 from individual users. Given the low response from administrators, the remainder of this section discusses their responses only in a qualitative manner.

This study included well-known open source projects in the following industries: a Java application server, password manager software; a source code editor written in C++; a file manager for Windows; a web-based enterprise resource planning (ERP) tool; a client/server tool for next-generation messaging and collaboration; a virtual world environment/game; and a customer relationship management (CRM) tool. The statistics of these projects are summarized in Table 4.1. Specific project names are not used in order to protect the anonymity of the forums.

OSS Project	Number of posts	User responses	Admin responses
Password Manager	> 17, 000	16	1
Source Code Editor	> 32,000	5	0
File Manager	> 4,000	13	0
ERP	> 20,000	26	1
Java app server	> 440,000	30	1
Virtual world/game	> 2,000,000	10	0
Groupware	> 110,000	7	0
* CRM	> 125,000	0	0

** Included in observational part of study but did not participate in surveys in accordance with administrator's negative response.*

Table 4.1. Number of posts and responses for each of the surveyed forums

During the first phase of this study we visited each of the forums and analyzed the available tools, adopted processes, and general culture of the forum. Some of the results from this inspection are reported in Table 4.2 and discussed in Section 4.1.2.

		Topic Search		New Feature Request				Dedicated Forum			Status Information			
Legend:		Browse	Search	Comment	Prioritize	Vote	Monitor	Feature Request	Issue Tracking	Sort	FR Status Field	FR Process doc	Release Schedule	Remove / Archive Old FRs from Forums
<input type="checkbox"/> A Admin or developer assigns priority														
<input type="checkbox"/> S Stakeholder can assign priority														
<input checked="" type="checkbox"/> Yes														
No Predefined Structure of Topics														
Password Manager		•	•	•	A		•	•	•	•	•			
Source Code Editor		•	•	•	A		•	•	•	•	•			
File Manager		•	•	•	A		•	•	•	•	•			
ERP		•	•	•	A		•	•	•	•	•			
Predefined Structure of Topics														
Java app server		•	•	•	A	•	•		•	•	•			
Virtual world/game		•	•	•	S	•	•		•	•	•	•	•	•
CRM		•	•	•	A			•	•	•	•			
Groupware		•	•	•	A	•	•				•	•	•	

Table 4.2. Features observed in the Open Source Forums

4.2 Feature Requests

All of the observed Vendor-based forums included an open and inclusive process for eliciting and prioritizing requirements from the users. In each case this process was at least partially conducted over the web using an open forum. For the purposes of this study we were interested in discovering the different techniques used by vendor-based OSS projects to elicit, negotiate, and prioritize software enhancement requirements and to evaluate their effectiveness. Some of these activities involved the general users while other activities were conducted solely by the vendors.

Most OSS forums follow a similar process for managing feature requests. This process is depicted in Figure 4.1. A user first logs in to the site, and then finds an appropriate forum and topic in which to enter their new feature request. Generally, a user can either create a new thread or add their request to an existing thread. Most forums allow users to browse through various forum threads looking for discussions and comments that are related to their topic of interest, or to perform a more structured search by using keywords or other attributes such as authors' names.

OSS administrators use a variety of techniques to encourage users to submit new feature requests, ranging from very passive to more proactive methods. For example, some administrators post details regarding planned releases and ideas for future development in the project's "Announcement" forum, while others not only post questions and solicit stakeholder feedback in the regular forums, but also maintain a dedicated "Feature Requests" forum. New requests can also be generated from bug reports and other sub-

forums of the issue tracking forum. Many forums also provide users with the option to vote for an existing feature request.

In many forums, the administrator is also responsible for updating and communicating the status of the request to the users. Though this process appears relatively straightforward; the results of this study reveal that forum users are often unclear as to the status of their requests; or in fact even whether their requests are being considered for a future release.

Also illustrated in Figure 4.1 is the typical OSS administrative process for handling and prioritizing a newly submitted feature request. Ideally, the administrators, or a team of reviewers, examine each feature request, determine the feasibility of developing and prioritize it, in relationship to the development team's schedule and resource availability.

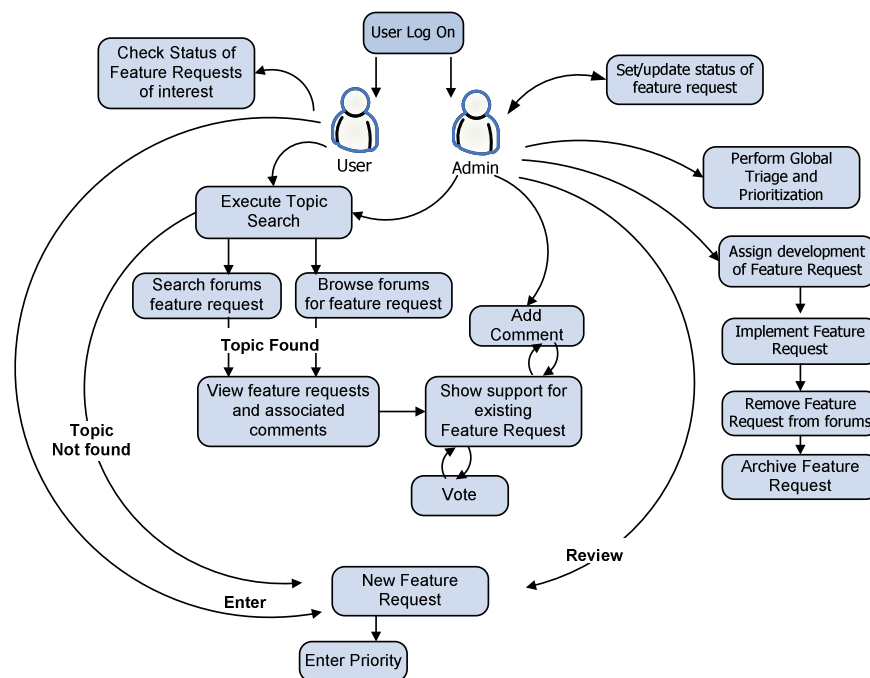


Figure 4.1. Vendor Based Open Source Process for Entering and Managing Feature Requests

4.3 Research Findings

The initial analysis of forum based requirements development processes in OSS projects helped to identify some of the strengths and weaknesses of implementing an online requirements development process using a forum. The primary strength is the inclusive nature of forums; which enables large numbers of stakeholders from geographically distributed regions, operating in different time-zones, to engage in the feature gathering process. This inclusive process provides the vendor with a much more complete view of the needs of the average user and also builds a sense of community amongst the stakeholders.

We also identified several activities that were difficult to perform in a forum. These activities are summarized here and then discussed in greater detail throughout the remainder of this section. They include:

- (i) **Creating Collaborations:** Difficulties in bringing relevant groups of users together to discuss related needs. In many cases stakeholders with common interests never engaged in shared conversations.
- (ii) **Prioritizing Features:** Problems in capturing users' priorities. Over simplistic voting schemes were either ignored by forum managers or failed to unearth stakeholders' real priorities.
- (iii) **Engaging and Communicating:** Problems in establishing two-way conversations in which administrators communicate process and decisions, and seek clarification from stakeholders or otherwise engage users in the requirements process.
- (iv) **Managing Feature Requests:** Problems in managing the status of feature requests in the forum. Users were frequently unclear whether features had been implemented, or whether their own feature requests were being considered for a future release.
- (v) **Identifying Users roles:** Lack of differentiation between users. Although some forums do allow frequent forum visitors to become recognized experts, there is no reliable way to differentiate between infrequent visitors who are heavily invested in the product versus those with more transient interests.

4.3.1 Creating Collaborations

One of the strengths of a more traditional requirements engineering process is that business and project analysts work hard to bring the right stakeholders together to brainstorm ideas and explore their product related needs. In contrast, the primary strength of web-based requirements engineering is the inclusive nature of forums; which enables large numbers of stakeholders who are not co-located to engage in the

feature gathering process, regardless of their location time-zones. Although forums do provide some structure for facilitating this process, our observations show that this task is not accomplished very well.

In a forum, stakeholders exchange ideas through shared discussion threads; however they sometimes fail to perform a thorough and successful search for relevant threads and as a result similar topics are found dispersed across multiple threads. This makes the analysis, negotiation and prioritization process very difficult to accomplish effectively.

It is the user's responsibility to search for and find appropriate discussion threads. Each of the forums we studied therefore provided both browse and search features, primarily designed to help users find relevant discussions. However, in a prior study of seven different forums [76] we observed an interesting phenomenon that over 50% of threads contained only 1 or 2 feature requests.

To explore this issue further, our survey asked users whether they searched for relevant topics before entering a new feature request. 11% of users said that they did not perform a search, and just entered their request into a new thread, while 89% of users claimed to perform a search. These results are depicted in Figure 4.2. However, the results from the prior study showing major overlap of topics between posts in individual threads and posts in larger threads, suggests that user searches are frequently ineffective.

An analysis of OSS forums suggests that browsing support tends to be very rudimentary. For example, five of the eight forums studied exhibited a very flat hierarchy of topics. However for the Java application server, virtual world, CRM, and groupware forums the administrators organized topics within a high-level topic hierarchy. This was helpful to users as they searched for relevant feature requests. In fact, one user specifically suggested that *"Requests could be categorized: User Interface, Options and Settings, File/Plugin/Feature Support, etc. Having this kind of organization helps people searching for related topics better find their own answers without duplicating requests."* Decker et al had previously mentioned similar problems when wikis were used to support requirements engineering activities, and observed that it was helpful for administrators to arrange wiki pages in a hierarchical fashion [69].

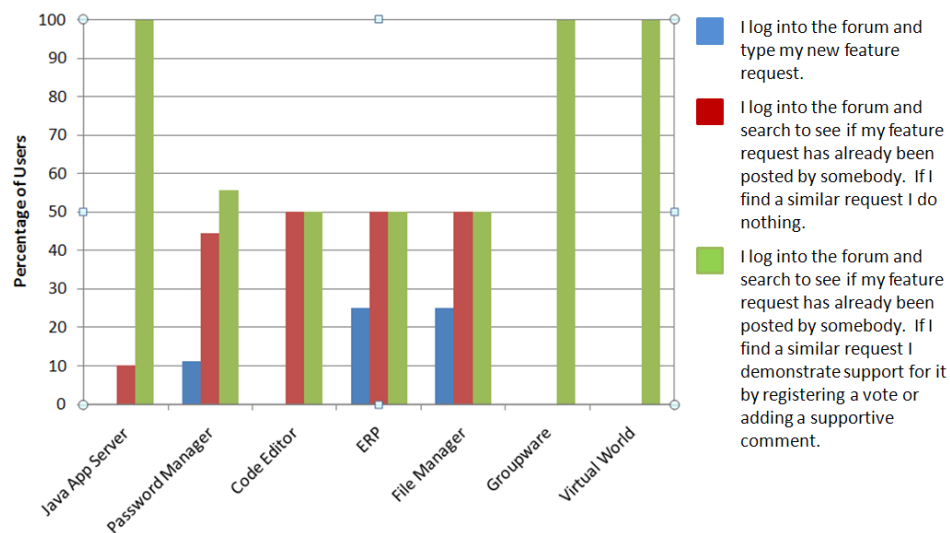


Figure 4.2. Methods Preferred by Users for Entering Feature Requests

4.3.2 Prioritizing Features

The forums we observed either provided no explicit inbuilt prioritization mechanisms or else provided a simple voting button that enabled users to register their support for a feature request. The virtual world game provided a feature that allowed stakeholders to assign priorities to feature requests. In all of the forums we observed that users attempted to prioritize certain features through adding comments. For examples, in several cases users included comments such as “*let my comment serve as a vote for this feature,*” or started their posts with the words “*Feature Request*” in order to attract attention. Although we could not substantiate this through responses to our survey, our informal discussions with forum users also suggested that one reason people created new threads for each feature request, was because in some forums this meant their feature request would be placed at the top of the list, which would make it more visible than if it had been entered as a response to an existing post.

It was clear that users wanted project managers to listen to them and to build features that were important to them. Several forums included queries from users who were obviously perplexed or annoyed that feature requests that were important to them were apparently ignored by the vendors. In one case a vendor responded to the question of “*Who decides which new feature requests to implement in a given release?*” with the comment that “*We have some polls on our website that might influence decisions,*” thereby highlighting the fact that for that particular project user opinion only marginally influenced development decisions.

Another more subtle problem was observed in the virtual world forum which provided both prioritization and voting functions. The initial contributor was allowed to prioritize the requirements, while other users were simply allowed to vote for it. This introduced an ambiguity as to whether users were voting for the feature, or agreeing to its prioritization level. For example, if a contributor had created a new feature request and assigned it a low priority, then subsequent users were unable to change its priority level, although they could cast their votes for it. In general, the forums we observed did not provide sophisticated support for the requirements prioritization process.

The three administrators who responded to our survey indicated that they were only partially satisfied with the requirements prioritization process. User responses to the question “How satisfied are you that your feature requests for new functionality are addressed by this process?” are reported in Figure 4.3, and showed that users were most dissatisfied were ERP and Java application server, which interestingly represented two of the projects with no separate feature request module. There was also a significant degree of dissatisfaction in the password manager and virtual world projects. The possible reasons for this are found in users responses to the question “Which of the following methods do you think your OSSP uses to prioritize feature requests?” The users’ responses, which are reported in Figure 4.4, indicate that in most cases prioritization decisions are made by administrators who do take users’ requests into consideration. It was interesting that in the groupware project, the users’ perception was that prioritization decisions were largely based on user input. This project notably had no users that reported being dissatisfied or somewhat dissatisfied with the prioritization method. It should be noted that the level of dissatisfaction by users of the Java application server, might be correlated to the fact that 28% of the surveyed users did not know how project administrators prioritized feature requests.

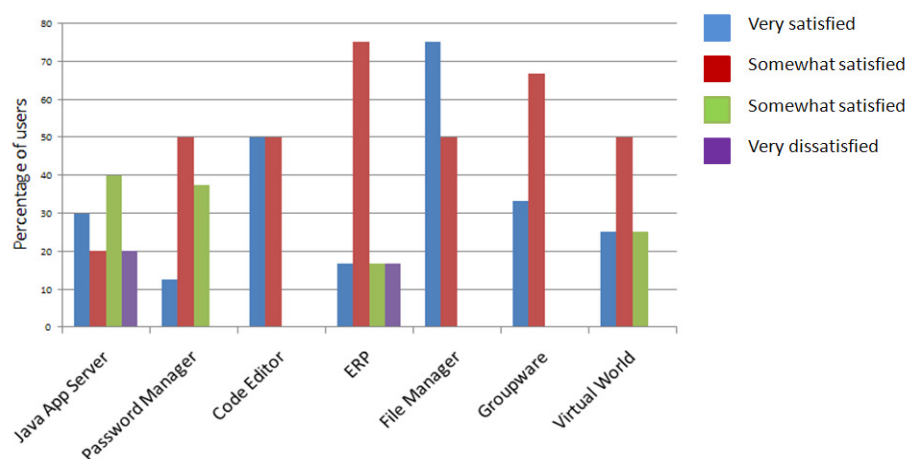


Figure 4.3. User Satisfaction with the Requirements Management Process.

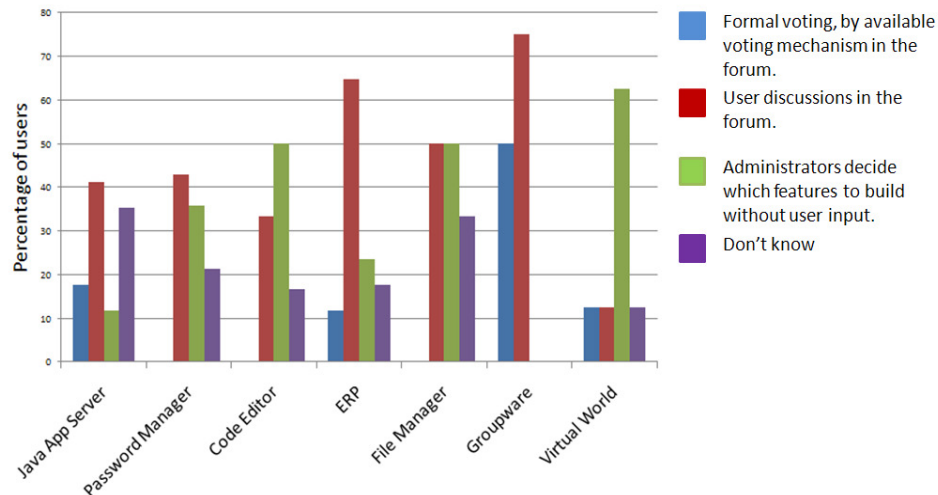


Figure 4.4. Methods for Prioritizing Feature Requests.

Users' responses from the virtual world project also provided some indication as to why they were so dissatisfied with the requirements prioritization process. Two respondents who checked the "other" option commented that features were prioritized by "*dart game, random selection*", while another user said that "*On rare occasions, there are discussions either in the blog or in the newly-active area in the forums; however, (the administrators) seem to disregard these for the most part although they are the ones who have openly solicited comments.*" Despite this perception of random prioritization, the virtual world forum does in fact provide a webpage describing how feature requests are processed. The page includes advice that "*Features are more likely to get implemented if the description of the feature is clear. For a complicated feature, a link to a specification on the wiki is a great way to help flesh out the idea.*" Nevertheless, the level of dissatisfaction in the process suggests that users do not believe their feature requests are handled in a satisfactory way despite the appearance of due process.

In CRM's very active discussion forum one of the project managers created a new discussion thread and asked users "*what would you like us to build next?*" In one sense, this demonstrated willingness to engage the user base in the prioritization process, but in another sense it demonstrated that the existing forum failed to explicitly capture this information and to create a prioritized ranking of feature requests, despite the active engagement of the user community. This problem illustrates one of the main challenges of gathering requirements in a forum, where large amounts of data must be processed in order to extract useful information. It seems that despite the active discussions in many of the forums, administrators are still not easily able to understand the users' real needs.

4.3.3 Engaging and Communicating

This study identified four primary techniques by which vendors attempted to engage users more proactively in the requirements elicitation process. First, we found that many administrators and project managers actively participated in the discussion threads. Two of the forums had special web-pages in which administrators published processes that they used, although interestingly one of them, the virtual world, was the forum for which users gave the most negative feedback about their prioritization process. The published process included a description of steps a user should take to get their feature requests noticed; however the general consensus by users of this particular forum was that the administrators largely ignored users' requests and built whatever features they felt inclined to build. The same two forums that published process descriptions also posted release schedules in which feature requests had been copied from the primary forums and ranked in order of their likely release.

Forum observations led to the conclusion that most forum administrators saw the forums as a means of eliciting information that might be considered in the requirements prioritization process. Notably absent from any of the forums however were the type of questions that analysts usually engage in during the requirements process to clarify and explore the needs of the users. We found few examples of project administrators asking users to explain something in more detail, although there were numerous peer-to-peer examples of this. This problem may be recognized by project administrators. For example, one of the administrators stated that *"We would like more involvement from the community and are experimenting with various tools to elicit more feedback."* Incidentally, this comment was made by the administrator of the ERP forum, which exhibited the highest level of dissatisfaction in response to our user survey.

4.3.4 Managing Feature Requests

This study also highlighted several problems related to managing feature requests. Perhaps the most challenging problem was that feature requests and other types of discussions were all inter-tangled in a single forum. For example, a single thread might contain suggestions for new features, personal comments from one user to another, 'how to' questions, and general comments about the software product.

Most forums had no way of extracting and removing feature requests once they had been either implemented or designated as non-implementable. This problem was especially confusing to new forum members who were trying to decide whether the product met their needs or not. In several cases they were led to believe that the product did not include certain features, simply because they found old feature

requests. There were also numerous occasions in the forum discussions that we observed users frustrated because they thought that features they had requested had been ignored, while in fact those features had been released in recent versions of the product.

Seven out of the eight forums we observed contained issue tracking features including feature status fields, and sorting features that users could use to check on the status of their requested feature requests; however only the virtual world forum had a method in place for removing feature requests from the forum once they were implemented, and also of archiving old feature requests. None of the forums had methods for reporting back to the user if a specific feature request was not considered feasible for implementation.

All three of the surveyed administrators offered suggestions for improving forum management. One suggested that *“We should remove feature requests that obviously never will be implemented, even though they are good ideas. Keeping a long list of feature requests that will never be implemented only disappoints users”*, while one user requested that *“There should be a website where new features are listed, documented and prioritized so the users can determine how possible it is for them to actually be implemented. This doesn’t change the way in which the feature requests are handled but informs the prospective users of them.”*

In general, almost all of the surveyed forums did a very poor job of managing the status of each feature request. For example, feature requests that were never implemented, generally languished in the discussion forum, and every now and again a user would complain that the feature was not implemented. Unfortunately, none of the forums we surveyed had any means of communicating that a given feature request would not be implemented. Furthermore, old discussion threads for features that had already been implemented were rarely removed from the forum. In most cases, when discussion threads were either sorted chronologically or according to activity level, old feature requests tended to drift to the bottom of the list. None of the forums provided traceability between old feature requests and the releases in which they were actually implemented, and so a user searching the forum might easily believe that an implemented feature was still an open request.

4.3.5 Identifying User Roles

Although one of the intrinsic strengths of online forums is their ability to elicit needs from any stakeholder, this is also a major limitation because all of the forums we studied did not differentiate between different users. One administrator specifically said that an area of improvement would include *“getting feedbacks directly from organizations using our product and then going over them and finding common denominators.”* To implement this type of differential prioritization requires forums to improve

their registration process so that the true role and affiliation of users are known. Understanding a user's role is an important contribution towards understanding the circumstances and urgency of their requests.

4.4 Proposed Solutions

Many of the issues outlined above can be at least partially addressed through enhancing the features in the open source forums to provide better tool support for the users as they participate in requirements related tasks. Still others can be addressed through improved processes.

The first set of solutions described below is designed to increase collaboration between stakeholders by getting them into relevant discussion threads. These solutions include access controlled thread creation and thread recommendations.

Controlled thread creation As described earlier in this chapter, a typical forum search and browse feature is designed to help a user identify relevant discussion topics. However the effectiveness of such features is often limited by the relevance of the keywords chosen by the forum user. Controlled thread creation adds an additional step to the thread creation process, using simple data-mining techniques to determine whether a new thread is associated with an existing topic [77]. If a similar theme is found in one or more existing threads, the user is asked to consider posting to one of these threads instead of creating a new one. As a result the user may choose to create a new thread anyway, to reword their post as a response to one of the suggested threads, or to entirely cancel their post. Controlled thread creation can help to minimize the number of redundant threads, and bring relevant users together in joint discussions.

Thread recommendations Recommender systems, which are commonly used in the ECommerce domain to recommend purchases to potential buyers, or to recommend movies or news stories, can also be used within open source forums to recommend discussion threads to users. Although there are many different types of recommender system, our prior work has demonstrated the effectiveness of using collaborative recommenders [78-80]. Recommender systems can therefore be usefully applied within open source forums to recommend specific discussion threads to users in order to help these users to find relevant threads and to ultimately increase the cross-pollination of ideas.

In addition to tool related solutions, there are a number of process-oriented practices that can be introduced to improve the effectiveness of a requirements gathering forum. For example, users can do a

better job finding relevant discussion threads if forum administrators create a predefined topic hierarchy within the forum [69].

Administrators and project managers also need to increase communication between themselves and the users of the forums so that users have a better understanding of the process that is used to evaluate and prioritize feature requests. Furthermore, there needs to be higher visibility concerning the status of each feature request. Project managers also need to more actively engage in the forum discussions in order to truly understand the stakeholders' needs through asking more meaningful questions. For example, in a more traditional face-to-face requirements gathering meeting, analysts often ask a user questions to explore the intent and details behind their requests for certain features. Although we saw little evidence of this in the forums, there is no reason why it could not be accomplished effectively through the discussion threads. Therefore, analysts and project managers should actively participate in the requirements gathering process, by visiting and participating in discussion threads, and requesting clarifications when needed.

Prioritization mechanisms must also be improved so that project managers can easily understand the current priorities of their users. This introduces the additional need to track roles and affiliations for each of the users so that prioritization decisions can differentiate between requests of long-term vested users versus more casual forum visitors. Our observation of the forums showed little support for project administrators and users in prioritizing needs. To increase user feedback, voting and other prioritization mechanisms need to be significantly improved so that users can provide weighted priorities for each feature request, and administrators can issue queries that return meaningful insights into the users' needs. Furthermore, as OSS users are unlikely to be available during actual prioritization meetings, tools should be provided for them to document the rationales behind their feature requests and their prioritization requests.

Finally, forums also need to be restructured so that it is possible to differentiate between feature requests, comments, and issues. The best approach we observed was to have specific forums dedicated to feature requests, and to establish a clear lifecycle for each feature request that tracks its status as created, prioritized, scheduled or deferred, and then implemented or marked as a feature that will not be considered for implementation in the near future. Once feature requests are implemented they must be removed to a separate webpage and must be explicitly traced to the release in which they have been implemented.

4.5 Conclusion

The distributed and asynchronous nature of vendor-based open source software projects has naturally led to the use of forums to capture feature requests. Conducting this study helped us gain some understanding of how these forums should be designed to provide increased support for an effective requirements process. Unfortunately as this study has shown, current forums suffer from a number of problems that inhibit the use of many normally accepted and successful requirements engineering practices. However the increased reliance on forums, in both open source and more traditional projects, makes it critical to identify these problems and address them in the second generation of forums designed to capture and manage feature requests.

The solutions outlined above represent a first-step towards this goal. Additionally, for large scale tools to be effective, we need to overcome three specific types of challenges that can be broadly classified under the three areas of data, process, and social related issues. Without overcoming these, forum-style tools are likely to always create the type of chaotic situation we observed in many of the open-source forums [28].

From a data perspective, forums and wikis tools have the potential to generate vast amounts of data that need to be processed and organized in order to extract useful information around which intelligent and productive requirements processes can be conducted. In prior work we have fine-tuned clustering techniques to organize requirements and feature requests into cohesive and distinct topics and cross-cutting concerns. Augmenting open source forums with tools that organize threads into meaningful topics, identify important cross-cutting themes, and make accurate recommendations to forum users, increase the usefulness and effectiveness of the forums. Our studies have shown that discussion threads anchored around each of these topics are superior to those created manually by human users [80]. We have also developed effective recommender systems that can be used to place stakeholders into appropriate discussion groups [80] in order to keep each individual stakeholder informed of relevant topics, and also to proactively search for stakeholders that might contribute to stagnant or incomplete discussions. Our current successes in this area suggest that applying data mining techniques might reasonably facilitate much larger scaled elicitation processes that would otherwise disintegrate under the vast amount of raw data contributed by stakeholders.

From a logistical perspective, processes are needed that identify and assign stakeholder responsibilities; promote and support multi-site collaboration; and provide robust requirements management and tracking functionality throughout the project life cycle.

From a social perspective, both formal and informal communication paths must be understood, and safeguards must be put in place to mitigate potential problems in which individuals or groups of stakeholders attempt to ‘game’ the system to push their own agendas or manipulate the outcome of the process in any other unfair way. Similarly increasing the level of conversation between vendors and users and building features that help users to prioritize their needs and describe rationales for their prioritization will result in better products that are more able to meet the needs of the users.

The move towards collaborative tools may drastically change the requirements elicitation landscape. Not only will these tools recreate practices that normally occur between colocated participants such as small brainstorming meetings, but they also introduce a catalyst for improving the way we gather and prioritize requirements through creating a much broader and more inclusive process. Implementing our solutions will help make forums a viable tool to support the requirements phase of all types of distributed software engineering projects.

Chapter 5

Distributed Requirements Engineering Interviews Data Analysis

The data described and analyzed in this chapter is the outcome of our Distributed Requirements Gathering Interviews study that began in May of 2009. Initial results of this research were presented and published at the international Conference on Global Software Engineering in 2010 [24]; however this section will also include the analysis of additional interviews. Refer to Chapter 4 and section 4.2 for a full discussion of our data-gathering methodology.

5.1 Distributed Requirements Engineering Interviews

This research project involves conducting and analyzing findings from a series of in-depth interviews with industry professionals responsible for gathering and managing requirements from stakeholders at distributed locations. Specifically we study communication paths and techniques, methods and tools used to document stakeholder requests and to refine them into more formal requirements, commonly recurring stakeholder roles, adopted processes, organizational patterns, and techniques for conducting specific requirements activities in a distributed setting. From this we intend to identify and document challenges and successful strategies in the form of organizational and activity patterns.

For each project we identified the team leader responsible for eliciting and gathering the requirements from geographically distributed groups of stakeholders. Interviews were conducted over the phone and audio recorded for later transcription and analysis. Each interview lasted for approximately one hour and was conducted between May 2009 and September 2010. Interviews included a series of relatively open-ended questions focused on the distributed requirements gathering process. The complete set of interview questions can be found in Appendix A.

Each team leader answered the questions with respect to a specific recent project in which they had been responsible for eliciting, gathering and managing requirements from geographically distributed stakeholders. Every participant was sent the questions at least two weeks in advance

of the interview so that they could be prepared. All of interviewees answered the complete set of questions. Following the interviews, researchers reviewed the transcribed interviews and analyzed the results to identify common successful practices and pitfalls across all of the projects.

As the researchers began examining and analyzing the transcripts, several patterns began to emerge, which led to additional research questions:

Did all of the Requirements Analysts employ telephone conferences for requirements gathering?

Was there a Location Spokesperson at each remote site?

Did the distributed stakeholders communicate with each other?

How were the requirements specifications shared between the RA and stakeholders?

How and where were the requirements specifications maintained?

To help document and analyze each of the projects, we employed CGREN to model stakeholder roles, locations, communication paths, shared documents, and tool usage as RGCN models. These RGCNs were used to represent current practices in each of the studied organizations, and helped the researchers to visually identify similarities and differences between the projects. Thus these models served as the basis of the process and organizational patterns that can be used to guide future distributed development projects.

Coding the text-based transcriptions proved to be a complicated task, since it was difficult to keep track of the information and verify responses across projects. Many times an RA's response would extend beyond the simply stated questions and would instead interweave throughout the majority of their interview. Thus the researchers decided to create a database. Each project was entered as its own row and the database columns represented the necessary coding. Interviews were coded on such attributes as project meta-data, i.e. industry, number of stakeholders and sites, etc.; and the RA's description of their requirements engineering activities, such as communication and analysis. SQL queries were then used to test hypotheses and further identify patterns. The database design is included in Appendix B.

The projects included in this study cover a broad representation of industries, including telecommunications, pharmaceutical, video games, financial services and retail projects. Projects were identified through referrals from industry and academic contacts. Any referred project that represented a globally distributed requirements gathering process was included in the study. Projects took roughly 7 to 24 months from inception to implementation, ranged from 50 to 25,000 requirements, and involved the participation of stakeholders from two to several hundred

locations in the US, Asia and Europe. The requirements engineering role was assigned to project managers, lead interaction designers, business analysts, and project coordinators. Stakeholders primarily included end users, and a variety of SMEs such as directors, artists, system architects, product managers, trainers, developers, and warehouse leads. Each of the projects is summarized in Table 5.1.

There are limitations to this study. One limitation is the small sample size of the projects. However we mitigate this by not attempting to draw quantitative conclusions from our results. The in depth nature of the study enables useful observations despite the small sample size. To minimize influence of the interviewer, the questions were sent to the interviewees well in advance of the interview. To increase the chance that data was reported correctly we assured participants that their names and their company names would be stored separately from all gathered data and would not be included in any research publications. Furthermore all interviews were conducted when the interviewee was outside their work environment.

We recognize researcher bias could be considered a possible threat to validity, but Dagenais explains that in grounded theory researcher bias is a required attribute since the researcher is responsible for selecting the participants, refining the questions, and developing the theory [70].

Projects				People			Tools
ID No.	Domain	Type	Number of Requirements	RA Org Title	Stake-holders	Sites	
1	Financial Services	Reconfiguration for new industry	10 to 12	Technical Project Lead	7	2 N. America	Basic office
2	Telecommunications	Enhancement	400 to 600	Lead Interaction Designer	5	3 N. America	Basic office Project repository
3	Software Requirements Consulting	Customize OTS enterprise solution	5,000 to 9,000	Product Manager	40	2 N. America 1 Europe 1 Asia	Basic office Project repository Requirements management
4	Software Requirements Consulting	Implementation	5,000	Requirements Analyst	65	3 N. America 1 Asia	Basic office Project repository
5	Video Games	Ongoing Enhancement	136-340 per year	Project Manager	45	6 N. America 1 Europe	Basic office Project repository
6	Retail	Enhancement	40 to 50	Business Analyst	8	2 N. America	Basic office Project repository
7	Software Engineering Research	New solution	200 to 400	Researcher / Requirements Engineer	25	6 Europe	Basic office Project repository
8	Corporate Research	Enhancement	25,000	Consultant	37	3 N. America 1 Europe	Basic office Project repository Requirements management
9	Information Technology	Enhancement	50+	Product Manager	51	6 N. America	Basic office Project repository Requirements management
10	Software Solutions Integrators	New solution	30	Business Analyst	111	3 N. America	Basic office Project repository

Table 5.1. Research study statistics

5.2 Meta-model and Visual Notation

This section describes a new taxonomy and associated meta-model for representing distributed RE projects based on observations from this research study. The meta-model is depicted in Figure 5.1. Our findings indicate that there are three general types of entities: *roles*, *sites*, and *artifacts*; and three general types of relations: *houses*, *accesses*, and *communicates*, that occur between those entities.

5.2.1 Meta-model

5.2.1.1 Roles

A **role** represents a well-understood function that a person performs. The Rational Unified Process (RUP) defines it as a “hat” that can be worn by either an individual or a group during a project [81]. The RUP definition highlights two important characteristics of a role; first that it can be held by more than one person at the same time and second that a single person can have more than one role. In this study we identified a set of common stakeholder roles that recurred across many of the projects. These roles included the default role of SME, as well as a RA, Customer, Location Spokesperson (LSP), Developer, Manager, User, and Tester. These roles were assigned various titles in different projects, but were still clearly recognizable. The meta-model depicted in Figure 5.1 shows both the *role* entity and its attribute *subtype*, which can be set to any of these predefined role types. Furthermore additional role types can easily be added. The three roles of SME, RA, and LSP were observed across more projects than any of the other roles and are, therefore, described in greater detail.

- A **subject matter expert** (SME) is a stakeholder who provides knowledge about the product that is to be developed. SMEs come in all shapes and sizes and include various types of potential users as well as experts in legal constraints or specific topics such as security or usability. In this study a large variety of SMEs, are noted, i.e. users, managers, artists, designers, developers and trainers.
- A **requirements analyst** (RA), also known as a requirements engineer or business analyst is responsible for overseeing or supporting the requirements elicitation, analysis, and specification tasks. Several related titles were found in the studied projects such as *technical project lead* and *lead interaction designer*. In general, actual titles often reflected the nature of the project. In fact, only one project, which was conducted by an RE consulting firm, actually had a position called “requirements engineer”. In all other projects the requirements analyst was responsible for other tasks in addition to the managing the requirements.

- A **location spokesperson** (LSP) may hold some of the responsibilities of an RA, but is characterized by being situated at one location and is responsible for coordinating requirements gathering activities at that location and serving as a liaison to the project level RA. Again, we found many different job titles for this position including *technical lead* and *designated region representative*. In certain projects, the LSP also served as a foreign language translator between local stakeholders and the RA or other non-local project personnel.

This study also indicates that communication flows often occurred between groups of people holding the same role and that the size of the group holding a specific role was an important factor for distributed requirements engineering. As a result, we adopted the counting concept used by Amazon's Pirahã tribe [82] and categorized a role as *one*, *few*, or *many* individuals holding the same function at a given location. The *role* entity of our meta-model contains an attribute named *multiplicity*, which must be instantiated with one of these multiplicity values (*one*, *few*, or *many*).

5.2.1.2 Sites

A second important concept in distributed projects is that of *site*, defined as a place at which one or more project stakeholders are situated. A *site* could refer to a single building or a group of buildings in close proximity to one another. A site is characterized by the ability of its occupants to meet together frequently to engage in same language, real time conversations. This study suggested that from the perspective of distributed requirements engineering, a common communication language and time zone characterize a site and determine its communication flows. Additionally a site may be assigned single or multiple location values. Therefore, we have included these three properties as attributes in our meta-model. The meta-model also defines a *Site* entity, which composes a number of Roles, and Stationary artifacts, which are discussed in the following subsection.

5.2.1.3 Artifacts

An artifact is defined in the Unified Modeling Language (UML) documentation as the specification of a physical piece of information that is used or produced by a software development process, or by deployment and operation of a system [19, 20]. One of the primary goals of a distributed requirements gathering process is to collaborate with participating stakeholders to generate an agreed-upon specification. In fact the interviews showed that most interviewees highlighted the importance of the specification, which was generally constructed in the form of use cases or more formal requirements stored in a text document or spreadsheet, or was represented graphically as a process diagram, dataflow diagram, or as a graphical prototype. An artifact is also frequently characterized by its physical location.

While some artifacts reside permanently in one location i.e. on a shared drive, online library, or in a repository, others are moved from stakeholder to stakeholder across multiple locations, primarily via email. We found that this property of an artifact is important for analyzing the maturity of a distributed requirements process, and therefore the *artifact* entity in the meta-model is specialized into *Stationary* and *Travelling artifacts* entities. A *Stationary artifact* belongs to exactly one site and stays there while stakeholders assigned to specific roles work on it (see composition between *Stationary artifact* and *Site* in the meta-model). Conversely, a travelling artifact has no persistent site but is instead passed between stakeholders like a token.

5.2.1.4 Relationship between roles

The study also identified several common forms of communication and work relations that occurred between the different roles and artifacts. We named these *communicates co-located*, *communicates distributed*, and *accesses*.

A **communicates distributed** relationship connects two roles that are not geographically co-located and depicts that some kind of communication occurs between them. These case studies showed many different ways that stakeholders communicated within and across distributed sites. For example in one project SMEs in North America communicated via email to SMEs in Asia, in another project the requirements analyst in North America held telephone conferences with developers in Europe, while in other projects we found examples of many-SMEs-to-many-SMEs discussions. An analysis of these findings unearthed two different properties that characterized communication. The first was the medium used, i.e. telephone, email, or web conference, while the second was the multiplicity of the participating roles i.e. a 1:N relationship in which a single analyst at one site communicates directly with all SMEs at another site. Our meta-model shows the *Communicates distributed* relationship as an association relating roles while the communication medium is represented as a stereotype during modeling. The multiplicity of participating roles is captured through the multiplicity attribute of role as previously discussed.

As with the *Communicates distributed* relationship, a **Communicates co-located** relationship also connects two roles; however it represents the case that the associated roles are co-located and can communicate face-to-face. While this might be the default case within one *Site*, it implies travelling of at least one participating stakeholder in cases where the associated roles are distributed across multiple sites. Several study participants mentioned the situation in which they traveled during the requirements engineering process. For example one subject reported that an RA traveled to two different North American sites and a European site in order to interview SMEs.

An **Accesses** relationship associates roles with artifacts and means that stakeholders adopting that role contribute to the construction or maintenance of the associated artifact. We further distinguish accesses relationships into read (R), write (W), and read/write (RW) of documents and shared resources. Our meta-model shows the accesses relationship as an association between role and artifact entities, while the type of access (R, W, or RW) is modeled as a stereotype and not visible in Figure 5.1.

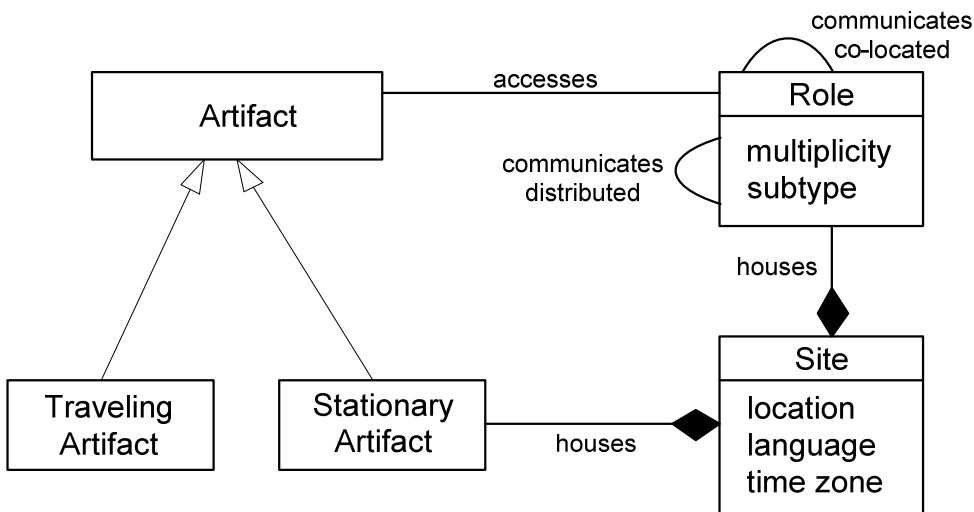


Figure 5.1. Meta-model Depicting Taxonomy for Distributed Requirements Engineering

5.2.2 VISUAL NOTATION

In order to enable project stakeholders to plan, assess, and improve their projects, we developed CGREN, a visual notation that represents all of the concepts defined in our taxonomy. In designing CGREN we followed Moody's nine principles for creating an effective visual software engineering language [83]. Refer to Chapter 3.3.4 of this thesis for a discussion of the visual notation icon selection process.

5.2.2.1 Basic Elements

The basic elements of CGREN are all depicted in Figures 5.2-5.3. Stakeholder roles are intuitively depicted as human shapes shown as one, few, or many stakeholders. Different stakeholder roles are depicted using specific adornments such as a pencil for the requirements analyst, a bullhorn for the location spokesperson, and a currency note for the customer. Sites are depicted as containers. Artifacts are depicted using a well-recognized database symbol to represent stationary work products, and using a selection of symbols to represent the most common types of traveling data types observed in our study. Finally, relationships are depicted intuitively using arcs. A solid line represents co-located communication between roles, a dashed line represents distributed communication between roles, and a dotted line represents the relationship between a role and an artifact. In addition relationship arcs are adorned by additional symbols representing various media of distributed communication, such as email or phone.

Moody explains that the basic elements of a notation should exhibit semiotic clarity, perceptual discriminability, semantic transparency, and visual expressiveness. Semiotic clarity requires all semantic constructs of a taxonomy to correspond 1:1 to the graphical symbols in a visual notation. To that end, our visual notation uses a different symbol to represent each of the taxonomy's elements and does not introduce any new concepts. Perceptual discriminability requires clearly distinguishable symbols for each different concept. Our visual notation achieves this through the use of color, shape, and size. For example, the three main concepts of stakeholder role, artifact, and site are clearly distinguishable by use of different shapes and sizes. Furthermore, relationships are visualized as arcs using different line types and are used to clearly distinguish between various relation types. Finally, semantic transparency is achieved when the visual appearance of a symbol immediately suggests its meaning. To achieve this, the symbols we chose represent either well-known concepts from other modeling languages i.e. lines to represent relations, or are intuitive symbols that clearly represent elements of the domain.

Visual expressiveness utilizes the full range and capacity of visual variables such as shape, color, and size to represent a rich variety of concepts. In our notation, dual encoding means that text is used to reinforce meaning and to add additional detail. Furthermore we allow practitioners to provide their own project-specific names for roles, artifacts, sites, and relationships. Additional symbols are used in our notation to avoid over-reliance on text. For example, multiplicity of stakeholder roles, (i.e. individuals, small groups, or crowds), specific types of stakeholders, and medium for distributed communications are all depicted using their own graphical symbols.

Finally, graphic economy, providing only a cognitively manageable number of different graphical elements for a notation is achieved by providing only the necessary set of symbols. Moreover initial discussions with practitioners showed that these symbols could easily be understood, distinguished, and memorized across meetings.

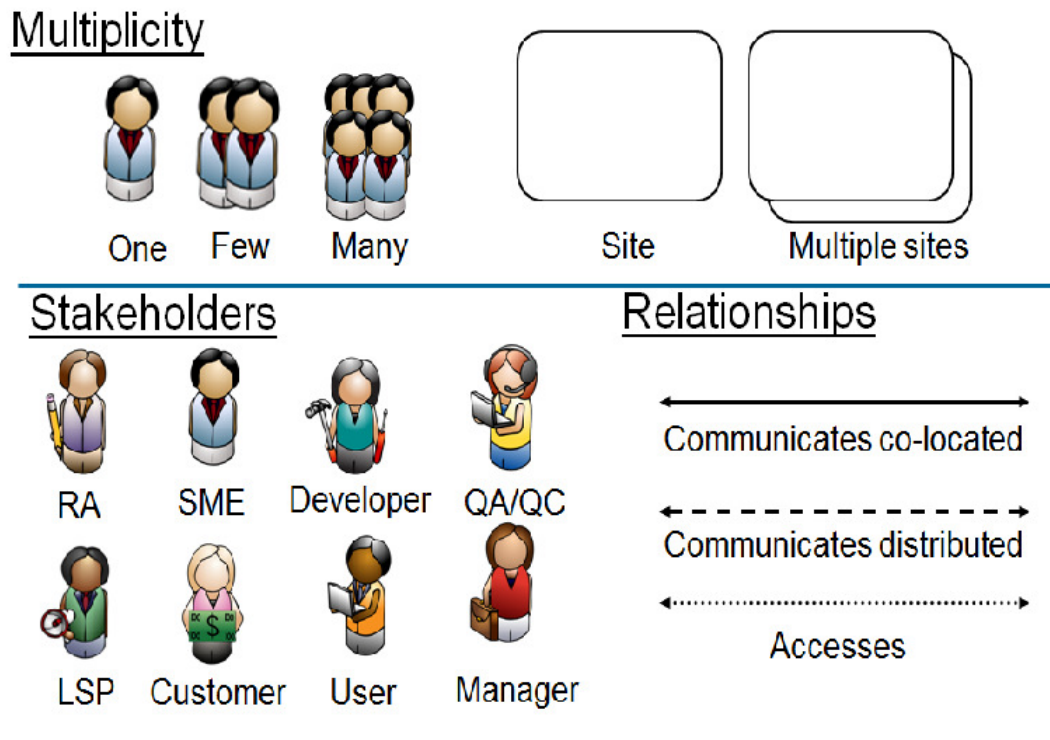
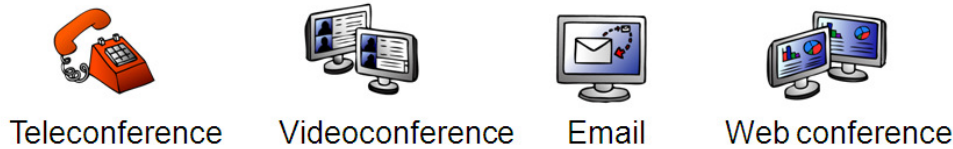


Figure 5.2 CGREN Icons – Sites, Stakeholders and Relationships

Distributed Communication Media



Stationary Artifacts

Traveling Artifacts



Figure 5.3 CGREN Icons – Communication Media & Artifacts

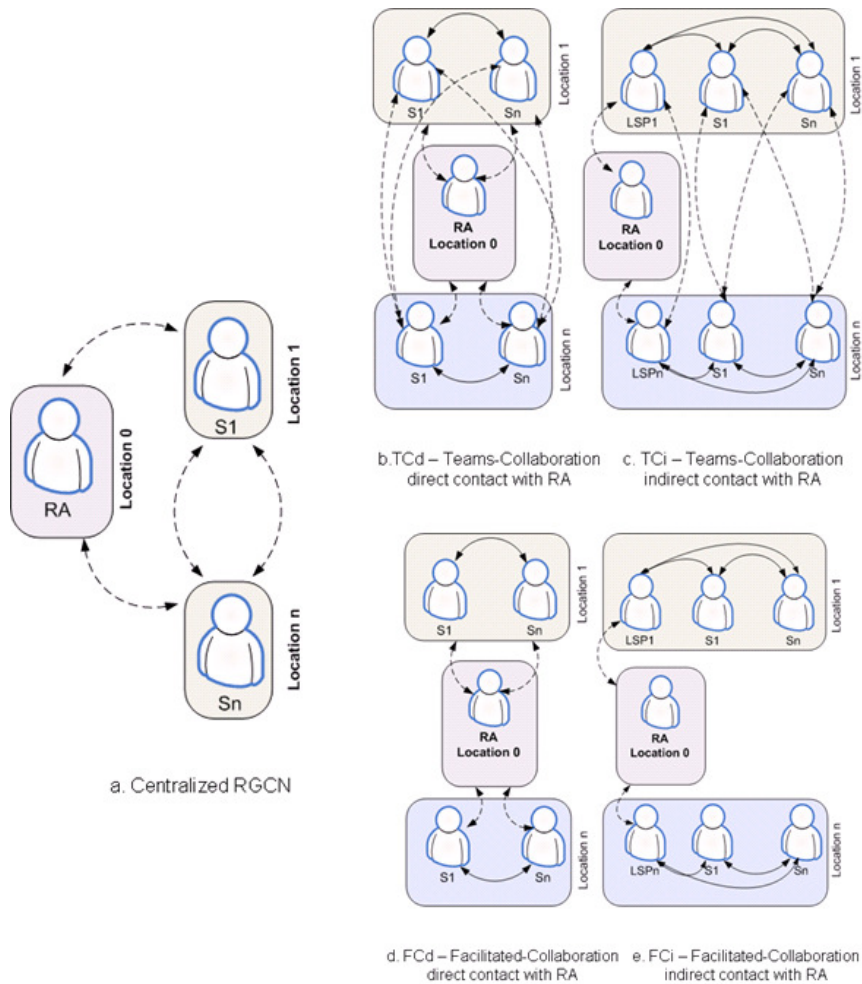
5.2.2.2 Complexity Management

As a project can grow in size and complexity, this new notation allows different concepts to be modeled at different levels of abstraction. For example, a general view of a project might show only groups of roles assigned to sites and associated via unspecified communication flows. A more focused view, on the other hand may depict single roles communicating via specific communication media while performing a certain activity within the requirements engineering process. Complexity is also supported through the notion of cognitive integration, which refers to explicit mechanisms that are used to integrate information from different diagrams. The various abstraction levels in this model integrate conceptually and perceptually within the general structure of the project, depicted by sites, roles, and general relationships described in a single, high-level diagram. All detailed models are derived from the project-level diagram and depict the same basic structure enriched with additional information.

5.2.2.3 Examples

As part of each case study, the social structures of the project were examined and modeled through instantiating the general RGCNs as shown in Figure 5.4. The Centralized RGCN model, Figure 5.4a, represents the basic communication flow between the RA and the stakeholders (S1, S2...Sn). Figures 5.4b-c depict fully collaborative RGCNs, in which both formal and informal communication occurs between all types of stakeholders at all sites. However, an organization might be more likely to implement one of the distributed collaboration RGCN models, shown in Figures 5.4d-e, if cross-communication between distributed locations is not supported. Quite possibly the fully collaborative RGCN models may

be “activated” for kick-off and status meetings; while the distributed collaboration RGCN models are the business-as-usual process for large-scaled projects. The distributed collaboration RGCNs facilitate a divide and conquer approach, whereby, different sites can be assigned different topics and functionality to work on. Understandably a given project might encompass some hybrid version of these different models. A complementary set of RGCNs that depict an RA being co-located with at least one of the stakeholders has also been created and is shown in figures 5.4f-i.



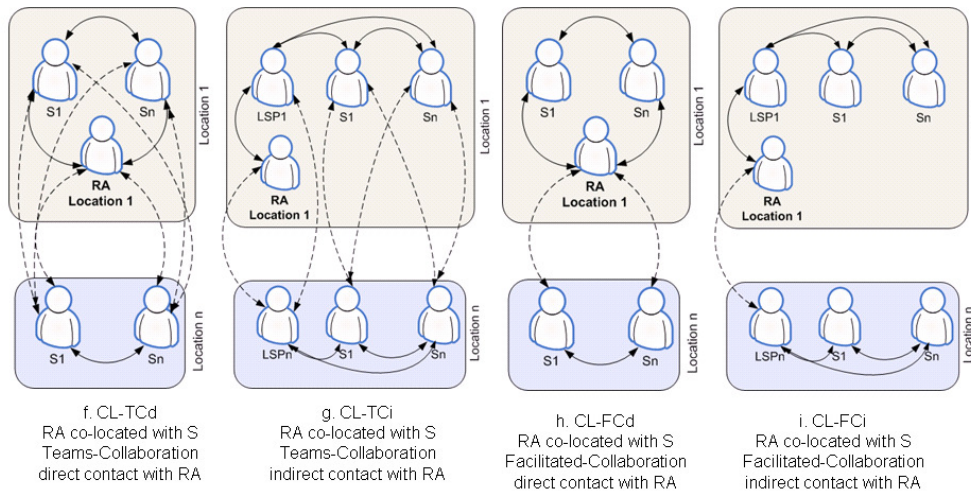


Figure 5.4 General structures of Requirements Gathering Collaborative Networks

The symbols described in the previous section can be combined in various different ways to model different components of a project. Below are several different examples that illustrate how the symbols can be composed to represent higher level concepts.

Figure 5.5 composes the concepts of site, role, and communication, and represents one requirements analyst communicating with a few subject matter experts. In this example all of the stakeholders are co-located at a single Asian location.

Similarly, Figure 5.6 illustrates a requirements analyst from a USA location communicating via telephone conference and collaborating on a requirements specification document, with a spokesperson in a Pakistani location. In this example the location spokesperson also communicates and works with a few co-located subject matter experts. Furthermore, in this scenario the RA and LSP both access a shared traveling document which is periodically passed between sites.

Finally, Figure 5.7 illustrates how to model stakeholders and their access rights to a stationary project artifact. In this example the requirements analyst from a German location, and SMEs in the USA all have read/write access to a shared repository. The Estonian SMEs can access the repository to review information, but cannot perform updates.

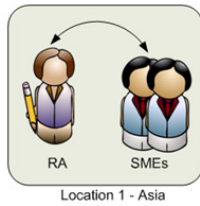


Figure 5.5 Modeling co-located communication

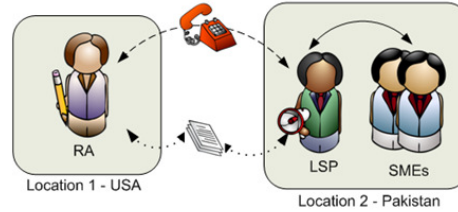


Figure 5.6 Modeling distributed communication and collaboration with a location spokesperson at remote site

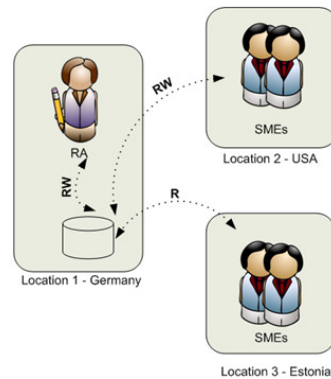


Figure 5.7 Modeling access to a shared artifact across multiple sites

5.3 Research Findings

In this section we use RGCNs and patterns to report on the processes, social organization, collaborations, and tool support that were used in 9 different requirements engineering domains.

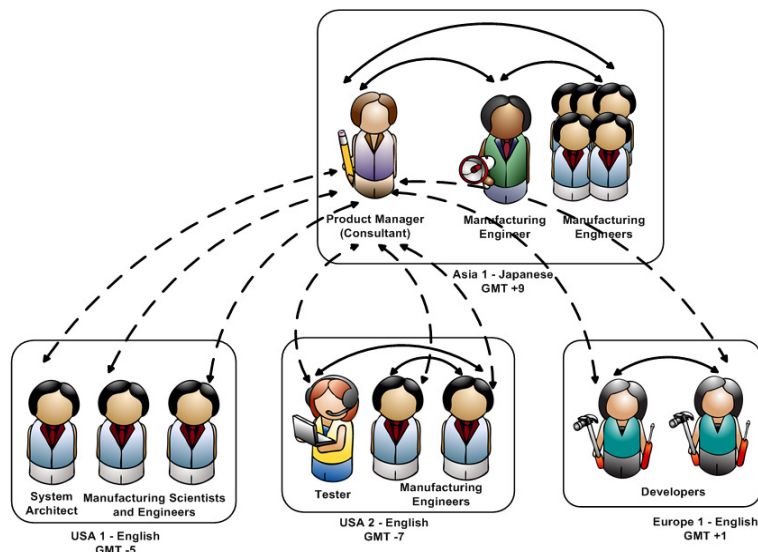


Figure 5.8 Communication Flow between Project 3's RA and Stakeholders

The *communication flow* can be regarded as a generic template for the project. Throughout the requirements development phase of a project, several flows of communication occur. Typically the requirements gather interacts directly with all of the stakeholders, as shown in Figure 5.8. But depending on the number of stakeholders located at a remote site, noted language barriers, or needed subject matter expertise; the RA may communicate to a group of stakeholders via an LSP. In all of our projects the LSP was more than a foreign-language-to-English translator; the LSP role was assigned to a current project stakeholder. Surprisingly, stakeholder-to-stakeholder communication occurs during most, but not all, projects. Distributed locations may work on specific project functionality or tasks may be assigned in a divide-and-conquer fashion. The communication flow model depicts the sites of the project stakeholders, who the primary stakeholders are, and the paths of communication across and within each location. We have designed four other models, built using the basic *communication flow* model as a base template, to illustrate specific requirements engineering tasks. In each of these models the communication paths are labeled according to the collaboration method and tools that are adopted to support the task. Each individual project in the study was represented using a set of RGCNs.

The following sections of Chapter 5.3 detail the Distributed RE Patterns that were discovered during this research:

- Requirements Analyst Title
- Telephone – RA’s Preferred Communication Tool
- Location Spokesperson Role
- Requirements Engineering Tools and Technologies
- Face to Face Communication
- Requirements Specifications Shared only via Email
- Distributed Stakeholder Communications
- Multipurpose Requirements Specification
- Travel Between Sites Unnecessary for Small Groups

Refer to Appendix B for the RE Interview database queries.

5.3.1 Pattern: Requirements Analyst Title

Pattern: Requirements engineering responsibilities are assigned to project manager, business analyst and other professional roles that participate in industry software engineering projects.

Observations:

In the software/requirements engineering domain the person responsible for conducting the requirements elicitation, analysis, specification and management activities is referred to as the Requirements Engineer. During our study, only two organizations had people with this title or the closely related “Requirements Analyst” title; both were members of requirements engineering consulting companies. Refer to table 5.2. This was expected since early feedback from professional contacts agreed with our request for participation message which stated that were looking for ‘business analysts, software engineering and industry professionals who are or were responsible for gathering requirements from stakeholders and users at multiple distributed locations.’ Most of these RAs were not requirements engineers; this could explain why frequently they were not familiar with requirements engineering tools and methodologies.

None of the interviewed RAs had requirements-related activities as their sole responsibility. In addition to gathering and managing the project requirements, the RAs were also responsible for project management, product design and management, and requirements management-related training.

Project 3’s RA described her duties as *“Gather requirements, sub-prioritize requirements, create requirements documents, ensure they were understood by the stakeholders and developers, and to perform traceability and requirements management...Total life-cycle project management. From the beginning of the IT project, sometimes through to testing, but at least up to the point of development.”*

For Project 6, the RA shared that he had to *“Take the high level requirements from the business and rework them to be mutually acceptable to both the business sign off and the IT delivery manager sign off; and that the high level requirements that would be signed off would also if applicable, have the scoping of the high level features... <The stakeholders put some type of requirements list> together on their own. And then that was the starting point for the official high level requirements, often times clarifications were needed.”* This was where the RA became involved. *“Requirements that weren’t very specific had to be massaged as far as the IT delivery team was concerned... <business analysts> BAs are usually the ones to handle the requirements, but sometimes User Experience people get involved before the business analyst so they have some other exposure to what the internal client is looking for, but they’re getting a different format I should say”,* the RA continued.

“As the product manager my responsibility is to collect, understand, and synthesize, requirements from specific customer so that we get a view of the markup requirements as a whole. So we’re trying to make sure that we’re getting a view of what customers want in general rather than what any one specific customer wants. And then, interpreting those business requirements of the customer-users for the engineering organization; so articulating them in terms of feature and function. And working with the engineering organization to understand how those can be best implemented, and to prioritize and cost them, and to break them down into a set of those features which deliver value to the market”, is how Project 9’s RA described her responsibilities.

Project 10 RA’s responsibility was *“technical liaison between the client and management and the client and any of our company, because we had a third-party company that we were working with. They were the actual solution developers. So a liaison between the client and my company, between the technical staff and my company.”*

One of the major challenges we observed was that RAs were not aware of the web-based collaboration tools that are available. Communication and collaboration tool usage is mostly based on the requirements analysts’ knowledge and experience of the current tools they are using. As one RA, a self-proclaimed Web-X snob noted, *“LiveMeeting is harder to get around and share. I was brought up on WebX. I know WebX very well. It’s a comfort thing.”*

Project No.	Industry	RA’s Organization Title
1	Financial Services	Technical Project Lead
2	Telecommunications	Lead Interaction Designer
3	Software Requirements Consulting	Product Manager
4	Software Requirements Consulting	Requirements Analyst
5	Video Games	Project Manager
6	Retail	Business Analyst
7	Software Engineering Research	Researcher / Requirements Engineer
8	Corporate Research	Consultant
9	Information Technology	Product Manager
10	Securities Software Solutions Integrators	Business Analyst

Table 5.2 RAs’ Organizational Titles

Supporting Data (queries from RE Interview database)

RA Query 1 used to produce Table 5.2

SELECT

REProjInfo.ProjNum,

REProjInfo.Industry,

REProjInfo.RATitle

FROM REProjInfo;

Pattern-RA-Title		
ProjNum	Industry	RATitle
1	Financial Services	Technical Project Lead
2	Telecommunications	Lead Interaction Designer
3	Software Requirements Consulting	Product Manager
4	Software Requirements Consulting	Requirements Analyst
5	Video Games	Project Manager
6	Retail	Business Analyst
7	Software Engineering Research	Researcher / Requirements Engineer
8	Corporate Research	Consultant
9	Information Technology	Product Manager
10	Securities Software Solutions Integrators	Business Analyst

RA Query 2 database information

SELECT

REProjInfo.ProjNum, REProjInfo.Industry,

REProjInfo.RATitle, REInterviews.RADuties

FROM REProjInfo INNER JOIN REInterviews ON REProjInfo.ProjNum = REInterviews.ProjNum

ORDER BY REProjInfo.ProjNum;

ProjNum	Industry	RATitle	RADuties
1	Financial Services	Technical Project Lead	Technical Project Lead.
2	Telecommunications	Lead Interaction Designer	Lead designer, interaction designer on that project.
3	Software Requirements Consulting	Product Manager	Gather requirements, sub-prioritize requirements, create requirements documents that they were understood by the stakeholders and developers, and to perform traceability.
4	Software Requirements Consulting	Requirements Analyst	Worked throughout of the project as a requirements analyst. We used to be called requirements engineers, and I also managed the team of other requirements engineers.
5	Video Games	Project Manager	Project Manager, Cheerleader, Mediator, Software Architecture/final decision maker. I oversaw 40 people. 3-4 people on my team interacted with 3-30 people on the client's team.
6	Retail	Business Analyst	Take the high level requirements from the business and rework them to be mutually acceptable to both the business sign off and the IT delivery manager sign off; and that the high level requirements that would be signed off would also if applicable, have the scoping of the high level features.
7	Software Engineering Research	Researcher / Requirements Engineer	Basically the project had already started when we were contacted so we'd do requirements engineering because the project partners that were supposed to do that they didn't.
8	Corporate Research	Consultant	I was brought in to help define a requirements process, to set up the process, tooling, the requirements tooling, train the staff, and also train some of the client's staff.
9	Information Technology	Product Manager	As the product manager my responsibility is to collect, understand, and synthesize requirements from specific customer so that we get a view of the markup requirements.
10	Securities Software Solutions Integrators	Business Analyst	My responsibilities would be technical liaison between the client and manage client and any of our company, because we had a third-party company that we

Existing Literature:

Similar to other industry researchers we found as Paech notes, “In my industrial requirements projects, rarely is there a role called *requirements engineer*. Typically there are customers, developers...and project managers; sometimes there are product managers. All of them do some requirements engineering, but often implicitly [84].” Project’s 5 RA, who had the title of Project Manager, described his responsibilities in the following manner, “I take the high level requirements from the business and rework them to be mutually acceptable to both the business sign off and the IT delivery manager sign off; and that the high level requirements that would be signed off would also if applicable, have the scoping of the high level features. If multiple iterations were indicated and then from that point it that would be the departure for them going deeper and doing the requirements for the different iterations.”

Sangwan et al, discuss the roles that are necessary for a global development team, and included on list are requirements engineering and project management roles [85]. The requirements engineering responsibilities align with the duties that our RAs performed; as does the project and product management responsibilities described by some of our RAs. For example Project 9’s RA / product manager shared “As

the product manager my responsibility is to collect, understand, and synthesize, requirements from specific customer so that we get a view of the markup requirements as a whole... then interpreting those business requirements of the customer/user for the engineering organization; articulating them in terms of feature and function. And working with the engineering organization to understand how those can be best implemented, and to prioritize and cost them, and to break them down into a set of those features which deliver value to the market.” For bigger projects Sangwan et al recommend that these 2 sets of responsibilities be assigned to different individuals.

Cacklenord asserts that the BA role arose from tasks that were previously performed by PMs, specifically eliciting, gathering and analyzing high-level business requirements. She continues that a BA may be known by a variety of titles in various organizations, i.e. systems analyst or engineer, or requirements engineer or project manager, etc. Included in her description of skills that a great business analyst must possess are a “tool kit of techniques to elicit, analyze and present excellent requirements [86]”; and the ability to assist the PM in scoping new projects.

Beranbach, et al also separate the project manager and *chief* requirements engineer roles; depicting them as jointly responsible for the project outcome; and include the project manager and chief requirements engineer as two of the most critical roles for managing distributed requirements engineering development projects [87].

Recommendations:

Researchers and industry professionals agree that there is no clear definition of what constitutes a requirements engineer, as opposed to a business analyst or project lead. Paech remarked that, “Companies trying to establish clear requirements engineering responsibilities don’t have clear standards on how to train their people, define the role, or choose the right people for the job [84].” To address this situation she has launched a professional certification program. Her goal, like ours, is not academic education, but instead to improve the situation in industry.

Many IT project managers and business analyst-from both the business and IT units, receive some type of formal or informal requirements engineering training, especially as pertains to the recommended processes for eliciting, gathering, analyzing and documenting requirements. Often though, they are not provided with guidance regarding RE tools evaluation and selection.

A possible solution, if an RE consultant is brought on board for a new project, would be to allot funding and time for cross-training, at the very least, to the person (within organization) who has been assigned the requirements engineering duties. If for some reason this training is not feasible, the RE consultant should be required to provide documentation about the processes and tools they recommend as best practices. Organizations should strive to include RE professionals as part of their PMO, and IT departments should do likewise for the project teams. Not only can the requirements engineers help contribute to the success of the company's software engineering projects; they can also help with training their teammates and sharing their knowledge of best practices and lessons learned.

Researchers can also expand the scope of their projects and educational outreach to include industry professional with requirements engineering roles and responsibilities; not just instead individuals with a "requirements engineer" title.

Open Questions:

1. Are the RA and her organization aware of the requirements engineer role as its own discipline within software engineering?
2. Are there requirements engineers within the RA's organization or department?
3. Did the RA receive any RE-specific training prior to performing in this role?

5.3.2 Pattern: Telephone – RA’s Preferred Communication Tool

Pattern: Telephone conferences are the appropriate communication tool for distributed requirements engineering activities, even though newer technologies are available.

Observations:

In all of the organizations that participated in our study, distributed stakeholders mostly communicated with each other via telephone conferences and collaborated by exchanging requirements documentation via email. All of the requirements analysts (RAs) utilized telephone conferences to some degree in their projects. Table 5.3 lists the possible number of telephone communication pathways. Face-to-face communication was the preferred method of communication, with some organizations allowing the RA and/or project stakeholders to travel between locations.

When the stakeholders are distributed across multiple locations, by necessity, informal communications also occur via telephone conferences and email exchanges. This is in contrast to the impromptu face-to-face communication opportunities, like conversations in the break room or at the coffee machine that take place when stakeholders are co-located.

Our research interviews revealed that requirements analysts prefer telephone conferences to video conferences for eliciting and gathering requirements due to technology issues that occur during video conferencing, such as a bad connection and equipment failures that can derail the meeting.

Though Project 3’s RA preferred meeting with stakeholders face-to-face, and when that hadn’t been possible on other projects, they used GoTo Meeting, *“that works pretty well because you can actually put the documents in front of people and have them look at it while you’re looking at it. I’ve used video conferencing, and that doesn’t really help a lot. It’s nice because you can see how people are reacting but it’s still not as good as face to face. In lieu of face to face I would say webinar-type teleconferencing is the best because with video conferencing there are technical difficulties that can pop up that make it a little more difficult to use.”* A webinar was this RA’s second choice for interacting with stakeholders. When meeting in person was not an option and webinar capabilities were not available, the RA organized phone conferences.

Project 3’s RA continued, *“There are a couple of things you miss when you’re not face to face, one is the ability to use the walls, use the whiteboard, and use the actual physical space to communicate. And I know that there are some whiteboard applications out there and I haven’t used them, but I think that if*

that can be incorporated into the webinar technology that would be big. And the other thing of course in not being able to actually see people. More robust video conferencing may be able to help that. But I think the biggest thing is being able to use the physical space in the room to get people to participate.” The RA’s description of “more robust video conferencing”, centers on the issues that the RA has encountered in the past, including connection problems and poor video quality.

Project 4’s RA shared, *“My first choice is always face to face mainly because there are many nuances that I lose even over the phone. I have been lucky enough to see video conferencing equipment working where I am right now, and that’s my first experience with that. It’s a lot better than not having it, but it is so technology dependent that if for some reason the video is not cooperating, it can very easily derail a session. So I try to keep these things as simple as possible where if I can meet with the person face-to-face I do that, if not I just do it over the phone.”*

When the RA for Project 6 was asked if video-conferencing was ever used, the RA responded, *“Maybe once or twice, not so much. It was so much harder to really get the focus on the artifacts we were looking at through video conferencing. (For example, the ability to view a diagram as opposed to seeing other meeting participants); the white board’s capabilities didn’t really seem to transmit very well through that channel...didn’t have clarity.”*

Though video conferences allowed the stakeholders to see each other; *“it is absolutely the worst thing for conflict”* according to Project 5’s RA. *“You would be in a <conference> room and the video screen would come up and every person who was <previously> complaining and said they were going to <argue with the other group>, would shut up; once they saw their own picture on the screen. Because it was a multiplex, <multiple screens> when you stare at yourself, you’re not going to be as big of a jerk. It was bad because it actually deflected the whole issue... instead of a small conflict rising up, which happened especially with <a European site>; they avoided conflict at all cost... So what could have been a 5-minute uncomfortable-ness became a territorial bashing back and forth...”*

Project 5’s RA continued *“... it’s all about confrontation and you can see yourself confronting, because you can see yourself in the little window. That stopped it. People didn’t do it. I don’t know what the phenomenon is.”* Video conferencing was company mandated at the RA’s organization; and eventually the RA found a way to disable the video feed so that conflicts could be discussed and addressed during project meetings.

“Not during this project <but> in previous releases we have done video conferences internally with the development team... I don’t think it adds much value beyond a phone conference”, said Project 9’s RA.

Project No.	Industry	Locations and Roles
1	Financial Services	RA, 2 sites, 6 stakeholders
2	Telecommunications	RA, 3 sites, 5 stakeholders
3	Software Requirements Consulting	RA, 4 sites, 39 stakeholders
4	Software Requirements Consulting	RA, 4 sites, 66 stakeholders
5	Video Games	RA, 8 sites, 45 stakeholders
6	Retail	RA, 2 sites, 8 stakeholders
7	Software Engineering Research	RA, 5 sites, 25 stakeholders
8	Corporate Research	RA, 3 sites, 37 stakeholders
9	Information Technology	RA, 6 sites, 51 stakeholders
10	Securities Software Solutions Integrators	RA, 3 sites, 110 stakeholders

Table 5.3 Possible Telephone Pathways for RA

Existing Literature:

Briggs, et al groups collaboration tools that are currently available into multiple categories, for instance, conversation, document sharing, voting, streaming and information access [88]. Audio and video conferencing technologies are considered streaming tools.

A telephone conference is an example of audio-conferencing which enables all of the stakeholders to hear the exact same information at the same time. Audio conferences are best suited for use when real-time communication is needed for discussion and clarification.

When stakeholders want to see each other in addition to hearing, they often try videoconferencing. During a videoconference stakeholders located at dispersed sites can see and hear the other meeting participants. This type of technology essentially attempts to re-create face-to-face communications.

Other researchers also have noted that distributed teams prefer viewing the same artifacts to viewing each other [89, 90]. In Ter Bush’s research on virtual teams she noted that a possible reason that distributed

team members preferred seeing shared information could be that the quality of the video conferencing feed was not yet at the level where team members could clearly read each other's facial expressions [89]. Mittleman et al noted that team members found focusing on the object of discussion, i.e. requirements document, use case, flow diagram, etc., provided information that was more pertinent to the project, than just watching a teammate talk [90].

Recommendations:

A telephone conference is the appropriate and preferred tool for many of the requirements engineering activities. A video conference may prove detrimental to these activities by hindering the communication and collaboration processes. However if an RA wants to supplement the appropriate activities by including video conferences, she should consider the technical infrastructure that is needed to support this process, especially at remote locations.

5.3.3 Pattern: Location Spokesperson Role

Pattern: A Location Spokesperson is a more than a translator. The stakeholder in this role is a site-specific subject matter expert who can assist with requirements engineering activities at their distributed site.

Observations:

As described earlier a location spokesperson (LSP), is a project stakeholder who is located at a site and is responsible for site-specific requirements gathering activities and interacting with the Requirements Analyst (RA), on behalf of his co-located teammates. It is important that of all the stakeholders, the person assigned the LSP role be a CRACK stakeholder [91]; that is someone who is Collaborative, Representative, Authorized, Committed, and Knowledgeable. As with the RA-Title pattern, the LSP role can be associated with many organizational management-related titles, i.e. region representative, technical lead; as well as an SME such as engineer or developer. For several projects, the LSP also served as a language translator between the RA and location stakeholders.

The LSP role was most essential when the RA was not co-located with a group of stakeholders; when a language barrier existed; or when having a sizable group of stakeholders participate in a meeting would result in an unproductive meeting. Refer to Table 5.4. In these research projects an LSP could be a member of a remote team even when there were only a few stakeholders at the distributed location. For example, for Project 1 since the RA and stakeholders were not co-located; the RA spoke mostly to the technical lead. The technical lead then communicated with the other 2 stakeholders at their site.

Project 5's RA shared that including all 20 stakeholders from a site as meeting participants was just inefficient and could cause a meeting to go on for as long as 2 hours. He stated, *"When these projects would start off, everybody wants to participate. So you get over-representation...you initially start off and there are a lot of people in the room. Even though there might be 20 users using the tool we had to reduce it down to a representative of 1-2 people who would participate in these meetings. The people who were there had 1 or 2 roles, the de-facto lead for that user group, where they could make the call... Everybody in the room had to be able to make a binding decision otherwise it didn't work...Or they were the expert who understood the problem. They were the resource who could give more elaborate data and detail about the material...We always tried to have representatives, people who were liaisons."*

Though the RA for Project 3 was co-located with stakeholders at the Asian site, an LSP was still needed due to language differences between the RA and stakeholders. The typical communication flow was RA-

to-stakeholder, stakeholder-to-their boss, LSP or stakeholder reply to RA. Not only did a stakeholder's ability to speak English play a role in the communication flow, it effected the stakeholder's position as well. Project 3's RA explained, *"Sometimes <a stakeholder> would bring people along who would also have an interest in their particular functional area and if they didn't know the answer to a question I asked they would confer amongst themselves, and a lot of those people had a very limited degree of English capability. So they would talk amongst themselves and the spokesperson would speak to me. In these situations it was pretty important that the spokesperson had subject matter expertise, as the concepts and the terminology was very focused and technical. So they would have to confer and have to understand what they were saying, to talk to me directly."*

Project No.	Industry	Locations and Roles
1	Financial Services	USA0 – RA USA1 – 1 LSP + 2 Stakeholders
3	Software Requirements Consulting	Asia1 – RA + 1 LSP + 29 Stakeholders Europe1 – 2 Stakeholders USA1 – 4 Stakeholders USA2 – 3 Stakeholders
4	Software Requirements Consulting	USA0 – Project RA USA1 – 15 Stakeholders USA2 – 20 Stakeholders Asia1 – Site RA/LSP + 30 Stakeholders
5	Video Games	USA1 – RA USA1A – 3 LSPs + 8 Stakeholders USA1B – 2 LSPs + 4 Stakeholders USA 2 – 2 LSPs + 5 Stakeholders USA 3 – 2 LSPs + 3 Stakeholders USA1C – 2 LSPs + 3 Stakeholders USA 4 – 2 LSPs + 3 Stakeholders Europe1 – 2 LSP + 4 Stakeholders
6	Retail	USA 1 – RA + 2 Stakeholders USA2 – 1 LSP + 5 Stakeholders
8	Corporate Research	USA0 – RA

		USA1A – 3 Stakeholders USA1B – 2 LSPs + 23 Stakeholders Europe1 – 1 LSP + 4 Stakeholders
9	Information Technology	NorthAmerica0 – RA USA1 – 2 LSPs + 5 Stakeholders USA2 – 2 Stakeholders USA3 – 8 LSPs + 8 Stakeholders USA4 – 7 LSPs + 6 Stakeholders USA5 – 6 LSPs + 7 Stakeholders
10	Securities Software Solutions Integrators	USA0 – RA + 3 Stakeholders USA1 – 3 LSPs + 100 Stakeholders USA2 – 4 Stakeholders

Table 5.4 Projects with Location Spokes Person role

Existing Literature:

The LSP role is synonymous with the roles of Broker and Gatekeeper [92-94] described in software development research. As discovered in our research too, a broker / LSP is usually an SME who controls the flow of information between distributed groups [95, 96]. In their book of organizational patterns Coplien and Harrison describe a Gatekeeper role as the stakeholder who helps to facilitate the information flow both within and outside of the development team [92]. This is similar to how the LSP serves as a facilitator between the RA and remote stakeholders.

Wolf *et al* describe the role of communication broker as a project team member who is the communication link when other members of the project cannot communicate directly with each other; either because of a language barrier or they are not co-located [94]. In requirements driven collaboration projects one of the stakeholder roles is that of a broker, who acts as a bridge between two groups of stakeholders [93].

Recommendations:

While the RA is the broker of all of the RE RGCNs, the LSP is the broker of the communication and requirements elicitation and gathering networks at their specific site. As such the LSP must receive some level of training to prepare them for this critical role. It is also necessary that the RA and LSP maintain open lines of communication between themselves.

Even for small groups of remote stakeholders, assume that an LSP is needed. In these research projects a specialist was usually assigned this role because of their experience and knowledge. An LSP can be useful as a site's requirements gatherer since in all likelihood a greater level of trust exists between the LSP and their co-located teammates than between the remote stakeholders and the RA.

A team member must be carefully chosen for the LSP role because of their power to control the information flow between various project stakeholders. There is also a higher potential for the RA to receive second-hand, filtered information. Furthermore an LSP can introduce misunderstandings and confusion in the transmission of the requirements information [93, 95]; or worse abuse this position as a way to push his or her own project agenda. For the most part these situations were mitigated in these research projects by the RA's access to the individual stakeholders; the RA maintained the option to communicate and collaborate with them.

5.3.4 Pattern: Requirements Engineering Tools and Technologies

Pattern: Requirements Analysts will rely on software packages and project management tools, i.e. word processing, spreadsheets, etc., that they are most familiar with.

Observations:

One of the major issues to emerge from this research was the inadequacy of tools that practitioners had selected to support distributed requirements engineering processes. Refer to Table 5.5. The requirements analysts (RAs) were often unaware of available tools or did not consider their needs for tool support until late in the project when it was logistically difficult to get them set up. As a result, activities that involved collaboration across geographical boundaries were often hampered by communication and version control problems, and in some instances relied upon extensive and frequent travel between sites.

None of the projects we studied used groupware tools to facilitate web-based real-time meetings. One project did set up a project-specific wiki, while several utilized web-based document sharing tools and used shared drive repositories to facilitate the exchange of information. Our study also showed that distributed stakeholders frequently communicated via telephone conferencing, as described in the Phone Conference Pattern, and email.

Our study revealed four primary challenges and problems that thwarted the adoption of collaboration tools in the requirements engineering process.

Requirements Engineering tools provided little support for collaboration: Although there are many different distributed requirements gathering online technologies available, such as EGRET, eConference, Rally, and intranet-based versions of commercially available requirements management tools, such as CRADLETM and DOORSTM, we found relatively little evidence of their use in the projects we studied. Our observation of these tools showed that in several cases, even though the tools were advertised as providing a collaborative environment; they actually only provided an interface that allows multiple user access for maintaining the requirements database. Effective project tooling would therefore require a fairly complex combination of requirements management and groupware tools.

Lack of a proactive tool adoption plan: In most of the studies tool adoption was relatively ad-hoc, and often reactive to problems that occurred after the start of the project. Furthermore, one of our research participants, a DOORS user, mentioned that “*the overhead of introducing such tools mid-project made them infeasible in certain situations*”. Another participant said “*At one point we started using a requirements management tool, CaliberRMTM, but it was so far into the project that it would have*

required significant work to move everything over to the tool so we dropped it.” The participant further stated that “If we had started at the very beginning of the project it probably would have worked out great. In fact, we’re using it in other projects right now, and we found that that tends to work better if you start out right at the beginning of the project.” These comments suggest that one of the adoption barriers to the use of tools was related to a lack of planning in the upfront stages of a distributed requirements engineering project.

Lack of tool-related knowledge: The third major adoption barrier we observed was simply that RAs were not aware of the web-based collaboration tools available to them. Tool selection was primarily based on the requirements gathers’ knowledge and experience of the current tools they are using. As one RA noted that even though “LiveMeeting is harder to get around and share. I was brought up on WebEx. I know WebEx very well. It’s a comfort thing.”

Technology issues: Our study also revealed that technology issues created a major adoption barrier. Several interviewees mentioned that they preferred telephone conferences to video conferences due to technology issues such as bad connections and equipment failures that occurred frequently and could derail the meeting. It is very clear that project participants were not willing to adopt higher level tools, if those tools created technical problems that got in the way of accomplishing their tasks rather than supporting them.

Tools and Technologies	Project Number									
	1	2	3	4	5	6	7	8	9	10
E-Mail	X	X				X				X
Instant Messaging									X	
Shared repository / drive / Intra-net / proprietary document management system	X	X					X	X		
SharePoint			X	X	X				X	
Telephone conference / meeting	X	X	X	X	X	X	X	X	X	X
Video / Web Conference		X	X		X	X			X	
Whiteboard						X	X		X	
Wiki		X			X	X	X			
Office										
Database							X			X

Graphics / Diagrams		X	X	X	X	X				X
Spreadsheet		X	X						X	X
Word Processing	X	X	X	X	X	X	X	X	X	X
Requirements Management										
CaliberRM			X							
DOORS								X	X	
Software Bug Tracking										
DevTrack Pro					X					
Software Development Platform										
Rational Team Concert (RTC)									X	

The video and web conferencing tools discussed during this research include AdobeConnect, GotoMeeting, NetMeeting, Sametime Unyte, and WebEx.

Table 5.5 Tools and Technologies used by RAs

Existing Literature:

Despite the current state of practice depicted in our study, groupware tools are available that facilitate the collaboration of large groups of people across time and distance as they engage in various distributed requirements engineering tasks such as brainstorming, categorizing and prioritizing ideas and requirements [97].

There are hundreds of groupware products available and more are continually introduced. Vendors of commercially available tools attempt to create “customized” packages by bundling core technologies together. For example, Mittleman et al noted that the most recent version of Yahoo Messenger includes the following basic technologies: instant messaging, audio conferencing and file exchange. Although this range of options may make it difficult for practitioners to understand and evaluate what functionality they need and to select the appropriate combination of tools that will handle their tasks [98], these tools can effectively help practitioners address some of the challenges of distributed requirements gathering and management.

Briggs, et al groups the available collaboration tools into multiple categories [88], five of which directly apply to the requirements engineering process: conversation, document sharing, voting, synchronous meeting and information access.

Conversation Tools: The tools in the conversation group include email, instant messaging, chat room and online forums. Email is essentially the electronic version of composing, storing, sending and receiving messages. This is an ideal technology to use when communicating with an individual or groups of stakeholders simultaneously. A major limitation of using email is that the sender cannot control when their message will be read, and henceforth does not know when they will receive a response.

Our study found many instances of using email. For example, it was used effectively for mass distribution of project artifacts such as documents and links to files. One of our RAs expressed that email was the only tool they needed since theirs was a relatively small project, requiring modifications to a current application. We also discovered several projects in which email was basically the only tool used to manage all of the collaboration needs. This led to various problems. For example, several of the interviewed RAs collaborated on documentation with stakeholders via email. One RA noted that all of the email messages and attachment cluttered their email account and stated that she “*would prefer (a) web-based interface to track things.*” This RA found it difficult to manage the requirements development process since there was not a way to automatically separate the project-related emails from their other emails.

We noted several instances of email folders being used to manage requirements. A major disadvantage to this process is that special attention is then needed to manage document versioning and to keep track of who has the current version, and the location of most updated documents. When describing their process for sharing use cases with stakeholders, an RG noted, “*I had to email them since we didn’t have a Wiki, and then our business users didn’t have access to ClearCase which would have been the alternative way, to have a folder for working documents that they could just access that way...*”

In an online environment, instant messaging (IM) is probably the least disruptive way to request someone’s attention. IM can be thought of as the virtual “ahem” or tap on a teammate’s open office door. Instant messaging is ideal for brief conversations much like the informal exchanges that occur when stakeholders are co-located. IM can mainly be used to obtain clarification during the requirements gathering and prioritization tasks. One RA stated that their local organization normally used I-Chat, an IM tool for the Mac environment, but that they were unable to use it on their distributed project because the other site did not have a Mac environment.

Chat room functionality allows multiple users to “converse” with each other simultaneously, and would enable stakeholders to participate in ongoing conversations by talking/writing and listening/reading. This

type of technology tends to require a substantial amount of system resources, so network bandwidths should be taken into consideration. Chat room technology is well suited for gathering requirements and ad-hoc discussions. Surprisingly our study found no use of chat rooms to support this type of communication. Although this is understandable when stakeholders are located in very different time zones, it could have been a useful tool for stakeholders in close time zones with overlapping workdays.

An online forum or discussion site lets stakeholders post messages, receive feedback from other project participants, and review existing postings for additional information. A major feature of online forums is its inclusive nature which allows large numbers of dispersed stakeholders to communicate from a location and time of their own choosing. Forums are ideal for sharing information and automatically provide a documented storehouse of that information. Much like email, a limitation of participating in forums is that a stakeholder cannot control when what they have posted will be read and when they will receive a feedback. Other limitations of forum usage include the proliferation of new message threads that occur when stakeholders are unsuccessful in searching for pre-existing relevant threads; and the difficulties in getting users with the same concerns to communicate [27, 28]. Gathering requirements is the most applicable distributed requirements development task.

Document Sharing Tools: Multi-cursor word processor and whiteboards provide electronic versions of the whiteboards or blackboards typically found in conference rooms. These boards can serve as effective document sharing tools that provide an online communal location where stakeholders can jointly create and edit documents and other project artifacts, e.g. spreadsheets, presentations, etc. A multi-cursor word processor or a wiki allows the project manager (PM) to delegate some of the responsibility for maintaining parts of the project repository. There is even an option to track which stakeholders are contributing. The major limitation of this technology is that the project manager or RA must also assign someone to be responsible for the overall review and maintenance of these artifacts.

Most of our RAs personally created and maintained the requirements documentation on their own personal computer or on a wiki, and shared it with other stakeholders via email, wiki access, or through a shared drive access. In one case, the RA modeled the requirements using pictures and diagrams since stakeholders spoke different languages. The documentation was created using desktop software like MS Word, Visio and PowerPoint; and shared with the stakeholders via email. According to the RG, *“In some cases, especially in the Asian location, it was better to print out the documents so that people could look at them and mark them up.”*

Though none of our RAs used an electronic whiteboard, one of them did note *“There are a couple of things you miss when you’re not face to face, one is the ability to use the walls, use the whiteboard, and use the actual physical space to communicate. And I know that there are some whiteboard applications out there and I haven’t used them, but I think that if that can be incorporated into the webinar technology that would be big.”*

Voting Tools: The suite of voting tools includes the mood meter and multiple-item ballot. A mood meter lets stakeholders discuss and vote on an open issue in an online environment. The stakeholders can change their votes as additional information becomes available provided. Formal voting occurs when consensus had been reached. The mood meter, which should be used when a group can focus on one issue at a time, can be a useful tool for prioritizing requirements.

Multiple-item ballot helps in tracking group consensus of multiple issues. Stakeholders can vote by ranking issues on a numeric scale, by agreeing/disagreeing or using a variety of other methods. Ballots are electronically distributed to each stakeholder and results are displayed as the votes are tallied. The RA may find the multiple-item ballot useful for obtaining an overall picture of the groups’ opinions about various issues.

Both mood meters and multi-item ballots could be used to support the requirements prioritization process. This type of tool was not used in any of the projects we studied; however it is probably to be expected given the overall low adoption rate of groupware tools. Our study showed that prioritization activities were primarily performed during face-to-face or telephone meetings.

Synchronous Meeting Tools: Desktop and application sharing, and audio and video conferencing technologies are considered synchronous meeting tools that allow stakeholders to simultaneously view and/or control a desktop or application from a remote location. These tools are ideal for demonstrating current applications, work-arounds and prototype designs and new functionality. They are also useful when stakeholders are jointly developing or reviewing project artifacts, such as requirements specification or models. However synchronous meeting tools require higher bandwidth availability, which may well be a consideration at some remote site networks.

Teleconferencing is a well-used technology, (and the most preferred by our RAs as described in Telephone Conference Pattern); which enables all stakeholders to participate in synchronous meetings. It is best suited for use when real-time communication is needed for discussion and clarification. A

limitation that should be noted is that after a session, meeting minutes and reports may have to be recorded and transcribed in a timely manner.

Videoconferencing enables stakeholders to actually see each other during meetings, and is therefore useful for facilitating face-to-face meetings. Organizations should consider the trade-off between the technical infrastructure that is needed at remote sites versus the focus of a meeting. One RA indicated that videoconferencing was only used a couple of times, as this technology, *“didn’t seem successful at focusing on details important to the meeting.”* They explained that it was less important for them to see other meeting participants than it was to view a diagram for example. Another RA shared the following, *“Web conferencing is not too bad but, it’s hard to be patient to watch people use a tool like Visio to draw a diagram...In a perfect world it would be fantastic just to have a tablet. With a tablet, just draw your activity diagram or draw your context diagrams. Just really quickly do it in a way that other people can see what you are drawing without having them see me. I just want them to focus on the model I am creating.”*

Information Access Tool: A shared repository is a central network location where project artifacts and data are stored. Stakeholders can be granted access to modify and maintain the artifacts. These tools provide settings to establish and maintain stakeholder access at the artifact level. Shared repositories are ideal for big projects with artifacts that will be shared over a long period of time by numerous stakeholders. They provide support for the tasks of documenting and managing requirements. Two of the projects we studied used SharePoint, which provides a virtual workspace and facilitates document sharing [88]. In one case, the tool was found to be *“very slow”* and the RA switched over to using an intranet-wiki instead. This issue may have been caused by insufficient bandwidth or a slow processor however rather than the actual tool.

Recommendations:

Since the RA is not necessarily a requirements engineer and therefore may not be familiar with RE best practices, we recommend providing guidance on the availability and selection of applicable tools. Based on the identified challenges and survey of collaboration tools we propose a simple instrument to help RAs select appropriate tools. The instrument utilizes the information in Table 5.6, which documents the types of tool suited to each of the primary requirements engineering activities, and Table 5.7 which highlights some of the strengths and limitations of each tool type.

At the IEEE International Conference on Requirements Engineering (RE09), Gotel and Mader presented a tutorial about how to select a requirements management tool [99]. Many of their recommendations for evaluating and selecting these tools are applicable to our research as well. They noted that one of the most extensive evaluation frameworks that exists and is maintained is available on the International Council on Systems Engineering (INCOSE) website. But to date, no such guidelines are available for selecting collaboration tools for the distributed requirements engineering domain.

Furthermore Mittleman and Briggs [88, 98] proposed specific criteria to help stakeholders evaluate a specific collaborative tool, with respect to factors such as core functionality, supported actions, content, access controls and alert mechanisms and can be used to select tools to support distributed requirements processes. These criteria are also incorporated in our proposed process.

Requirements Development Tasks	Category of Collaboration Tools
Gather	Conversation, Document Sharing, Synchronous Meeting
Document	Document Sharing, Synchronous Meeting, Information Access
Prioritize	Conversation, Document Sharing, Voting, Synchronous Meeting
Manage	Conversation, Information Access

Table 5.6 Requirements development tasks and categories of tools

Type of Tool	Strengths	Limitations
Conversion		
Email	Communicate with ≥ 1 stakeholder simultaneously. Mass distribution of project information.	No control over when message is read / when to expect response. Complicated to manage project documentation in folders.
Instant Messaging	Brief conversations / informal communications. Obtain clarification.	
Chat Room	Track group consensus on multiple issues.	Time zone issue - stakeholders must participate in real-time. Uses a lot of network bandwidth.
Online Forum	Stakeholders can post messages and receive feedback from other stakeholders.	No control over when message is read / when to expect response. Proliferation of duplicate message threads if robust search functionality is not available.
Document Sharing		
Multi-cursor Word processor	Stakeholders can collaborate on documentation synchronously.	Time zone issue - stakeholders must participate in real-time.
Wiki	Stakeholders can collaborate asynchronously. PM can delegate some of the documentation responsibility.	PM / RA may have to establish guidelines for feedback loop.
Multi-cursor Whiteboard	Stakeholders can view and update whiteboard during online meetings.	
Voting		
Mood Meter	Stakeholders discuss and vote on one open issue at a time.	
Multiple-item	Track group consensus on	

Ballot	multiple issues.	
Synchronous Meeting		
Application and Desktop Sharing	Current system and prototype demos.	Tool needed on at least one machine in every site.
Audio-conference	Allows all stakeholders to hear the exact same information at the same time.	Time zone issue - stakeholders must participate in real-time. May need to prepare and distribute meeting minutes in a timely manner.
Videoconference	Allows all stakeholders to see as well as hear each other.	High-quality technical infrastructure is needed.
Information Access		
Shared Repository of Files	Stakeholders can access and update project documentation as fits their schedules.	PM / RA must monitor repository access.

Table 5.7 Types of collaboration tools

Guidelines for Selecting Distributed RE Communication and Collaboration Tools

Step 1: Model sites, stakeholders, and communication paths as requirements-gathering collaborative networks (RGCNs) as described in Chapter 5 of this thesis. For example, Figure 5.9 depicts the communication model for Project 3 which involved four different sites, two of which were in the U.S., one in Europe, and one in Asia. The RA was a U.S. resident assigned to the Asian location. As depicted by the communication flows in the diagram, the RA communicated directly with stakeholders at all remote sites.

Step 2: Map specific tasks and current tool support onto the RGCN and evaluate pain points

The next step involves modeling the current practices by mapping the four requirements development tasks of elicitation, documentation, prioritization, and management onto the basic RGCN. The mapping of the documentation task is depicted in Figure 5.10. In this case, documentation was centralized around the RA. Currently available tools include the use of textual documents, spreadsheets, and diagrams exchanged by email. In this particular project there were several observable pain points. First, stakeholders created documents on their own machines, and as there was no centralized formatting template, the format of the project documents was not consistent and needed to be manually “translated” into a standard format. Secondly, significant time was spent tracking documents to determine who had the

latest copy, what changes were made to it, and where it was stored. Finally, the RA had to ensure that all sites were using compatible versions of the desktop software used for the documentation.

Step 3: Evaluate potential for tool support

The next step uses Tables 5.6 and 5.7 to identify possible tools for supporting each of the requirements activities. First, Table 5.6 is used to select an appropriate tool category, and then Table 5.7 is used to identify specific types of tools in each category. In this example, the “document” task is selected from Table 5.6, and then multi-cursor word processors and wiki technologies are selected from the Document Sharing and Information Access tools section of Table 5.7. These tools have the capability to address the identified pain points, i.e. document updates made and tracked within one “master” copy; afterwards the documents can be stored and shared from a repository that has been implemented with check-in and check-out privileges.

Step 4: Tool Selection

The final stage of the process involves actual tool evaluation and selection. Tools offering multi-cursor word processing and also wiki tools are evaluated using the following 9 criteria proposed by Mittleman [98]: required functionality, type of data, relationships, supported actions, access controls, access parameters, persistence, awareness indicators (i.e. methods for allowing stakeholders to find out what other stakeholders are doing) and alert mechanisms. As clearly depicted in our study, tools that do not fit the available platform, that require too much bandwidth, or are simply difficult for participants to use will not result in successful tool adoption. This step is therefore a critical one in the process of tool adoption.

Step 5: Tool deployment and adoption

Again based on our observations from the projects we studied, tool use must be carefully planned and provided with appropriate support to iron out technical problems.

Step 6: Tool Evaluation

To ensure that tooling decisions were appropriate for the project, the tool choices should be assessed throughout the project. In many cases, problems with tools, such as slow response times can be resolved by adding a faster processor or increasing bandwidth availability. Other times, tool adoption may be slow due to poor training. Providing an evaluation loop is likely to increase the chance of successful tool adoption.

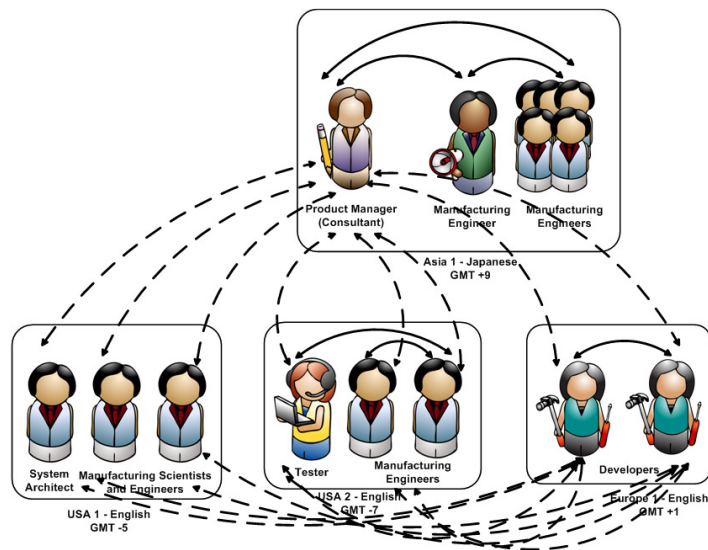


Figure 5.9 Communication Flow of Project 3 modeled as a Requirements Gathering Collaboration Network (RGCN)

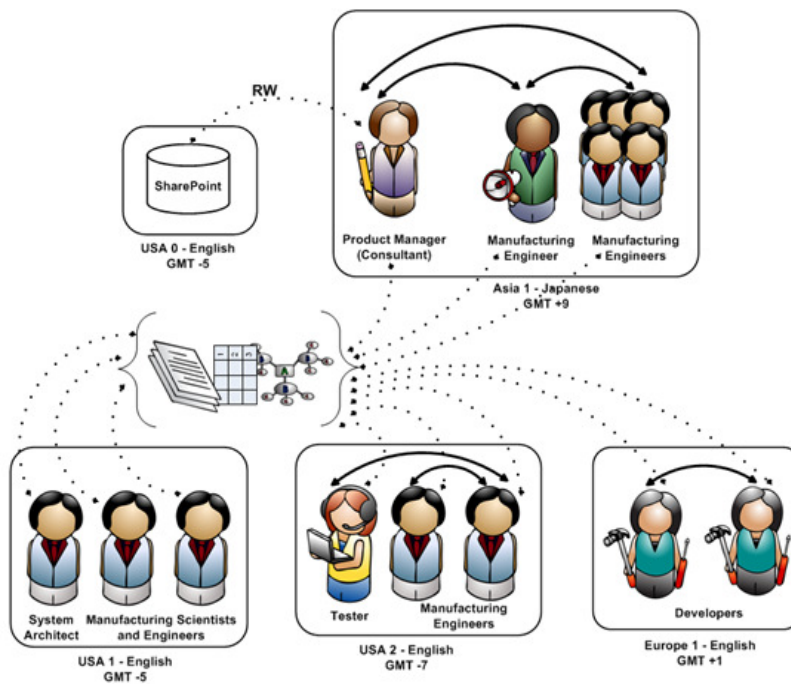


Figure 5.10 Documentation flow of Project 3

Open Questions:

1. How / when are RAs first introduced to the tools and technologies they currently use?
2. Do RAs have the authority to recommend new tools and technologies?
3. What is the organizational / departmental process for purchasing new tools and technologies?

5.3.5 Pattern: Face to Face Communication

Pattern: Though the RA is not co-located with any stakeholders face-to-face meetings occur when the RA travels to remote sites.

Observations:

Even when the requirements analyst (RA) was not co-located with any stakeholders, many of the requirements engineering activities occurred during face-to-face meetings since the RA, usually a consultant working on a new project, traveled between the remote sites. In all of the projects included in Table 5.8 there was some degree of face-to-face communication between the RA and the stakeholders. Projects 3, 4, 7, 8 and 10 were projects where the RA was a consultant brought in to lead the requirements engineering activities. Additionally, for projects 3, 4 and 7 the RA traveled to the customer/stakeholder sites to elicit and gather requirements for new software engineering projects.

For project 3 typically the RA, *“spent several weeks in the Asian location and then a few weeks in the US.”* In-person communication between the stakeholders and Project 3’s RA was also possible when the stakeholders from the U.S. and European sites traveled to the Asian location. The project requirements were also analyzed and prioritized via face-to-face meetings in Asia, since this is the where majority of the end-users resided. The prioritization meeting occurred mostly with the main representatives of different functional areas in one big meeting.

Project 4’s RA shared the following, *“My personal preference is always to do the elicitation in person, when that was practical and possible. For that reason, we used to travel to and from Asia, and because I had the luxury of having a large team, we could have had about six or seven people at any given point of time. We typically attempted to do one on one interviews with local subject matter experts at their respective locations.”* The RA continued, *“it was not uncommon for the people from USA1 and USA2 to also be traveling to Asia. So if I knew, for example, that so-and-so was going to be there then even though that particular topic may not be coming up as a requirement for let’s just say two releases down the road I would basically take the opportunity to shoot ahead of the curve because now I knew that I had everybody in the room at the same time.”*

The RA’s team consisted of the RA and 2 others who traveled to the five sites, for Project 7. The RA’s team mostly conducted face-to-face interviews with groups of 2-5 stakeholders in a conference room at one time. As the RA explained, *“We always went together because basically you’d need a moderator; someone to document everything, and a third person as co-moderator. I mean you can do it with two*

persons too, but we just decided because we're <involved> in a new project that we needed as many ears and eyes as possible."

Not only did Project 8's RA travel, but Europe 1's LSP also took trips between the U.S. and Europe, which provided additional opportunity for in-person communication. The RA shared that she conducted face-to-face stakeholders meetings, *"To conduct system requirements reviews, to conduct contract requirements reviews; that type of thing."*

Interestingly enough for Project 10 the RA was not involved in the kickoff meeting. Instead it was the RA's manager that visited the USA1 customer/stakeholder site for the kickoff meeting. The requirements development activities were then continued with the customer and vendor sites via separate telephone conferences. *"For that particular kickoff, I really didn't have to be there. <The management team> really had to do most of that work. They went through the script, they gathered the information of who to talk to, and then I went in and called <the customer/stakeholders>...Normally I would actually go to the kickoff... They only had a couple of people go to cut down on the expenses for the project",* said the RA.

In addition to RE specific activities the RA and stakeholders also met in person during the RA's user/customer site visits. These visits helped to strengthen the rapport between the RA and stakeholder. According to an RA, *"the best way we <elicit requirements> is by a customer visit; if you have the existing relationships with the customer.... that have bought other products of ours..."* And another RA noted, *"We tried to visit <the distributed sites> every 6 months to a year. It was piggy-backed on top of another event, <like a> conference. Yes those meetings definitely did help the relationship. No questions at all."*

Project No.	Industry	Locations and Roles	RA at own site	Traveler	Travel purpose
3	Software Requirements Consulting	Asia1 – RA + 1 LSP + 29 Stakeholders Europe1 – 2 Stakeholders USA1 – 4 Stakeholders USA2 – 3 Stakeholders	No	RA	Gather requirements
4	Software Requirements Consulting	USA0 – Project RA USA1 – 15 Stakeholders USA2 – 20 Stakeholders Asia1 – Site RA/LSP + 30 Stakeholders	Yes	RA	Gather requirements
5	Video Games	USA1 – RA USA1A – 3 LSPs + 8 Stakeholders USA1B – 2 LSPs + 4 Stakeholders USA 2 – 2 LSPs + 5 Stakeholders USA 3 – 2 LSPs + 3 Stakeholders USA1C – 2 LSPs + 3 Stakeholders USA 4 – 2 LSPs + 3 Stakeholders Europe1 – 2 LSP + 4 Stakeholders	No	PM RA	Site meetings
7	Software Engineering Research	Europe0 – RA and Team Europe1 – 5 Stakeholders Europe2 – 5 Stakeholders Europe3 – 5 Stakeholders Europe4 – 5 Stakeholders Europe5 – 5 Stakeholders	Yes	RA	Gather requirements
8	Corporate Research	USA0 – RA USA1A – 3 Stakeholders USA1B – 2 LSPs + 23 Stakeholders Europe1 – 1 LSP + 4 Stakeholders	Yes	RA LSP	Gather requirements
9	Information Technology	NorthAmerica0 – RA USA1 – 2 LSPs + 5 Stakeholders	Yes	RA	Customer site meetings

		USA2 – 2 Stakeholders USA3 – 8 LSPs + 8 Stakeholders USA4 – 7 LSPs + 6 Stakeholders USA5 – 6 LSPs + 7 Stakeholders			
10	Securities Software Solutions Integrators	USA0 – RA + 3 Stakeholders USA1 – 3 LSPs + 100 Stakeholders USA2 – 4 Stakeholders	Yes	PM	Kickoff meeting

Table 5.8 Face-to-Face Communications

Existing Literature:

Much of the academic and industry research stress the importance and benefits of in-person meetings, especially during for the project kick-off. Our observations align with those of Coplien and Harrison [92] that since it's harder to communicate when geographically separated, it is beneficial to the project to bring people together for face-to-face meetings to help “establish project unity” and so stakeholders will get to know each other. Though their “5.1.10 Face to Face Before Working Remotely” pattern specifically advocates this for the project kick-off meeting; any time users can meet each other helps to build a sense of team unity.

Face to face kickoff meetings for globally distributed project teams are also recommended by Damian and Zowghi “to establish initial personal relationships between key stakeholders and to put the bases for strategic planning” [100]. As the project continues, scheduled informal meetings between the distributed sites should continue as a way to strengthen these relationships.

Recommendations:

Our findings are in agreement with literature – at the very least the initial project kickoff meeting should be an in-person, face to face gathering that includes all of the stakeholders.

5.3.6 Pattern: Requirements Specifications Shared only via Email

Pattern: When the stakeholders do not have access to the requirements repository the RA will collaborate on the specification with them via email.

Observations:

The RA was the only person with access to the requirements repository for smaller scale projects or when the infrastructure does not support stakeholder access. Surprisingly, it didn't matter if stakeholders were members of the RAs organization or not, the system infrastructure appeared to be a prime factor affecting the stakeholders' access to the project repository, if one existed. Table 5.9 contains the projects where the RA discussed their email process.

Project 1's RA created and maintained the requirement specifications in MS word. The specs were then stored on a public drive within the RA's organization, and emailed to external stakeholders for their review. Meanwhile the requirements documentation for Project 3 was drafted in MS Word, with imbedded Visio diagrams, spreadsheets and PowerPoint, and kept up to date by the RA and other members his consulting team. Project 3's RA would share these specifications with their stakeholder/customers by either emailing a copy of the specs or a link for downloading the specs.

According to Project 4's RA, SharePoint was their document storage mechanism. Because English was not the first language of many of the stakeholders, *"We tried to make our requirements documents as graphical as possible. But we also had detailed use cases and things like that in the documents along with very detailed requirements statements..."* The process for creating and maintaining the requirements specification was as follows: an RA would draft a document and then email it one-to the librarian for safekeeping, and two-to the necessary stakeholders for their review and approval. The librarian who was located in the U.S., *"because we were so distributed amongst so many physical locations"*, was responsible for loading the documents into SharePoint. After reviewing, the stakeholders would reply to the RA. The RA would then make the necessary modifications and update the version number before emailing the revised artifacts to the librarian.

Project 6's RA developed the requirements documentation in MS Word; while Use Cases were created using Visio. Prototypes were an appendix to the Use Cases and Wire Frames illustrating main and alternate scenarios would follow the use cases. The requirements specification was emailed to the users for their review. Per the RA, *"In most cases I would get <documentation updates> through email. So <the stakeholders> would either put their comments in the body of the email message without an attachment of*

their markups; or I would get an attachment with their markups embedded and I would have to go through the document; and then I would integrate these comments and of course you run into situations where there's maybe discrepancies or something's not clear. You have to negotiate those different things together." The use cases were also shared via email. As the RA explained, *"Well unfortunately at the time that we did this, I had to email <the use cases> since we didn't have a Wiki, and then our business users didn't all have access to ClearCase which would have been the alternative way, to have a draft folder for working documents that they could just access that way. So we just had to email them."* The RA also shared that this difficulty in sharing the use cases was one of the reasons that his department later created a project wiki.

For Project 10 requirement specifications consisted of MS Word documents, spreadsheets, and user interface and use case diagrams that were created by the RA. The documentation was shared with the stakeholders via email. Next the stakeholders provided feedback by noting corrections on a copy of the relevant artifact and returning it back to the RA, who then updated the master copy. When asked if it was difficult using email to keep track of the specifications, the RA replied, *"not really. It may have worked out a little bit better if it went directly into a tool for tracking, a database for tracking. But since the project wasn't that complicated it was easy enough to look the information up in email. I could see if that got to be a huge project that it would definitely be an issue."* The artifacts were stored and maintained in the RA's own email folder. The RA continued, *"And of course I copied pretty much the world for almost everything. That's another reason why it would've been better for it to be in one repository and one database if it was a larger project. Rather than having it proliferated all over the network for our business."*

Project No.	Industry	Locations and Roles	Stakeholder access to Repository
1	Financial Services	USA0 – RA USA1 – 1 LSP + 2 Stakeholders	No
2	Telecommunications	USA0 – 1 RA USA1 – 3 Stakeholders USA2 – 2 Stakeholders	Yes
3	Software Requirements Consulting	Asia1 – RA + 1 LSP + 29 Stakeholders Europe1 – 2 Stakeholders USA1 – 4 Stakeholders USA2 – 3 Stakeholders	No
4	Software Requirements Consulting	USA0 – Project RA USA1 – 15 Stakeholders USA2 – 20 Stakeholders Asia1 – Site RA/LSP + 30 Stakeholders	No
6	Retail	USA 1 – RA + 2 Stakeholders USA2 – 1 LSP + 5 Stakeholders	No
7	Software Engineering Research	Europe0 – 3 RAs Europe1 – 5 Stakeholders Europe2 – 5 Stakeholders Europe3 – 5 Stakeholders Europe4 – 5 Stakeholders Europe5 – 5 Stakeholders	Yes
8	Corporate Research	USA0 – RA + 6 Stakeholders USA1A – 3 Stakeholders USA1B – 2 LSPs + 23 Stakeholders Europe1 – 1 LSP + 4 Stakeholders	Yes
10	Securities Software Solutions Integrators	USA0 – RA + 3 Stakeholders USA1 – 3 LSPs + 100 Stakeholders USA2 – 4 Stakeholders	No

Table 5.9 No Face-to-Face Communications

Recommendations:

Our observations concur with Mittleman et al as described in Section 5.3.4 Requirements Engineering Tools and Technologies Pattern that email can be an effective tool. Much like the telephone in Section 5.3.2 Telephone - RA's Preferred Communication Tool; stakeholder collaboration via email can be sufficient, especially if it is not a large project, for example Projects 1, 6 and 10 were comprised of twelve, fifty and thirty requirements, respectively.

5.3.7 Pattern: Distributed Stakeholder Communications

Pattern: Distributed stakeholders also communicate and collaborate with each other.

Observations:

While it is expected that co-located stakeholders communicate with each other while working on projects, we observed several projects in which distributed stakeholders talked to each other on a regular basis, at times without the RAs presence as facilitator. Refer to the projects in Table 5.10.

Project 3's RA shared that the Europe1 developers conversed with the U.S. SMEs and technical stakeholders, even though the European stakeholders had limited English-speaking capabilities. The RA also observed that though stakeholders in the US1 and US2 did talk to each other, the conversations were not related to this project. While Project 6's RA commented that, *"there was some <stakeholder> communications that I was not privy to... I would get it through the spokesperson."*

According to Project 4's RA the distributed stakeholders *"were always talking to each other."* The RA agreed that she did not have to arrange and facilitate separate meetings to discuss a particular requirement because the dispersed SMEs would be aware of the topic of concern since they had already discussed it amongst themselves. The RA continued *"...it was not uncommon for the people from US1 and US2 to also be traveling to <Asia1>."*

Project 5's stakeholders did communicate with each other across distributed sites. The RA observed that though this was not an assigned task; *"For almost every team <of stakeholders> that got added on, there was one buddy team or mentoring team, to help and train them; not only on the software process but on how to use this custom tool."* The RA shared that at times the stakeholder-only communications caused friction between the stakeholders and the development team, since the RA and developers would have *"no knowledge of some of the communications or decisions that were done, yet we were responsible for the support of it."* The RA continued, *"The problem here is the dissemination of information. What would happen is that we would have these <group> meetings and certain things were agreed on.... <Later when the stakeholders talked amongst themselves>, certain directions turned, things were tweaked. The leads would not disseminate the <new> information. They would even agree to certain issues or topics or directions or talk about resource allocation.... then later change their story and say <to the RA>,' we didn't agree, you guys are at fault'".* In response the RA changed the process so that all "official" meetings were recorded, and created a packet that contained *"summary of notes, summary of the commitment for that sprint, also summary of the proof of deliverables, participants, and the video or*

audio recordings.” The packet of information was now, “disseminated to any user and everybody. So we had this huge distribution list....Everybody gets the information. <Stakeholders> can’t cry wolf.”

Project No.	Industry	Locations and Roles
3	Software Requirements Consulting	Asia1 – RA + 1 LSP + 29 Stakeholders Europe1 – 2 Stakeholders USA1 – 4 Stakeholders USA2 – 3 Stakeholders
4	Software Requirements Consulting	USA0 – Project RA USA1 – 15 Stakeholders USA2 – 20 Stakeholders Asia1 – Site RA/LSP + 30 Stakeholders
5	Video Games	USA1 – RA USA1A – 3 LSPs + 8 Stakeholders USA1B – 2 LSPs + 4 Stakeholders USA 2 – 2 LSPs + 5 Stakeholders USA 3 – 2 LSPs + 3 Stakeholders USA1C – 2 LSPs + 3 Stakeholders USA 4 – 2 LSPs + 3 Stakeholders Europe1 – 2 LSP + 4 Stakeholders
6	Retail	USA 1 – RA + 2 Stakeholders USA2 – 1 LSP + 5 Stakeholders

Table 5.10 Distributed Stakeholder Communications

Recommendations:

Our observations support previous research in the software engineering and project management domains [1, 7, 14, 17, 21, 24, 92], stakeholder-to-stakeholder communication is essential to the successful development and implementation of software engineering projects.

Open Questions:

Which distributed stakeholders are communicating with each other?

What are distributed stakeholders preferred methods of communications?

How often do distributed stakeholders communicated?

Are these interactions confined to specific phases of the project?

5.3.8 Pattern: Multipurpose Requirements Specifications

Pattern: The requirements specifications, usually authored by the RA, are used in multiple phases of the software engineering process.

Observations:

No matter the software development methodology practiced by the organization, the requirements specification is instrumental during various phases of the engineering life-cycle. For example it utilized is during the conceptual phase by project managers establishing scope; during the design and development phases by technical team members such as architects and developers; and during the testing phase by the quality assurance (QA) team, etc. Table 5.11 lists the applicability of the specifications to the software engineering tasks.

During Project 1 the RA gathered the requirements and shared the specifications with the stakeholders and development team. The RA explained, *“I did <share requirements specs with the developers at my organization, specifically the lead architect and the software engineer >. I’m not sure if it was important to them as just getting the appendix for the actual field requirements because that’s usually/really what they are looking for”*, since it was the software engineer who performed the actual coding.

For Project 2 once the requirements specification are posted to the RA’s internal internet site, *“Everyone who needs it has access so developers, other designers who might be designing something different that kind of ties in to my design. Product managers, marketing people, testing people should all have access to that document.”*

The requirement artifacts for Project 3 were a combination of SharePoint, MS Word documents containing many Visio diagrams; spreadsheets, and PowerPoint slides. This set of documentation became the requirements specifications that *“developers would consume... and then create design specifications out of them”*, according to the RA.

Project 4’s RA explained *“...because of the unique nature of the project, ‘unique nature meaning’ that when they created the statement of work they created it against the requirement document. So the requirement document also had to be pretty specific in terms of the specific requirements, the specific use cases and so on; because that’s what the components were eventually being measured against to determine whether or not they had satisfied the terms of that particular statement of work. So the requirements document had a dual purpose. One was obviously to provide information to the*

development community so that they had sufficient information to create the software. It was also serving a legal purpose in providing the documents of record against which a statement of work was being estimated by the vendor, against which they were delivering and so on and so forth.”

For Project 5 the development team employed an agile methodology. Per the RA, “...*we did iterative <development> and then interacted with the customer to get their feature requirements.*” The RA elicited and gathered requirements using a wiki. All of the user requests (i.e. bugs, feature requests) were or became requirements for the development team to work on.

For Projects 6, 8 and 9 the project stakeholders were mainly developers and other technical resources, so the RAs worked with each of their teams to create the specifications. According to Project 6’s RA at the beginning of the project the high-level requirements had to be re-worked so that they would be “*mutually acceptable to both the business and IT delivery manager sign-offs*”. Project 9’s RA explained the following, “...*in discussions with development, we’d say ‘we have these 50 requirements which emerge as, say ten features’. Their focus is on the features because they are looking at software for us, in terms of features, functionality, its capabilities. Then they would go to break up those features into workable tasks, units of development work. And then they work through those tasks in order to find the background... to inform how they should implement it. They’re going to be referring to the requirements and use cases that are linked to those features and tasks...*” Additionally the requirements specifications were also “*passed on to the subcontractors and the developers*” according to Project 8’s RA.

During the first iteration the Project 7’s RA and his team gathered the requirements and a prototype, which was basically the first implementation of the application, was constructed and shared with the stakeholders. Then “*the researchers and the developers implemented the system or parts of the system, and then we had an evaluation with all of the stakeholders that we interviewed. We <iterated through this process> three times in all, over the year; and we evaluated the different features from the researchers that they implemented <to see> if it met the <stakeholder> requirements or not. And <we also> had to ask if the requirements were implemented well or not.*”

During Project 10 the requirements specification was shared with the RA’s technical staff, “*because they needed to know what we were telling the client that we were producing and what the client agreed on that we would produce. Also we had to make sure that our technical staff gave us the OK that ‘yes, we can produce that’.*”

Project No.	Stakeholders	Purpose
1	Developers	Code updates and change implementations.
2	Developers Testers Managers	Design and development Testing Marketing
3	Developers	Review to create design specifications.
4	Developers Project Managers	Create software. Review to create Statement of Work.
5	Developers	Code updates and implement changes.
6	Developers and Technical SMEs	Code updates and implement changes.
7	Developers	Design and development
8	Developers and subcontractors	Design and development
9	Product Managers Developers	Managing products Design and development
10	Technical staff	Code updates and modify vendor software

Table 5.11 Requirements Specification Utilization

Recommendations:

Here again our observations corroborate with previous research about the importance and utility of requirements specifications [1, 2, 4, 6, 8, 40]. The creation and maintenance of such project artifacts is considered a best practice in successfully creating, managing and delivering software projects.

5.3.9 Pattern: Travel Between Sites Unnecessary for Small Groups

Pattern: Face-to-face communications are not mandatory for projects consisting of a small group of distributed stakeholders.

Observations:

Travel was not mandatory when the project team was comprised of a small group of stakeholders and the stakeholders had access to project artifacts for bigger projects. With the participation of fewer than ten stakeholders the RA can effectively schedule and manage remote meetings. This pattern is the opposite of the Face-to-Face Pattern, so consequently the projects listed in Table 5.8 are the ones excluded from Table 5.12 in the Face-to-Face Pattern description.

All projects in this pattern were basically enhancements to organizations' current product line. For example, Project 1 was a sales opportunity for the organization to repurpose a current application for a new industry. According to the RA, this was a data integration project. The organization had to provide data already available from their current platform plus new customer-requested data. For this project the RA gathered requirements from only one remote site and felt that telephone conferences and email proved to be sufficient tools. According to the RA, *"I didn't use a requirements database since this project was a small scale change to an existing platform. When it's less than two dozen <requirements>, I just kind of base it on a document that you can count the bullet points. That's really what it is, where you're tracking your changes. It was such a small scale change to the platform that it didn't require a requirements database."*

Project 2 was an enhancement to an existing cell phone camera application. The application was being re-designed for a new suite of products. The stakeholders were members of the organization's internal product development teams. The RA communicated and collaborated with them during telephone conferences and shared design and flow documentation. According to the RA, *"I had to get all of the stakeholders involved because there were different products and different product managers who were sometimes not aware of each other that were trying to do the same thing. So I at the central point had to reach out to all of those people, get everyone on the same phone call and start saying, 'look, you're all trying to do the same thing.' And my job is trying to provide that we can reduce the amount of design effort and costs by just sharing the code across the different..."* As the requirements gathering process continued, *"Usually I had mockups or screenshots that we had created to help talk through them... We do have design documents that are detailed specs but it's easier when you're dealing with people who are more visual, it's easier to show them a picture and talk about it."*

The RA for Project 6 described his project as “*an extension of existing functionality*”. His organization already had an application that provided their external users with the capability to purchase a physical gift card online. “*The goal of the project was to provide the retail/random (external) customer the capability to select a design for an online gift card, define the denomination amount or value of the gift card, be able to submit multiple gift cards simultaneously to separate recipients, (with) recipients being equal to email addresses, and also provide the capability for reloading value on to those online gift cards*”, said the RA. One of his main responsibilities was to “*take the high level requirements from the business and rework them to be mutually acceptable to both the business... and IT.*” Communication between the RA and his co-located stakeholders occurred either face-to-face or by telephone; and with USA2’s stakeholders via telephone conference. The RA did not express any dissatisfaction with this communication flow, especially as all of the stakeholders were internal to his organization and accessible to him.

Project No.	Industry	Locations and Roles	RA at own site
1	Financial Services	USA0 – RA USA1 – 1 LSP + 2 Stakeholders	Yes
2	Telecommunications	USA0 – 1 RA USA1 – 3 Stakeholders USA2 – 2 Stakeholders	Yes
6	Retail	USA 1 – RA + 2 Stakeholders USA2 – 1 LSP + 5 Stakeholders	No

Table 5.12 No Face-to-Face Communications for Small Groups

5.4 Proposed Solutions and Conclusion

We have identified current industry distributed RE challenges and successful strategies; and categorized our findings in the format of the organizational and activity patterns described above. These patterns can enhance traditional RE methodologies for large-scale distributed software development projects by functioning as guidelines that can assist practitioners in effectively planning and executing their global requirements engineering projects.

This study found many instances of stakeholders using email. It was viewed as an effective tool for mass distribution of project artifacts such as documents and links to files. An RA expressed that email was the only tool they needed since theirs was a relatively small project, requiring modifications to a current application. We also discovered several projects in which email was basically the only tool used to manage all of the collaboration needs. This led to various problems. Some of the RAs collaborated on documentation with stakeholders via email. An RA noted that all of the email messages and attachment cluttered their email account, *“I would prefer web-based interface to track things.”* This RA found it difficult to manage the requirements development process since there was not a way to automatically separate the project-related emails from their other emails.

We noted several instances of email folders being used to manage the requirements documents. A major disadvantage to this process is that special attention is then needed to manage document versioning and keep track of who has the current version, and the location of most updated documents. When describing their process for sharing use cases with stakeholders, an RA noted, *“I had to email them since we didn’t have a Wiki, and then our business users didn’t have access to ClearCase which would have been the alternative way, to have a folder for working documents that they could just access that way...”*

These case studies represent the results from our survey of distributed requirements gathering projects; and they suggest that a number of factors constrained the structuring and interactions that occur within RGCNs. First, specific barriers such as different languages limit interaction and force a restructuring of the RGCN that facilitates stakeholder communication indirectly through a central representative, the LSP. Secondly, in some cases, PMs chose to isolate sites because the stakeholders were working on separate parts of the project, and direct interaction was deemed unnecessary. In a non-distributed project, this separation would be augmented through informal relationships across teams; however in a distributed project there are few opportunities for informal communication. As a result, no social or technical framework was created between sites, which made communication difficult even when it could have been

beneficial to the project. In most of the case studies, there were strong indications that technology limitations hindered collaboration between stakeholders, and meant that project managers had to opt for less sophisticated means of communication such as teleconferencing or emailing instead of more interactive methods such as video-conferencing.

Our findings show that the challenges of geographically distributed software projects affect the ways in which stakeholders cooperate to complete core requirements-related activities such as requirements elicitation, analysis, prioritization, and specification, and highlight the importance of explicitly planning the infrastructure and communication flow needed to support collaboration between remotely located stakeholders [12]. Our findings also illustrate the importance of proactively identifying stakeholder roles, along with their locations, communication flows, critical project artifacts, and necessary tool support. Results from our study suggest that projects which failed to consider these issues were impeded by problems such as disorganized stakeholder interactions, lack of appropriate tool support, data over-load, increased travel requirements, and inefficient processes for supporting specific requirements engineering tasks.

Chapter 6

RE Modeling Research Sessions Data Analysis

The data described and analyzed in this chapter are the analysis of our RE Modeling Session study that began during the Fall of 2012. Refer to Chapter 4 for a full description of our data-gathering methodology.

6.1 RE Modeling Research Sessions

For this follow-up research study we observed industry professionals utilizing and evaluating our CGREN as they modeled distributed requirements engineering activities associated with their respective projects. Three software engineering industry professionals who had the role of RA for the discussed projects, participated in this study. Each RA represented a different industry, RA1 was from technical consulting, RA2 was from academia and RA3 was from healthcare; and held a minimum of three years experience as an RA. Their job titles within their companies were consultant, business analyst and director, respectively.

We conducted this study using a tactile approach in which icons were printed onto small cards, and the participants utilized a white board and markers to construct their models. The researcher met individually with each participant in an office setting that also contained a speaker phone for recording the session.

At the beginning of each session we discussed the requirements phase of a sample project with the participant and then demonstrated how to model the project using our framework at the whiteboard. We then distributed and reviewed the glossary of our visual notation as displayed in Figure 6.1. Participants were then asked to think of a current or past project in which they had to elicit and gather requirements from stakeholders who were not co-located with them. After describing their project in terms of number and type of stakeholders; number of locations, communications methods, etc., each participant was asked to model their projects' RE activity using CGREN at the whiteboard. The researcher assisted by providing the requested icons from the visual notation toolset. The study involved a 'think-aloud' protocol

augmented by specific questions from the researcher. After completion of the modeling task researchers solicited feedback from the participants regarding the CGREN and its usefulness in modeling activities in their work settings, i.e. plan distributed RE activities, early detection of potential issues, etc. using the series of exit questions depicted in Appendix A.

The study was designed to address three primary research questions (RQ):

- RQ1:** *To what extent are project managers able to utilize the CGREN to model distributed requirements engineering processes in their projects? Are any important concepts missing or in need of improvement?*
- RQ2:** *Does the CGREN help analysts identify problems and/or improve the infrastructure of their projects?*
- RQ3:** *What is an effective process model for utilizing CGREN to model a project?*

Our study was qualitative in nature. Research questions were systematically answered as a result of observing the participants utilizing CGREN, reviewing transcripts of the sessions, and through evaluating the answers to the open-ended exit survey questions.

6.2 Research Findings

Using the CGREN to create RGCNs all of the RAs were able to successfully model the roles, locations, communication methods, and artifacts of their selected projects. When asked “were you able to model all the concepts from your project?” all three participants responded positively. Furthermore the RGCN models produced during each of the three sessions demonstrated that all three RAs developed models which they claimed fully represented their projects, and which were correct with respect to the meta-model. However, when specifically asked if any graphical symbols were missing, two of the participants mentioned the need for the notation to allow stakeholders to assume multiple roles, sometimes simultaneously, and sometimes at different phases in the project. RA2 also pointed out the need to “denote frequency of communication” in order to differentiate between varying communication frequencies along different communication channels. In general, the results of this study confirmed that CGREN provided the ability to model most aspects of the distributed requirements engineering processes that the RAs were engaged in.

6.3 Proposed Solutions

Based on our previous experiences and our observations during the participatory study we developed a simple set of guidelines that can be used in conjunction with CGREN. The process assumes that the purpose of the project has already been clearly articulated and that an initial set of project stakeholders have been identified. The process includes the following steps:

1. Identify primary locations and model them as sites in CGREN.
2. Identify project-level organizational roles. These roles can be assigned to specific sites as the organizational plan evolves.
3. For each site, identify key local roles and communication patterns between roles within the site. Differentiate between roles which communicate externally and those which do not.
4. Establish basic communication patterns between critical roles across sites and assign communication responsibilities to specific roles. In CGREN add appropriate relationship arcs to depict the flow of communication between each role, and attach applicable communication media to each of the relationship arcs. Decide how each communication path will be supported by technology. Assign communication icons to each path.
5. Determine the key artifacts that are to be created collaboratively, and model them along with each role's access and privileges. Include the applicable tooling/version control infrastructure, such as DOORS, RequisitePro, etc.
6. Revisit project-level organizational structures and ensure that all roles are assigned to specific sites.
7. Model specific elements of the requirements engineering process such as requirements elicitation or requirements prioritization processes, by mapping task-specific roles, artifacts, and communication mechanisms onto the previously identified sites.
8. Develop RGCN models based on the appropriate level of clarity and usability. For instance, individual RGCNs can be used to plan each requirements activity.

This process can be supported through the use of exemplar project templates from previous projects. Ideally the CGREN modeling exercise would be conducted as part of the kick-off event, but it can also be revisited throughout the project. One of the RA's in our study specifically mentioned that she saw the RGCN models as part of a *"living document."*

Refined CGREN Meta-model / Taxonomy

Another outcome of this study was the refinement of our meta-model and taxonomy. The CGREN taxonomy continues to focus on three primary entities of: *roles*, *sites*, and *artifacts*; as well as three general types of relations: *houses*, *accesses*, and *communicates*, that were observed between the entities. Refer to Figure 5.1 for the original meta-model representing the CGREN taxonomy.

As a result of the findings of our study we have extended the meta-model to support the concepts of communication volume, stakeholder who wear multiple hats, and elicitation techniques. We also adopted the OMG UML [19, 20] approach of modeling any association with attributes as a class. As depicted in Figure 6.2 the new meta-model uses classes to model *access* with associated *type* and *frequency* attributes, and the *communication* class with *frequency* and *media* attributes. The *frequency* attribute addresses our study participants' request to model the volume of communication between two roles. Three additional classes are added to the meta-model to depict the notion of elicitation techniques used with specific collaborative events. To this end, an event is modeled as a *collaboration* between *participants*. A collaboration is associated with meeting *type* (i.e. JAD, Storyboarding, etc), a meeting *name*, and an outlook-style schedule depicting actual meeting times and duration.

Each *participant* has a *role* in the meeting and each *collaboration* is assigned to a primary *site*. *Communication* and *participation* elements are represented as associations in instantiated models, while the *collaboration* type is modeled using one of the meeting type entities in Figure 6.3. To support the extended taxonomy, we also added an additional “many hats” icon, and introduced the visual notation that the width of the communication arc is approximately proportional to the estimated communication frequency. In addition, we introduced the icons shown in Figure 6.3 to represent a variety of elicitation techniques.

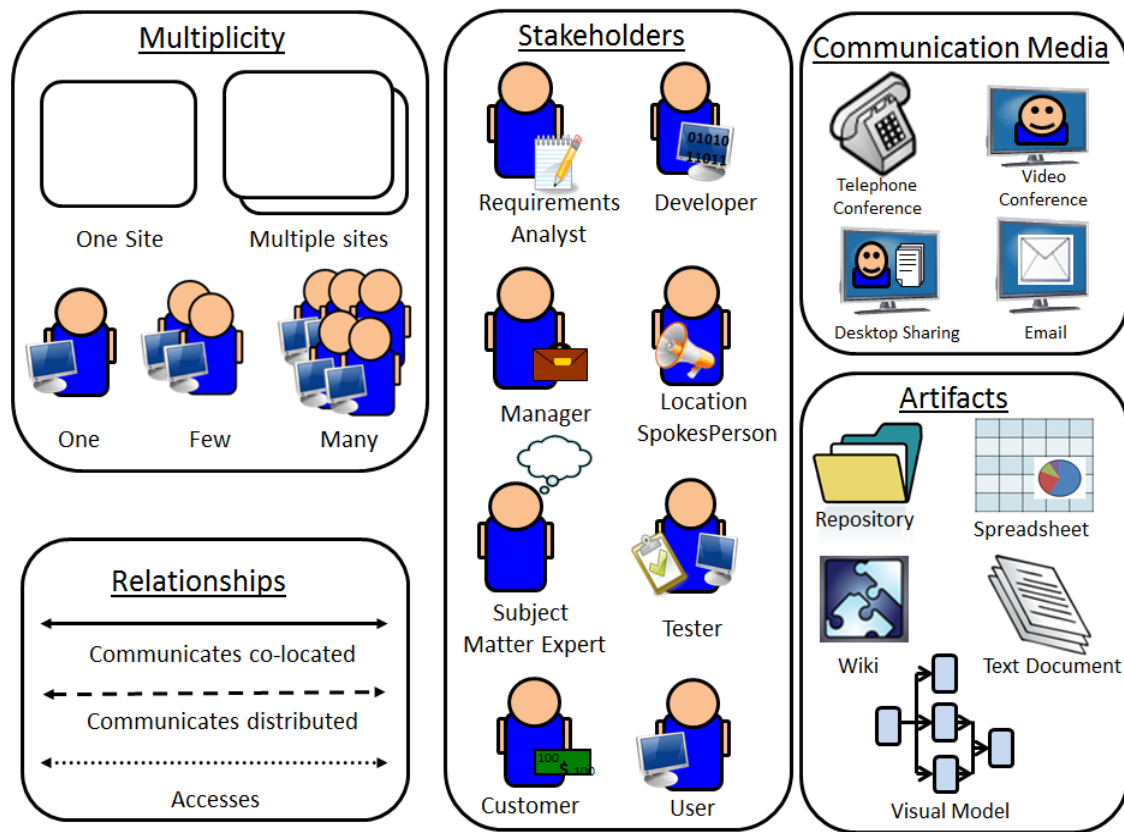


Figure 6.1 RE Sessions CGREN Glossary that study participants used to model their distributed requirements activities.

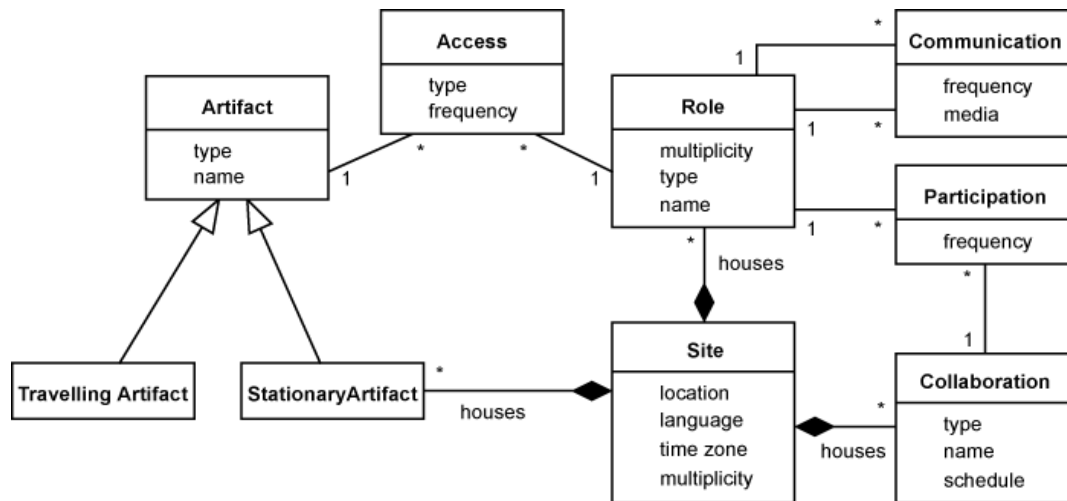


Figure 6.2 Updated Meta-model reflecting new concepts of communication frequency, multiple hats, and collaboration techniques used for requirements activities such as elicitation.



Figure 6.3 New Icons for Multiple Roles and Requirements Elicitation

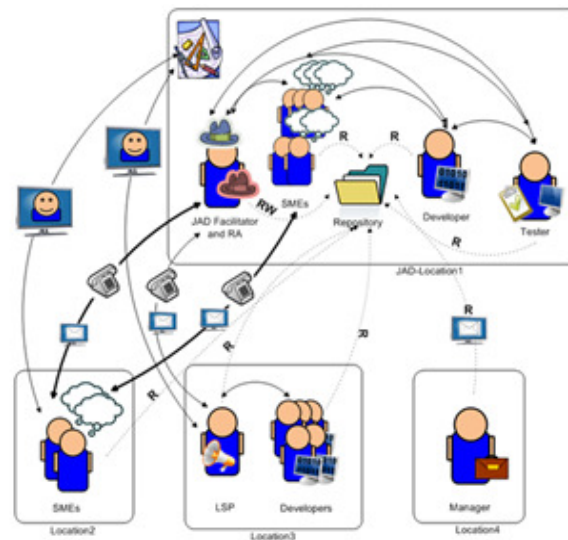


Figure 6.4 Utilizing the proposed new taxonomy and icons to model a Joint Application Design (JAD) session

Figure 6.4 provides an illustrated example of how the new taxonomy and related notation could be used to plan a globally distributed JAD session. In this session the JAD meeting is being organized at

Location-1 by a project stakeholder wearing dual hats of JAD Facilitator and RA. Many participants, including SMEs, a developer, and a tester all physically participate in the JAD session, while SMEs from Location 2 and an LSP from Location 3 participate remotely using video-conferencing. The Location2 LSP communicates with local developers if issues arise during the JAD session. Finally, a report is sent to the manager at Location 4 at the end of the session.

6.4 Conclusion

Each participant in our study was asked “what, if anything, did you gain from using CGREN?” RA1 stated that she gained “*A better understanding of the project (and a) better understanding of the stakeholders, the access they had, and ... their reach (impact in the project).*” Using the communication diagram (Figure 12) she identified a specific problem that occurred because of the distribution of the major stakeholders. In this case the lead developer was located in Knoxville, while most of the communication to establish requirements took place in Atlanta. As a result of modeling these interactions, the RA commented “*Wait a minute, all this communication is happening here (while) we have this one person who has to do all of these things, but they’re doing it remotely.*” She stated that if CGREN had been available to her earlier, this observation would have led to a restructuring of the communication patterns in the project.

RA2 noted that for their project “*the model is helpful for showing that ... in some of my locations I don’t really have a Spokesperson. And so there’s (sic) multiple SMEs that I’m going to... and (it is unclear) to what extent are they truly the authority.*” She also stated that as a result of modeling the stakeholder roles, this reinforced that it would be helpful for her to have a designated spokesperson for each site that would be responsible for identifying SMEs. She further commented that “*there’s multiple SMEs that I’m going to. And so that’s a lot of people I’m communicating with. ... I feel like it would be helpful to have fewer people and more people that were kind of designated as Spokespeople,*” which echoes the findings of Turner and Boehm that stress the importance of finding CRACK (Collaborative, Representative, Accountable, Committed, Knowledgeable) people during the requirements elicitation phase of project planning [91].

Finally, RA3 pointed out that CGREN would “*shed some light on what some of the possible constraints and limitations could be*” with respect to the current project configuration. In particular she pointed out that in her project all communication was via email, and that planning in advance would enable better infrastructure setup that could include video-conferencing technology and other techniques to support communication between stakeholders.

One of the key results of the modelling activity for the RAs was that the method and quantity of communication during the planning and execution phase of the project was highlighted. For example, RA1 commented “...*I never really noticed that I didn’t talk to the testers, even though they definitely wrote their test cases and complained sometimes about the way we wrote our requirements... after doing this <exercise> now I notice it.*” Both RA1 and RA2 noted that the exercise made them painfully aware of the complexity of their communication needs.

The study participants also pointed out that CGREN would be useful for planning resource allocation across distributed sites, and for supporting project post-mortem analysis which may be of use for future projects or for ongoing maintenance on the project at hand.

As researchers, we also noticed the paucity of different elicitation techniques used in the three projects. In all cases, the RAs relied on individual interviews and general group meetings either conducted in face-to-face meetings or using phone or video-conferencing technologies. There were no examples of more creative elicitation techniques such as Joint Application Design (JAD) sessions, creativity workshops, or even basic scenario-writing using storyboarding or other similar techniques [101]. As a result of this observation we noted that if CGREN were extended to include the notion of meeting types and/or elicitation techniques, it could serve to inspire and educate project stakeholders about new techniques, and encourage them to think beyond their previous planning experiences.

Chapter 7

Research Contributions

This research has resulted in (i) CGREN, a novel modeling technique that can be used to plan distributed requirements development projects, (ii) a base set of RGCN models that illustrate the general requirements development communication and collaboration paths and serve as templates for researchers and practitioners modeling their projects, (iii) guidelines for employing CGREN to create RGCNs, and (iv) organizational and activity patterns that capture successful strategies for distributed requirements development. Additionally we prototyped a web-enabled tool, GRETA, based on CGREN, to help practitioners in planning project-specific distributed requirements development processes.

These contributions are meant to help enhance distributed RE processes so that they support highly interactive, person-to-person relationships, namely elicitation, analysis, specification and management of requirements, across geographical distances.

Overall we recommend that practitioners proactively model their requirements engineering activities using our CGREN terminology and the recommendations described in our patterns, in order to identify and understand the issues as early as possible; and thereby avoid mistakes, as early in the development lifecycle as possible.

7.1 CGREN for creating RGCN Models

These RE studies show the importance of proactively identifying stakeholder roles, along with their locations, communication flows, critical project artifacts, and necessary tool support. Results suggest that projects which failed to consider these issues were impeded by problems such as disorganized stakeholder interactions, lack of appropriate tool support, data over-load, increased travel, and inefficient processes for supporting specific requirements engineering tasks. As a result of these findings we have developed a new taxonomy and associated meta-model for representing the primary components of a globally distributed requirements project. In addition to the meta-model we also introduce a new visual notation, CGREN that allows users to visually model their own globally distributed requirements engineering processes. The simplicity of the language makes it intuitive to understand, while still providing the level

of expressivity needed to model a wide variety of projects. The notation fills a gap in the field of distributed RE, because existing modeling notations fail to provide all of the concepts needed to capture the concepts of site, roles, communication, and shared resources needed to model most distributed RE projects.

The new meta-model and associated visual notation provides several benefits. From an industrial perspective, it enables project managers to plan, analyze, and optimize their distributed requirements engineering processes, so that they can understand their existing processes, identify weaknesses and problems, and establish improved processes and appropriate supporting infrastructure. From a research perspective it provides a common language for modeling distributed requirements projects and activities, and thereby facilitates comparisons across projects [18]. These comparisons make it possible to identify recurring patterns of collaboration, common obstacles, and best practices used for collaborative requirements engineering activities. Such observations enable researchers to propose new techniques or improve existing methods to handle the specific challenges of global requirements processes.

7.2 Patterns

Though additional studies need to be conducted before the organizational and activity patterns can be more formally defined; we have identified several candidate patterns that have contributed to project success in several of the studied projects.

7.3 GRETA

We have applied the lessons learned from these RE studies to the design and prototype of GRETA, a web-based global requirements engineering tool/application that can assist practitioners in planning and executing the RE phase of their projects. GRETA will be the software version of the modeling study described in Chapter 6. With GRETA users will be able to generate RGCNs that illustrate their own project specific distributed RE processes and activities.

Chapter 8

Future Research Opportunities

An initiative that encourages RE researchers and professionals to reach out to industry with training and networking opportunities to share best practices should be considered. Opportunities for future research also include extending the study of distributed RE through continued testing in industrial settings, across a more extensive set of projects. The study can also be expanded to include a broader set of project stakeholders such as SMEs and developers. Although successful practices have already emerged as a result of our studies, an extended study could be designed to identify and document a more extensive set of patterns and anti-patterns for successful global requirements engineering processes. For example, extend the study of distributed RE projects:

- From Patterns 5.3.1 RA Title, 5.3.3 Location Spokesperson Role and From Pattern 5.3.9 Multipurpose Requirements Specification: Interview additional project stakeholders, i.e. members of the technical team and project management, an in-depth study of LSPs and their responsibilities, etc. across a more extensive set of projects and industries. Organizational artifacts including project plans and organization charts can be analyzed to further study the roles and relationships that are involved in distributed RE projects.
- From Patterns 5.3.2 Telephone RA's Preferred Communication Tool and 5.3.4 Requirements Engineering Tools and Technologies: A more in-depth study of the communication and collaboration tools and technologies. This research identifies classes of tools that could be used to support each of the primary requirements engineering activities, and describes a process and associated strategies that practitioners can use to help select a suitable set of tools to support collaborative distributed requirements engineering tasks. Ongoing research would include identifying organizations and RAs interested in enacting and evaluating the proposed guidelines.
- From Pattern 5.3.5 Face-to-Face Communications: Follow up with RAs to gain insight into project kickoff process, especially if handled by another project leadership role.

- From Pattern 5.3.8 Distributed Stakeholder Communication: Delve into the topics of conversation and interaction, i.e. project 3 Appears that distributed technical/SMEs communicated with each other, but not stakeholder. Could it be that the tech people were not allowed to talk to the end-users?

Continued GRETA Development

We have applied the lessons learned from our distributed RE studies to the invention of GRETA, a web-based global requirements engineering tool/application that can assist practitioners in planning and executing the RE phase of their projects. GRETA users will be able to model their requirement development activities such as communication paths, elicitation, creating specs and managing requirements; while GRETA analyzes an underlying graph representation of said model to identify potential problems. We will continue to augment GRETA with new and modified icons. From our preliminary results GRETA can be enhanced to trigger recommendations to RAs if it notices a bad practice. via a Recommendation Report.

Appendix A

Data Collection Tools

Opens Source Software Projects Research

OSSP Admin Survey

For the following question, select ONLY ONE RESPONSE:

1. **Which of these open source project development models do you most closely follow?**
 - ☐ Vendor-managed development, where code is primarily written in-house.
 - ☐ Community-managed development, where code is primarily written by members of the OSSP community.
 - ☐ Other (please explain)

For each of the following questions, select ALL RESPONSES that apply:

2. **How do you elicit and gather feature requests for new functionality?**
 - ☐ <OSO> Forum @ <OSO project forum URL>
 - ☐ Online Newsgroups
 - ☐ Invited Groups, such as Google, Yahoo, etc.
 - ☐ Via Email
 - ☐ Other
3. **How is functionality added to your product?**
 - ☐ In planned releases.
 - ☐ Continuously as developers complete and add new features.
 - ☐ Continuously by our team. Contributing developers submit their updates and we deploy them in staged releases.
 - ☐ Other (please explain)

For each of the following questions, select ONLY ONE RESPONSE:

4. **Do you review feature requests from your user community?**
 - ☐ Yes, if so please respond to question #5
 - ☐ No
5. **Do you consider all feature requests equally?**
 - ☐ Yes
 - ☐ No (please explain)

6. Please **SELECT** the **IMPORTANCE** of each of these **FACTORS** in prioritizing feature requests for your product.

	Very Important	Somewhat Important	Not Important
Formal voting, by available voting mechanism in the forum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
User discussions in the forum	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Face-to-face meetings conducted with user groups.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Emails from users.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
In-house staff members self-select features to build.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Other (please explain)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

For each of the following questions, select **ALL RESPONSES** that apply:

7. **Who decides which new feature requests to implement in a given release?**

- ☐ A single person (i.e. release manager, project manager)
- ☐ A team of people (please explain)
- ☐ Other (please explain)

For each of the following questions, select **ONLY ONE RESPONSE**:

8. **Do you have a process in which you extract information from the users' feature requests and transform them into more formal requirements?**

- ☐ No
- ☐ Yes (please explain)

9. **How satisfied are you with your process for gathering feature requests?**

- ☐ Very Satisfied
- ☐ Somewhat Satisfied
- ☐ Somewhat Dissatisfied
- ☐ Very Dissatisfied
- ☐ Comments

10. **How would you improve your process for gathering feature requests? (Please describe)**

11. **How satisfied are you with your process for prioritizing feature requests?**

- ☐ Very Satisfied
- ☐ Somewhat Satisfied
- ☐ Somewhat Dissatisfied
- ☐ Very Dissatisfied
- ☐ Comments

12. **How would you improve your process for prioritizing feature requests? (Please describe)**

13. **How satisfied do you think your users are with your processes for gathering and prioritizing feature requests?**

- ☐ Very Satisfied
- ☐ Somewhat Satisfied
- ☐ Somewhat Dissatisfied

- Very Dissatisfied
- Don't Know
- Comments

OSSP User Survey

For the following question, select ONLY ONE RESPONSE:

1. Primary role in OSSP community?

- Current user of product
- Prospective user of product
- Current provider of product
- Prospective provider of product

2. Ever contributed to OSSP code base?

- No
- Yes

For each of the following questions, select ALL RESPONSES that apply:

3. How do you request new features and new functionality?

- <OSO> Forum @ <OSO project forum URL>
- Online Newsgroups
- Invited Groups, such as Google, Yahoo, etc.
- Via Email
- Other

4. Which of the following methods do you think your OSSP uses to prioritize feature requests?

- Formal voting, by available voting mechanism in the forum.
- User discussions in the forum
- Administrators decide which features to build without user input
- Don't know
- Other (please explain)

For the following questions, select ONLY ONE RESPONSE:

5. Who do you think decides which new feature requests to implement in a given release?

- A single person (i.e. release manager, project manager)
- A team of people (please explain)
- Other (please explain)

6. On average, how often do you access the OSSP forums?

- Daily
- Weekly
- Monthly

7. How many feature requests have you made in the last 6 months?

- 0 requests
- 1 request
- 2-5 requests
- 6-10 requests
- More than 11

For the following questions, select ONLY ONE RESPONSE:

- 8. Which of these scenarios most closely resembles how you interact with the <OSSP> forums to request new features?**
- ☐ I log in to the forum and type in my new feature request.
 - ☐ I log in to the forum and search to see if my feature request has already been posted by somebody. If I find a similar request, I do nothing.
 - ☐ I log in to the forum and search to see if my feature request has already been posted by somebody. If I find a similar request, I demonstrate my support for it by registering a vote or adding a supportive comment.
 - ☐ Other (please explain)
- 9. How satisfied are you that your feature requests for new functionality are addressed by this process?**
- ☐ Very Satisfied
 - ☐ Somewhat Satisfied
 - ☐ Somewhat Dissatisfied
 - ☐ Very Dissatisfied
 - ☐ Comments
- 10. How would you improve the process for requesting new features and functionality? (Please describe)**

RE Industry Interview Questions

C2. RE Industry Interview Questions

Participant Demographics

Name

Company Name

Industry

Number of years you have worked for this company

Number of years that you have worked in this department / reported to the same senior manager(s)?

Your number of years of professional IT experience

Total number of years you have worked with and/or supported multiple-sites/locations

Project Demographics

1-Project Name

2-Please describe your responsibilities for the project.

3-Was this a new application/product/solution/system or an enhancement to a current application/product/solution/system?

4-Briefly describe the project in terms of functionality, domain, etc.

5-Is the project In progress, Completed, On hold, Canceled, Planned?

6-Approximate the project cost in

a) Dollars

b) Hours (conception to implementation)

7-How was this project originally scoped? Was there a central vision before you started gathering the requirements? Was there a clear vision statement?

Requirements

8-How did you elicit and gather requirements?

9-When is technology critical? When are you forced to use technology?

10-When doesn't technology work? Why?

11-How are requirements documented? Do you use an automated tool, requirements database, spreadsheet, document?

12-Who, in terms of role, is responsible for entering the data entered into the requirements database, spreadsheet, document?

13-As of today, approximately how many requirements have been generated for this entire project?

14-How many hours, in total, do you think were spent gathering these requirements?

15-Do you or your team produce a Requirements Specification document?

If so, who's responsible for designing and structuring it?

If not, how are requirements documented?

16-How were stakeholders chosen for this project?

Do you feel that any other project representation is missing?

17-About how many stakeholders were directly involved in contributing requirements to the project?

18-How do individual stakeholders participate in the requirements development and documentation process? Can you identify specific stakeholder roles?

For instance is there a stakeholder that interviews the other project participants at their site?

19-How many geographic locations and stakeholders participate in this project?

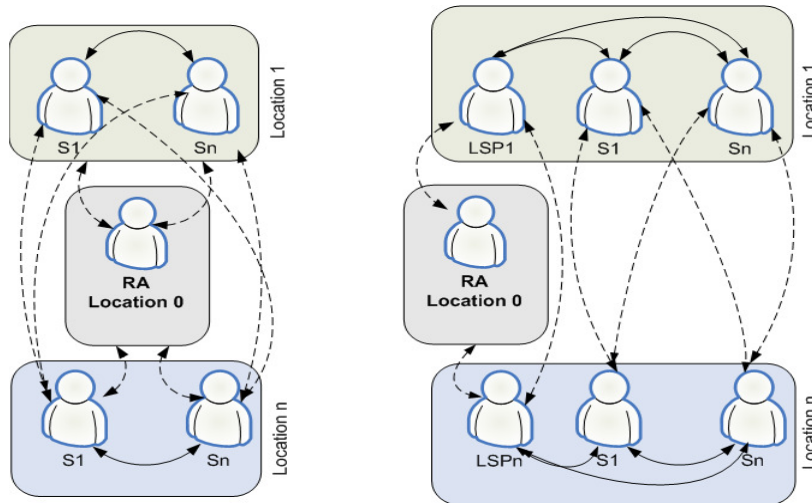
20 - Are you and the participating stakeholders in a conference room or at your own desks for the requirements gathering sessions?

Requirements Elicitation Process

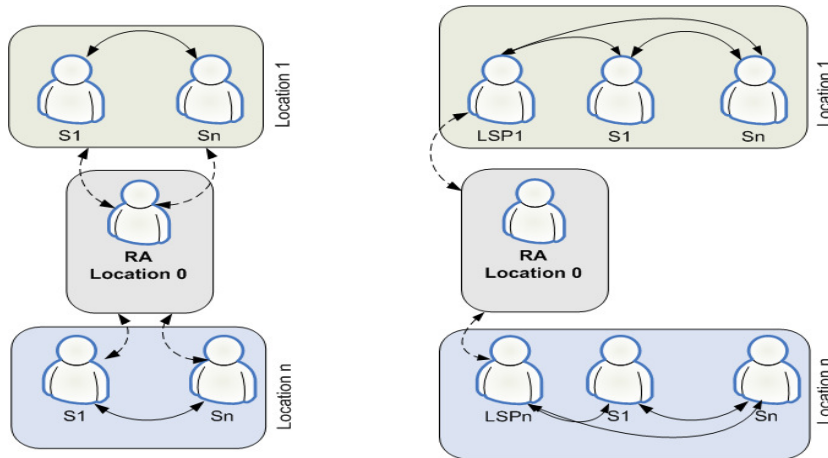
21-Use the following models as a starting reference to describing the organization structure that most closely resembles your project?

Legend: RA = Requirements Analyst, S=Stakeholder, LSP=Location Spokesperson

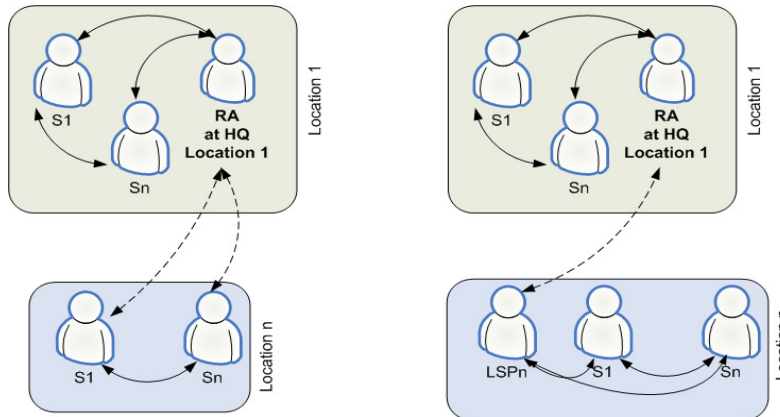
○ Teams-Collaboration



○ **Facilitated-Collaboration**



○ **RA co-located with S**



Requirements Management

22-Did you experience or identify any stakeholder/requirement conflicts?

23-How were the conflicts handled?

24-How were requirements prioritized?

25-Describe the overall success of this requirements gathering process, including tools.

RE Modeling Session Exit Survey Questions

1. How useful was the modeling notation?
 - a. Were you able to model all of the concepts from your project?
 - b. Any problems using the graphical symbols / are there any concepts missing?
 - c. What was easy to model?
 - d. What did you find difficult?
 - e. Were the stakeholder types sufficient? Enough roles? Was it helpful to differentiate the roles in this way?
 - f. How well did the concept of one-few-many help you?
2. What, if anything, did you gain from using this modeling notation?
 - a. Though this is a past project, did or would the models help you identify any potential issues?
3. Could this modeling technique be used for modeling other requirements engineering and management tasks and activities?
4. If we developed software that allowed you to accomplish this modeling, using drag-and-drop functionality, would you use it during your next project?
 - a. At what phase of the project?
 - b. What if we provided a tool kit, with magnetic icons that you could use the white board?
5. Can you think of anything else that you would add to the modeling language that would be helpful?

Appendix B

RE Research Interviews Database Design

RE Interview database tables and field definitions

Field Name	Data Type	Description
ProjNum	Number	Project number
Industry	Text	Organization's industry
RATitle	Text	RA's title in organization
TotalSites	Number	Total number of sites participating in this project
Total LSPs	Number	Total number of location spokespersons participating in this project
TotalStakeholders	Number	Total number of stakeholders participating in this project, including RA(s) and LSP(s)

Table B.1 REProjInfo: Project metadata

Field Name	Data Type	Description
ProjNum	Number	Project number
SiteNum	Number	Count of Site records
SiteName	Text	Name used to designate site, i.e. USA1, Europe3, Asia2
TotalRAs	Number	Total number of RAs at site
Total LSPs	Number	Total number of LSPs at site
LSPTitle	Text	LSP's Title
TotalStakeholders	Number	Total number of stakeholders participating in this project, excluding RA and LSP

Table B.2 RESites: Site-specific metadata

Field Name	Data Type	Description
ProjNum	Number	Project number
CommAct	Memo	Description of the Communication Activity
CommQte	Memo	Quotes about Communication Activity
ElicitAct	Memo	Description of the Elicitation Activity
ElicitQte	Memo	Quotes about Elicitation Activity
AnalysisAct	Memo	Description of the Analysis and Prioritization Activity
AnalysisQte	Memo	Quotes about Analysis and Prioritization Activity
ConflictAct	Memo	Description of any conflict and conflict resolution activities
ConflictQte	Memo	Quotes about conflict and conflict resolution activities
ManageAct	Memo	Description of the Requirements Management Activity
ManageQte	Memo	Quotes about Requirements Management Activity
SpecsAct	Memo	Description of the Specification Activity
SpecsQte	Memo	Quotes about Specification Activity
SuccessAct	Memo	Description of the project success
SuccessQte	Memo	Quotes about project success
ToolQte	Memo	Quotes about software tools
RACHalls	Memo	Description of challenge and/or pain point RA encountered
RACHallsQte	Memo	Quotes about challenge and/or pain point RA encountered
RADuties	Memo	All of the RA's project responsibilities

Table B.3 REInterviews: Description of RE activities and RA quotes

RE Interview database queries

Query used to produce Table 5.3

SELECT

REProjInfo.ProjNum, REProjInfo.Industry,

REProjInfo.TotalSites, (REProjInfo.TotalStakeholders-1) AS {"Stakeholders"}

FROM REProjInfo;

Pattern-Phone-Conf			
ProjNum	Industry	TotalSites	"Stakeholders"
1	Financial Services	2	6
2	Telecommunications	3	5
3	Software Requirements Consulting	4	39
4	Software Requirements Consulting	4	69
5	Video Games	8	45
6	Retail	2	8
7	Software Engineering Research	5	25
8	Corporate Research	3	37
9	Information Technology	6	51
10	Securities Software Solutions Integrators	3	110

LSP Query 1 used to produce Table 5.4

SELECT

RESites.ProjNum, REProjInfo.Industry,
RESites.SiteName, RESites.TotalRAs,
RESites.TotalLSPs, RESites.TotalStakeholders

FROM RESites INNER JOIN REProjInfo ON RESites.ProjNum = REProjInfo.ProjNum

WHERE (((REProjInfo.TotalLSPs)<>0))

ORDER BY REProjInfo.ProjNum ASC, RESites.SiteNum ASC;

LSP-Q1						
ProjNum	Industry	SiteName	TotalRAs	TotalLSPs	TotalStakeholders	
1	Financial Services	USA0	1	0	0	
1	Financial Services	USA1	0	1	2	
3	Software Requirements Consulting	Asia1	1	1	29	
3	Software Requirements Consulting	Europe1	0	0	2	
3	Software Requirements Consulting	USA1	0	0	4	
3	Software Requirements Consulting	USA2	0	0	3	
4	Software Requirements Consulting	USA0	1	0	0	
4	Software Requirements Consulting	USA1	0	0	15	
4	Software Requirements Consulting	USA2	0	0	20	
4	Software Requirements Consulting	Asia1	0	1	30	
5	Video Games	USA1	1	0	0	
5	Video Games	USA1A	0	3	8	
5	Video Games	USA1B	0	2	4	
5	Video Games	USA2	0	2	5	
5	Video Games	USA3	0	2	3	
5	Video Games	USA1C	0	2	3	
5	Video Games	USA4	0	2	3	
5	Video Games	Europe1	0	2	4	
6	Retail	USA1	1	0	2	
6	Retail	USA2	0	1	5	

LSP Query 2 database information

SELECT

```

REInterviews.ProjNum,
REProjInfo.TotalLSPs,
REInterviews.CommAct, REInterviews.CommQte,
REInterviews.ElicitAct, REInterviews.ElicitQte,
REInterviews.AnalysisAct, REInterviews.AnalysisQte,
REInterviews.ConflictAct, REInterviews.ConflictQte,
REInterviews.ManageAct, REInterviews.ManageQte,
REInterviews.SpecsAct, REInterviews.SpecsQte,
REInterviews.SuccessAct, REInterviews.SuccessQte,
REInterviews.RAChalls, REInterviews.RAChallsQte

```

```

FROM REInterviews INNER JOIN REProjInfo ON REInterviews.ProjNum = REProjInfo.ProjNum

```

```

WHERE (((REProjInfo.TotalLSPs)<>0))

```

```

ORDER BY REInterviews.ProjNum;

```

ProjNum	TotalLSPs	CommAct	CommQte	ElicitAct	ElicitQte	AnalysisAct	AnalysisQte
1	1	1 For the project discussed our RA had to gather requirements from only one remote location. The RA communicated and collaborated with an LSP who was one of 3 stakeholders at the customer location. The RA noted that the telephone was the only communication tool needed since this was a small project. For the		RA organized and led teleconferences for eliciting and gathering requirements. LSP and other two stakeholders were on conference call. RA communicated mostly with		During analysis, prioritization of requirements occurred via telephone conference. The prioritization criterion was customer's "data needed by a specific date."	
3	1	1 For this project the RA had to collect requirements from stakeholders at four locations, 2 in the US, 1 in Asia and 1 in Europe. Approximately 50 stakeholders were directly involved in contributing requirements. Not only were Asian stakeholders SMEs, they would also become the end-users of the solution.		RA preference was to lead face-to-face, one-on-one and small group interviews. When that was not possible, some interviews were conducted via teleconferences and a few via emails.	Typically the RA "spent several weeks in the Asian location and then a few weeks in the US. If we happened to be in the US at the same time	Requirements were prioritized by business objectives and then release date. Requirements were prioritized via face-to-face	According to the RA "When we first started the project they (the stakeholders) started off just ranking them one
4	1	1 The RA communicated with stakeholders at 3 locations. There were approximately 15 SMEs at the US1 location, 15-20 SMEs at the US2 location and, 25-30 SMEs at the Asian location. For this project there were 2 RAs: our interviewee, the project RA who led the requirements gathering effort and communicated mostly with the SMEs	Per the RA all of the SMEs "would have been almost unanimously across the board very technical people who would	The RA led telephone conferences when she could not meet with stakeholders face-to-face. Face-to-face or phone conference participation by the stakeholders was dependent	The RA shared, "My personal preference is always to do the elicitation in person, when that was practical and possible. For that reason, we	SMEs prioritize requirements by business justification. Project RA facilitates the prioritization process.	According to the RA "Initially the requirements were not prioritized. They were prioritized by whoever had the most political clou
5	15	15 RA held face-to-face meetings with the USA1 and USA2 since they were all co-located at the same site. The meetings turned into telephone conferences once USA3 joined. Telephone conferencing worked well, especially as an mp3 recording of the actual meeting was attached to the distributed meeting notes; so that even the	"We started out with telephone and later moved to video conferences." Though this was not an assigned task,	The project team employed an agile methodology and performed a modified Scrum. RA elicited and gathered requirements using a wiki. All of the user requests (i.e. bugs, feature requests) were	Per RA, "We used modified Scrum. We did iterative [development] and then interacted with the customer to get their feature	The requirements list from the project wiki was the one presented to the stakeholders for prioritizing. The RA and development team would just negotiate for the current iteration.	"I have Team A requesting the same 5 features, Team B requesting 5 features; I can only do 4 features. What do I do? That idea
6	1	1 Initially the RA communicated with 2 co-located stakeholders at USA1 and 5-6 stakeholders at USA 2. The stakeholder roles at USA1 were user experience and project manager, and stakeholder roles at USA2 were website	Initially - Per the RA, "I met with the stakeholders, most input seem to come from user	The RA met with the co-located user experience stakeholder face-to-face; and the product management stakeholder for USA2 was also	The RA mentioned that, "there was some communications that I was not privy to... I would get it through	Prioritization was based on what could be implemented for the holiday season, prior to the end of year system lockdown.	"Originally the requirements coming from product management did have that scoping

Sample queries used to produce Table 5.5

Spreadsheet Query

SELECT

REInterviews.ProjNum, REInterviews.ConflictQte,
REInterviews.ManageAct, REInterviews.ManageQte,
REInterviews.SpecsAct, REInterviews.SpecsQte,
REInterviews.SuccessAct, REInterviews.ToolQte,
REInterviews.RACHalls, REInterviews.RACHallsQte

FROM REInterviews

WHERE

((REInterviews.ConflictQte) Like '*spreadsheet*')) OR (((REInterviews.ManageAct) Like '*spreadsheet*')) OR (((REInterviews.ManageQte) Like '*spreadsheet*')) OR
(((REInterviews.SpecsAct) Like '*spreadsheet*')) OR (((REInterviews.SpecsQte) Like '*spreadsheet*')) OR (((REInterviews.SuccessAct) Like '*spreadsheet*')) OR
(((REInterviews.ToolQte) Like '*spreadsheet*')) OR (((REInterviews.RACHalls) Like '*spreadsheet*')) OR (((REInterviews.RACHallsQte) Like '*spreadsheet*'))

ORDER BY REInterviews.ProjNum;

ProjNum	ConflictQte	ManageAct	ManageQte	SpecsAct	SpecsQte	SuccessAct	ToolQte	RACHalls	RACHallsQte
9	As for conflicts the RA explained, "...there's always customers who want it to do one thing and then customers who want it to do another. One of the difficulties is	As already described, the requirements were managed by the RA and his organization using Rational Team Concert.		The RA took notes during the requirements elicitation and gathering meetings which he later transcribed into MS Word. These meeting reports were then stored in a SharePoint	When asked how the requirements were documented, the RA responded, "First of all...for any specific interview we will write out a summary of the conversation.	The RA felt that as long as he and his team can somehow communicate with the customers, they can build a relationship. The RA continued, "The other side of that is,	"We use the IBM tool which is called SameTime Unite <for web conferencing>. In the past I've also used WebEx." When asked if he remembers DOORs		
10	And yes, I could see if the project were a little bit more complicated, NetMeeting where things were white-boarded right away, would have worked	The RA maintained the master copy of the requirements artifacts in a folder in her email box.	"And of course I copied pretty much the world for almost everything. That's another reason why it would've been better	The requirement specification consisted of MS Word documents, spreadsheets and user interface and use case diagrams that were created	When asked if it was difficult using email to keep track of the documentation the RA replied, "not really. It may have worked out a little bit better if it	The RA felt that overall the requirements gathering and management process and tools were successful	The RA did not hold video or web conferences because "<the other project stakeholders> didn't really have that set up on their end, and we didn't have		

Wiki Query

SELECT

REInterviews.ProjNum, REInterviews.ConflictQte,
 REInterviews.ManageAct, REInterviews.ManageQte,
 REInterviews.SpecsAct, REInterviews.SpecsQte,
 REInterviews.SuccessAct, REInterviews.ToolQte,
 REInterviews.RACHalls, REInterviews.RACHallsQte

FROM REInterviews

WHERE

((REInterviews.ConflictQte) Like '*wiki*')) OR (((REInterviews.ManageAct) Like '*wiki*'))
 OR (((REInterviews.ManageQte) Like '*wiki*')) OR (((REInterviews.SpecsAct) Like '*wiki*'))
 OR (((REInterviews.SpecsQte) Like '*wiki*')) OR (((REInterviews.SuccessAct) Like '*wiki*'))
 OR (((REInterviews.ToolQte) Like '*wiki*')) OR (((REInterviews.RACHalls) Like '*wiki*')) OR
 (((REInterviews.RACHallsQte) Like '*wiki*'))

ORDER BY REInterviews.ProjNum;

RETool-Wiki-quotes									
ProjNum	ConflictQte	ManageAct	ManageQte	SpecsAct	SpecsQte	SuccessAct	ToolQte	RACHalls	RACHallsQte
2	According to the RA the hope was that eventually all of the products would contain the basic camera application. The first	As previously discussed, the RA created and managed the project specifications and artifacts.	"<My project documentation> just gets posted to this internal internet site, and then I'm done; once everyone agrees upon the	RA authored the project documentation which included flow diagrams, state tables, requirements, wire frames, etc. The project specs			When asked about the different operating systems, the RA shared, "Mac is something that's very new to my company.		
5	Though video conferences allowed the stakeholders to see each other; "it is absolutely the worst thing for conflict" according to	By the time USA3 joined the project the RA noted that "the problem came in with them because the teams didn't know each other at	When asked if they were following a documented process the RA responded "...we had started to formalize the process. "One of the	The aforementioned requirements and requests, project documentation and artifacts were maintained in a project wiki	The stakeholders could "post requests, screenshots, mockups, and priorities, whatever. They could also see our specifications,		Pain point of commercial products-1: RA had considered using a commercial product called DevTrack Pro but, "It was a complete	Pain point: Dissemination of information when RA was not included in stakeholder communications. During these	
6		As described above, the RA maintained the "official" copy of the		RA documented requirements in MS Word and Use Cases	Per the RA, "In most cases I would get (documentati		Pain point of web conferencing When asked stakeholders'	Pain point of using email to manage requirements docs:	

Tools Query database information

SELECT

REInterviews.ProjNum, REInterviews.ToolQte,
 REInterviews.CommQte, REInterviews.ElicitQte,
 REInterviews.ManageQte, REInterviews.SuccessQte

FROM REInterviews

ORDER BY REInterviews.ProjNum;

Pattern-RE-Tool					
P	ToolQte	CommQte	ElicitQte	ManageQte	SuccessQte
2	When asked about the different operating systems, the RA shared, "Mac is something that's very new to my company. We've typically always used PCs... We started moving to a new developing platform and the first team that started that work effort	"My company is really big on NetMeetings as well. Unfortunately I use a Macintosh at work and NetMeeting doesn't work with a Macintosh. It's created some problems for us. But we usually get around it by sending the document ahead of time and just calling out a	As the requirements gathering process continued, "Usually I had mockups or screenshots that we had created to help talk through them...We do have design documents that are detailed specs but it's easier when you're dealing with people who are more visual, it's easier to show them a picture and talk about it."	"<My project documentation> just gets posted to this internal internet site, and then I'm done; once everyone agrees upon the wireframe and the requirements pdf. There are additional people that create other documents that are more Wiki or TWiki based where there's a list of product	
3	"At one point we started using a requirements management tool -CaliberRM- but it was so far into the project that it would have required significant work to move everything over to the tool so we dropped it. I think it was more that we were really well-		Typically the RA "spent several weeks in the Asian location and then a few weeks in the US. If we happened to be in the US at the same time we would go in person. If we happened to be in different places we would do a phone call."	Though the RA preferred meeting with stakeholders face-to-face, when that hadn't been possible on other projects, they used GoTo Meeting, "that works pretty well because you can actually put the documents in front of people and have them look at it while you're looking at it. I've used video	
4	In response to, if it's not in person you use the phone conference, is that your preference? Is there some other method where if you can't do it face to face is there another tool or process you prefer to use?, the RA responded, "My first choice is	Per the RA all of the SMEs "would have been almost unanimously across the board very technical people who would specialize in one or more areas of factory automation or factory management. Factory systems management.... But in general, you have experts in	The RA described the dynamics of The RA shared, "My personal preference is always to do the elicitation in person, when that was practical and possible. For that reason, we used to travel to and from Asia, and because I had the luxury of having a large team, we could have had about six or seven people at any given point of time. We typically attempted to do one on one		
5	Pain point of commercial products-1: RA had considered using a commercial product called DevTrack Pro but, "It was a complete disaster. At the end of the day people are human and not ticket agents....So it's very much a ticketing system...and	"We started out with telephone and later moved to video conferences." Though this was not an assigned task, "For almost every team <of stakeholders> that got added on, there was one buddy team or mentoring team, to help and train them: not only on the software	Per RA, "We used modified Scrum. We did iterative [development] and then interacted with the customer to get their feature requirements."	When asked if they were following a documented process the RA responded "...we had started to formalize the process. "One of the interesting things was USA3 actually had a software process that they'd been using for years... Well we'd been tweaking and adapting ours, because when there's a	"How do you know you're successful? You work it back going to test it then the customer test it. How are you going to know we are successful? That's part requirements gathering. We have communication of what it is... give you the feature. How do
6	Pain point of web conferencing When asked stakeholders' locations for web conferences, RA stated "Requirements gathering	Initially - Per the RA, "I met with the stakeholders, most input seem to come from user experience."	The RA mentioned that, "there was some communications that I was not privy to... I would get it through the spokesperson."		

Face-2-Face Query used to produce Table 5.8

SELECT DISTINCT

RESites.ProjNum, REInterviews.CommAct,
REInterviews.CommQte, REInterviews.AnalysisAct,
REInterviews.AnalysisQte

FROM RESites INNER JOIN REInterviews ON RESites.ProjNum = REInterviews.ProjNum

WHERE

((RESites.SiteName) Like '*0*') OR ((REInterviews.CommAct) Like '*face*')) OR
(((REInterviews.CommQte) Like '*face*')) OR (((REInterviews.AnalysisAct) Like '*face*')) OR
(((REInterviews.AnalysisQte) Like '*face*'))

ORDER BY RESites.ProjNum;

Pattern-Face2Face				
ProjNum	CommAct	CommQte	AnalysisAct	AnalysisQte
1	For the project discussed our RA had to gather requirements from only one remote location. The RA communicated and collaborated with		During analysis, prioritization of requirements occurred	
2	The RA communicated with stakeholders via telephone conferences. Stakeholders usually assembled in a conference room at their	"My company is really big on NetMeetings as well. Unfortunately I	In summary the requirements are prioritized by P0 priority	The RA explained that the requirements are assigned priority ratings in the
3	For this project the RA had to collect requirements from stakeholders at four locations, 2 in the US, 1 in Asia and 1 in		Requirements were prioritized by business objectives and then	According to the RA, "When we first started the project they (the stakeholders)
4	The RA communicated with stakeholders at 3 locations. There were approximately 15 SMEs at the US1 location, 15-20 SMEs at the US2	Per the RA all of the SMEs "would have been almost unanimously	SMEs prioritize requirements by business justification.	According to the RA, "Initially the requirements were not prioritized. They were
5	RA held face-to-face meetings with the USA1 and USA2 since they were all co-located at the same site. The meetings turned into	"We started out with telephone and later moved to video	The requirements list from the project wiki was the one presented to the	"I have Team A requesting the same 5 features, Team B requesting 5 features; I can
7	The RA's team consisted of the RA and 2 others who traveled to the five sites. They mostly conducted face-to-face interviews	Telephone conferences were used for status meetings and follow-	In the beginning the requirements were prioritized by the most	"In the first iteration we didn't prioritize them I believe...we did it on a use
8	According to the RA she and her team were not the only project stakeholders performing RE tasks. The RA's team trained domain	"Once we trained their staff some of the people, some of the	The requirements were prioritized, "Primarily based on delivery dates.	
9	Typically the RA communicated with stakeholders via telephone conferences. The RA preferred to have as many LSPs as possible	"When we have support conversations that require highly	According to the RA the requirements were prioritized, "based on	When asked about the customer's involvement in the prioritization process,
10	RA and her team communicated with the client representative mainly via telephone conferences.		In response to the question of how the requirements were	

Email-Specs Query used to produce Table 5.9

SELECT

REInterviews.ProjNum, REInterviews.SpecsAct, REInterviews.SpecsQte

FROM REInterviews

WHERE

((REInterviews.SpecsAct) Like '*email*')) OR (((REInterviews.SpecsQte) Like '*email*'))

ORDER BY REInterviews.ProjNum;

Email-Specs-Pattern		
ProjNum	SpecsAct	SpecsQte
1	RA created and maintained requirements documentation in MS Word. RA would email the specification to the stakeholders for their review.	According to the RA, "I didn't use a requirements database since this project was a small scale change to an existing platform. When it's less than two dozen I just kind of
2	RA authored the project documentation which included flow diagrams, state tables, requirements, wire frames, etc. The project specs were converted to pdf format using OmniGraffle, basically a	
3	RA and other product managers on the team created and maintained the requirements documentation. RA modeled the requirements mostly using pictures because some of the stakeholders did not speak	RA shared documentation with stakeholders by emailing the docs or links for downloading the docs. "In some cases, especially in Asian location, it was better to print them out so
4	Initially an RA or RE (RA/E) would draft a document and then send it to the librarian for safekeeping, and to the necessary stakeholders for their review and approval. Drafts of the documentation were shared	In answer to this follow questions, When several people were gathering the requirements, then would all of you get together on the phone and discuss the
6	RA documented requirements in MS Word and Use Cases were created using Visio. Prototypes were an appendix to the Use Cases. Wire Frames illustrating main and alternate scenarios would follow the use	Per the RA, "In most cases I would get (documentation updates) through email. So (the stakeholders) would either put their comments in the body of the email message
7	At the beginning of the project the RA and his team created text documents and an Access database as they gathered the requirements. "When we were doing interviews we basically met and consolidated	"We had a couple of tables with who said the requirements, what types of requirements, how many times has it been stated, and so on. We used that in the beginning and we
8	The responsibility for inputting the requirements in DOORs was shared by the RA and her team, the business analysts, trained SMEs and, "in some cases the analysts at the sub-contractors site."	As the RA explained, "but there was more that wasn't in DOORs. And this caused problems initially because there were things, and vice versa, there were things in the Doors database
10	The requirement specification consisted of MS Word	When asked if it was difficult using email to

Stakeholder-Communication Query used to produce Table 5.10

SELECT

REInterviews.ProjNum, REInterviews.CommAct,

REInterviews.CommQte, REInterviews.ElicitAct, REInterviews.ElicitQte

FROM REInterviews

WHERE

((REInterviews.CommAct) Like '*each other*')) OR

((REInterviews.CommQte) Like '*each other*')) OR

((REInterviews.ElicitAct) Like '*each other*')) OR

((REInterviews.ElicitQte) Like '*each other*'))

ORDER BY REInterviews.ProjNum;

Note that though this query generated seven projects, three projects are not applicable as they described projects where the stakeholders “did not talk to each other”.

Pattern-5-Communicate				
ProjNum	CommAct	CommQte	ElicitAct	ElicitQte
3	For this project the RA had to collect requirements from stakeholders at four locations, 2 in the US, 1 in Asia and 1 in Europe. Approximately 50 stakeholders were directly involved in contributing requirements. Not only were Asian stakeholders SMEs, they would also have been the end		RA preference was to lead face-to-face, one-on-one and small group interviews. When that was not possible, some interviews were conducted via teleconferences and a few via emails. The RA interviewed the USA1 stakeholders in person, the USA2	Typically the RA “spent several weeks in the Asian location and then a few weeks in the US. If we happened to be in the US at the same time we would go in person. If we happened to be in
4	The RA communicated with stakeholders at 3 locations. There were approximately 15 SMEs at the US1 location, 15-20 SMEs at the US2 location and, 25-30 SMEs at the Asian location. For this project there were 2 RAs: our interviewee, the project RA who led the requirements	Per the RA all of the SMEs “would have been almost unanimously across the board very technical people who would specialize in one or more areas of factory automation or factory management. Factory managers	The RA led telephone conferences when she could not meet with stakeholders face-to-face. Face-to-face or phone conference participation by the stakeholders was dependent upon the RA’s location, whether the RA was in US1, US2, or Asia, during the project.	The RA shared, “My personal preference is always to do the elicitation in person, when that was practical and possible. For that reason, we used to travel to and from Asia, and
5	RA held face-to-face meetings with the USA1 and USA2 since they were all co-located at the same site. The meetings turned into telephone conferences once USA3 joined. Telephone conferencing worked well, especially as an mp3 recording of the entire session was attached to	“We started out with telephone and later moved to video conferences.” Though this was not an assigned task, “For almost every team <of stakeholders> that got added on, there was one buddy team	The project team employed an agile methodology and performed a modified Scrum. RA elicited and gathered requirements using a wiki. All of the user requests (i.e. bugs, feature requests) were or became requirements for the development team to work on. Keeping	Per RA, “We used modified Scrum. We did iterative [development] and then interacted with the customer to get their feature requirements.”
6	Initially the RA communicated with 2 co-located stakeholders at USA1 and 5-6 stakeholders at USA 2. The stakeholder roles at USA1 were user experience and project manager, and	Initially - Per the RA, “I met with the stakeholders, most input seem to come from user experience.” During later phases - It didn’t	The RA met with the co-located user experience stakeholder face-to-face; and the product management stakeholder for USA2 was also included in some of these meetings via telephone conference. RA	The RA mentioned that, “there was some communications that I was not privy to... I would get it through the

Share-Specification Query used to produce Table 5.11

SELECT

REInterviews.ProjNum, REInterviews.SpecsAct, REInterviews.SpecsQte

FROM REInterviews

ORDER BY REInterviews.ProjNum;

Pattern-Share-Specs		
Proj	SpecsAct	SpecsQte
4	Initially an RA or RE (RA/E) would draft a document and then send it to the librarian for safekeeping, and to the necessary stakeholders for their review and approval. Drafts of the documentation were shared with the stakeholders via email. After reviewing the stakeholders would reply to the RA/E. The RA/E would then make the necessary modifications and update the version number before sending the document to the librarian. Afterwards	In answer to this follow questions, When several people were gathering the requirements, then would all of you get together on the phone and discuss the requirements?, the RA responded, "Yes either that or we would manage it via documentation, where I would check the document and say, 'here is my first part of whatever the US cases', or, 'the part is for whatever documentation I think needs to be
5	The aforementioned requirements and requests, project documentation and artifacts were maintained in a project wiki that was accessible to all stakeholders for addition and modification of information.	The stakeholders could "post requests, screenshots, mockups, and priorities, whatever. They could also see our specifications, design docs, verification, and schedule... But team-to-team it was individually negotiated. Again it's tribal, we were all separate companies and they still kind of held on to that independence."
6	RA documented requirements in MS Word and Use Cases were created using Visio. Prototypes were an appendix to the Use Cases. Wire Frames illustrating main and alternate scenarios would follow the use cases. The requirements document was emailed to the users for their review.	Per the RA, "In most cases I would get (documentation updates) through email. So (the stakeholders) would either put their comments in the body of the email message without an attachment of their markups; or I would get an attachment with their markups embedded and I would have to go through the document; and then I would integrate these comments and of course you run into situations where there's maybe discrepancies or
7	At the beginning of the project the RA and his team created text documents and an Access database as they gathered the requirements. "When we were doing interviews we basically met and consolidated what we had written down.... Then we clustered them into problems, wishes and so on. And soon they were tools basically..."	"We had a couple of tables with who said the requirements, what types of requirements, how many times has it been stated, and so on. We used that in the beginning and we switched over during the project time. We had to do a deliverable, so we brought all of the requirements into the deliverable which was in a normal Word format... And then I transformed it, basically I put all of the requirements into a wiki." The RA designed the
8	The responsibility for inputting the requirements in DOORS was shared by the RA and her team, the business analysts, trained SMEs and, "in some cases the analysts at the sub-contractors site."	As the RA explained, "but there was more that wasn't in DOORS. And this caused problems initially because there were things, and vice versa, there were things in the Doors database that shouldn't have been, for example,

No-Travel Query used to produce table 5.12

SELECT

REProjInfo.ProjNum, REProjInfo.Industry,
RESites.SiteName, RESites.TotalRAs,
RESites.TotalLSPs, RESites.TotalStakeholders

FROM REProjInfo INNER JOIN RESites ON REProjInfo.ProjNum = RESites.ProjNum

WHERE

((REProjInfo.ProjNum)=1 OR (REProjInfo.ProjNum)=2 OR (REProjInfo.ProjNum)=6))

ORDER BY REProjInfo.ProjNum, RESites.SiteNum;

Patterns-No-Travel						
ProjNum	Industry	SiteName	TotalRAs	TotalLSPs	TotalStakeholders	
	Financial Services	USA0	1	0	0	
1	Financial Services	USA1	0	1	2	
2	Telecommunications	USA0	1	0	0	
2	Telecommunications	USA1	0	0	3	
2	Telecommunications	USA2	0	0	2	
6	Retail	USA1	1	0	2	
6	Retail	USA2	0	1	5	

Bibliography

- [1] S. Robertson and J. Robertson, *Mastering the Requirements Process*, 2 ed: Addison-Wesley, 2006.
- [2] J. Cleland Huang, "Software Requirements," in *Software Engineering, Volume 1: The Development Process*, vol. 1, R. H. Thayer and M. J. Christensen, Eds., Third ed: John Wiley & Sons, 2005.
- [3] S. Easterbrook, "Elicitation of Requirements from Multiple Perspectives," in *Department of Computing*. London: Imperial College of Science, Technology and Medicine, University of London, 1991, pp. 126.
- [4] P. Sawyer, "The Context of Software Requirements," in *Software Engineering, Volume 1: The Development Process*, vol. 1, R. H. Thayer and M. J. Christensen, Eds., Third ed: John Wiley & Sons, 2005.
- [5] S. Easterbrook, J. Singer, P. Storey, and D. Damian, "Empirical Research Methods in Software Engineering," in *Advanced Topics in Empirical Software Engineering: A Handbook*, F. Shull and J. Singer, Eds.: Springer, 2007.
- [6] K. E. Wiegers, *Software Requirements*, 2nd ed. Redmond, WA: Microsoft Press, 2003.
- [7] B. H. Cheng and J. A. Atlee, "Research Directions in Requirements Engineering," *IEEE Future of Software Engineering*, 2007.
- [8] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: A Roadmap," presented at International Conference on Software Engineering, Limerick, Ireland, 2000.
- [9] R. Smith, "Tame The Requirements Beast: Aligning Application Development Business Goals," *Information Week Analytics Report*, 2009.
- [10] B. Boehm and V. R. Basili, *Software Defect Reduction Top 10 List*, 2001.
- [11] D. Leffingwell and D. Widrig, *Managing Software Requirements: A Unified Approach*: Addison Wesley, 2000.
- [12] J. D. Herbsleb, "Global Software Engineering: The Future of Socio-Technical Coordination," *Future of Software Engineering (FOSE'07)*, pp. 188-198, 2007.
- [13] K. Desouza, Y. Awazu, and P. Baloh, "Managing Knowledge in Global Software Development Efforts: Issues and Practices," *IEEE Software*, vol. 23, pp. 30-37, 2006.
- [14] D. Damian, "Stakeholders in Global Requirements Engineering: Lessons Learned from Practice," *IEEE Software*, vol. 24, pp. 21-27, 2007.

- [15] D. Damian, S. Marczak, and I. Kwan, "Collaboration Patterns and the Impact of Distance on Awareness in Requirements-Centered Social Networks," presented at 15th IEEE International Requirements Engineering Conference (RE 2007), New Delhi, India, 2007.
- [16] A. Taweel, B. Delaney, T. N. Arvanitis, and L. Zhao, "Communication, Knowledge and Coordination Management in Globally Distributed Software Development: Informed by a Software Engineering Case Study," presented at ICGSE, 2009.
- [17] D. E. Damian and D. Zowghi, "The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization," presented at IEEE Joint International Conference on Requirements Engineering (RE '02), 2002.
- [18] M. Shaw, "Writing Good Software Engineering Research Papers," presented at 25th IEEE International Conference on Software Engineering (ICSE '03), 2003.
- [19] OMG, "OMG Unified Modeling Language (OMG UML) Infrastructure [Online]," 2011.
- [20] OMG, "OMG Unified Modeling Language (OMG UML) Superstructure [Online]," 2011.
- [21] I. Alexander and S. Robertson, "Understanding Project Sociology by Modeling Stakeholders," *IEEE Software*, vol. 21, pp. 23-27, 2004.
- [22] N. Stanford, *Organizational Structures: Guide to Organisation Design: Creating High-Performance and Adaptable Enterprises*: Profile Books, 2007.
- [23] B. Berenbach, "Impact of Organizational Structure on Distributed Requirements Engineering Processes: Lessons Learned," presented at International Workshop on Global Software Development for the Practitioner, New York, NY, USA, 2006.
- [24] P. Laurent, P. Mader, J. Cleland Huang, and A. Steele, "A Taxonomy and Visual Notation for Modeling Globally Distributed Requirements Engineering Projects," presented at 5th IEEE International Conference on Global Software Engineering, (ICGSE '10) Princeton, USA, 2010.
- [25] P. Laurent, "Globally Distributed Requirements Engineering," presented at 5th IEEE International Conference on Global Software Engineering, (ICGSE '10) PhD Symposium, Princeton, USA, 2010.
- [26] P. Laurent and J. Cleland Huang, "Requirements-Gathering Collaborative Networks in Distributed Software Projects," presented at 17th IEEE International Conference on Requirements Engineering, Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills (CIRCUS '09) workshop, Atlanta, GA, 2009.
- [27] P. Laurent and J. Cleland Huang, "Lessons Learned from Open Source Projects for Facilitating online Requirements Processes," presented at 15th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2009), Amsterdam, Holland, 2009.

- [28] P. Laurent and J. Cleland-Huang, "Requirements-Gathering Collaborative Networks," presented at 31st International Conference on Software Engineering, Socio-Technical Congruence (STC '09) workshop, Vancouver, Canada, 2009.
- [29] P. Laurent and J. Cleland Huang, "Managing Requirements in Vendor-Based Open Source Software Development Projects," in *Open Source Software Developments*. Hauppauge: Nova Science Publishers, Inc., 2010.
- [30] P. Laurent, J. Cleland-Huang, and C. Duan, "Towards Automated Requirements Triage," presented at 15th IEEE International Requirements Engineering Conference (RE 2007), Delhi, India, 2007.
- [31] P. Laurent, "Task Modeling and Analysis of Traceability Users in the Wild," presented at 15th IEEE International Requirements Engineering Conference (RE 2007) Doctoral Consortium, Delhi, India, 2007.
- [32] P. Laurent, "Traceability Task Modeling," presented at DePaul CTI Research Symposium (CTIRS 2007), Chicago, IL, 2007.
- [33] P. Laurent, "Trace Support for Requirements Prioritization," presented at ACM International Symposium on Grand Challenges of Traceability (GCT '07), 2007.
- [34] C. Duan, P. Laurent, J. Cleland Huang, and C. Kwiatkowski, "Towards automated requirements prioritization and triage," *Requirements Engineering*, vol. 14, pp. 73-89, 2009.
- [35] P. Laurent and A. Steele, "Using GOMS KLM to Support Cross Platform Prototyping," presented at Midwest Business Administration Association (MBAA 2005), Chicago, IL, 2005.
- [36] P. Laurent and A. Steele, "From Prototype to Application," presented at Midwest Software Engineering Conference, Chicago, IL, 2004.
- [37] P. Laurent, "Cross Platform Prototyping," presented at DePaul CTI Research Symposium (CTIRS 2001), Chicago, IL, 2001.
- [38] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques*. West Sussex, England: Wiley & Sons, 1998.
- [39] E. M. Bennatan, *On Time Within Budget*, 3rd ed. New York: John Wiley & Sons, Inc., 2000.
- [40] J. Sodhi and P. Sodhi, *Managing IT Systems Requirements: Management Concepts*, 2003.
- [41] A. Davis, O. Dieste, A. Hickey, N. Juristo, and A. Moreno, "Effectiveness of Requirements Elicitation Techniques," presented at IEEE International Requirements Engineering Conference (RE06), Minneapolis, MN, 2006.
- [42] A. M. Davis, "The Art of Requirements Triage," *IEEE Computer*, vol. 36, pp. 42-49, 2003.

- [43] J. Cleland-Huang, B. Mobasher, N. Tomuro, and C. Duan, "Towards Automated Requirements Triage: Increasing Product Value through Informed Recommendations," presented at Software Engineering Research Center (SERC) Showcase, Muncie, Indiana, 2006.
- [44] N. R. Mead, "Requirements Prioritization Introduction," 2006.
- [45] J. W. Brackett, "Software Engineering," presented at Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1990.
- [46] J. Karlsson, "Towards a Strategy for Software Requirements Selection. Licentiate Thesis 513, Department of Computer and Information Science," Linköping University, 1995.
- [47] K. Beck, *Extreme Programming Explained*, 7th ed. Boston, MA: Addison-Wesley, 2001.
- [48] D. Leffingwell and D. Widrig, *Managing Software Requirements: A Use Case Approach*, 2nd ed. Boston, MA: Addison-Wesley, 2003.
- [49] B. W. Boehm and R. Ross, "Theory-W Software Project Management: Principles and Examples," *IEEE Transactions on Software Engineering*, vol. 15, pp. 902-916, 1989.
- [50] J. Park, D. Port, and B. Boehm, "Supporting Distributed Collaborative Prioritization for Win-Win Requirements Capture and Negotiation," presented at International Third World Multi-conference on Systemics, Cybernetics and Informatics (SCI '99), Orlando, FL, 1999.
- [51] K. E. Wiegers, *Software Requirements*. Redmond, WA: Microsoft Press, 1999.
- [52] F. Moisiadis, "Prioritising Scenario Evolution," presented at International Conference on Requirements Engineering (ICRE 2000), 2000.
- [53] F. Moisiadis and A. McRae, "A Requirements Prioritisation Tool," presented at 6th Australian Workshop on Requirements Engineering (AWRE 2001), Sydney, Australia, 2001.
- [54] J. Karlsson and K. Ryan, "A Cost-Value Approach for Prioritizing Requirements," *IEEE Software*, vol. 14, pp. 67-75, 1997.
- [55] T. L. Saaty, *The Analytical Hierarchy Process*. New York, NY: McGraw-Hill, 1980.
- [56] J. Karlsson, "Software Requirements Prioritization," presented at The 2nd International Conference on Requirements Engineering (ICRE '96), 1996.
- [57] S. Group, "CHAOS Report," 1995.
- [58] K. Stapel, E. Knauss, K. Schneider, and N. Zazworka, "FLOW Mapping: Planning and Managing Communication in Distributed Teams," presented at 6th IEEE International Conference on Global Software Engineering (ICGSE, 2011), Helsinki, Finland, 2011.
- [59] J. Cleland-Huang, H. Dumitru, C. Duan, and C. Castro-Herrera, "Automated support for managing feature requests in open forums," *Communication of the ACM*, 2009.
- [60] B. Decker, E. Ras, J. Rech, P. Jaubert, and M. Rieth, "Wiki-Based Stakeholder Participation in Requirements Engineering," in *IEEE Software*, 2000, pp. 28-35.

- [61] D. J. Dekker and P. H. J. Hendriks, "Social Network Analysis," in *Encyclopedia of Knowledge Management* D. G. Schwartz, Ed.: IG Publishing, 2006.
- [62] L. Freeman, *The Development of Social Network Analysis*. Vancouver: Empirical Press, 2006.
- [63] C. deSouza, R.B., S. Quirk, E. Trainer, and D. F. Redmiles, "Supporting Collaborative Software Development through the Visualization of Socio-Technical Dependencies," presented at GROUP '07, Sanibel Island, Florida, 2007.
- [64] D. Amyot, "Introduction to the User Requirements Notation: learning by example," *Computer Networks*, pp. 285-301, 2003.
- [65] A. van Lamsweerde, "Requirements Engineeri: from craft to discipline," presented at 16th ACM SIGSOFT International Symposium on Foundation of Software Engineering, 2008.
- [66] I. Davies, P. Green, M. Roseman, M. Indulska, and S. Gallo, "How do practitioners use conceptual modeling in practice?," *Data Knowledge Engineering*, pp. 358-380, 2006.
- [67] D. L. Moody and P. Heymans, "Visual Syntax does matter: improving the cognitive effectiveness of the i* visual notation," *Requirements Engineering*, pp. 141-175, 2010.
- [68] A. Monk and S. Howard, "Methods and Tools: The Rich Picture: A tool for reasoning about Work Context," *Interactions*, vol. 5, pp. 21-30, 1998.
- [69] B. Decker, E. Ras, J. Rech, P. Jaubert, and M. Rieth, "Wiki-Based Stakeholder Participation in Requirements Engineering," *IEEE Software*, vol. 24, pp. 28-35, 2007.
- [70] B. Dagenais, H. Ossher, R. Bellamy, M. Robillard, and J. P. deVries, "Moving into a New Software Project Landscape," presented at International Conference on Software Engineering (ICSE '10), Capetown, Software Africa, 2010.
- [71] A. Fink, *The Survey Handbook*, vol. 1. Thousand Oaks: Sage Publications, Inc., 2003.
- [72] A. Strauss and J. Corbin, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 2nd ed. Thousand Oaks: Sage Publications, Inc., 1998.
- [73] B. Gillham, "Developing a Questionnaire," in *Real World Research*, 2 ed. London: Continuum International Publishing Group, 2007.
- [74] Wikipedia, "IP Address," pp. Wikipedia, the free encyclopedia.
- [75] W. Scacchi, "Free/Open Source Software Development: Recent Research Results and Emerging Opportunities," presented at 6th Joint Meeting on European Software Engineering conference (ESEC) and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE) '07, Dubrovnik, Croatia, 2007.
- [76] J. Cleland Huang, H. Dumitru, C. Duan, and C. Castro-Herrera, "Automated Support for Managing Feature Requests in Open Forums," *Communications of the ACM*, vol. 52, pp. 68-74, 2009.

- [77] J. Cleland-Huang, B. Berenbach, S. Clark, E. Romanova, and R. Settini, "Best Practices for Automated Traceability," *Computer*, vol. 40, pp. 27-35, 2007.
- [78] C. Castro-Herrera, J. Cleland Huang, and B. Mobasher, "A recommender system for dynamically evolving online forums," presented at 3rd ACM Conference on Recommender Systems, RecSys '09, New York, NY, USA, 2009.
- [79] C. Castro-Herrera, J. Cleland Huang, and B. Mobasher, "Enhancing Stakeholder Profiles to Improve Recommendations in Online Requirements Elicitation," presented at 17th IEEE International Requirements Engineering Conference, RE '09, Atlanta, GA, 2009.
- [80] C. Castro-Herrera, C. Duan, J. Cleland Huang, and B. Mobasher, "A Recommender System for Requirements Elicitation in Large-Scale Software Projects," presented at ACM Symposium on Applied Computing, Data Mining track, Honolulu, HI, 2009.
- [81] P. Kroll and P. Krutchen, *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*: Addison-Wesley, 2003.
- [82] P. F. Gordon, "Numerical Cognition Without Words: Evidence from Amazonia," *Science*, vol. 306, pp. 496-499, 2004.
- [83] D. L. Moody, "The 'Physics' of Notations: Towards a Scientific Basis for Constructing Visual Notations in Software Engineering," *IEEE Transactions on Software Engineering*, vol. 35, pp. 756-779, 2009.
- [84] B. Paech, "What is a Requirements Engineer?," *Software* vol. 25, pp. 16-17, 2008.
- [85] R. Sangwan, M. Bass, N. Mullick, D. J. Paulish, and J. Kazmeier, *Global Software Development Handbook*: Auerbach Publications, 2007.
- [86] B. Carkenord, *Seven Steps to Mastering Business Analysis*: J. Ross Publishing, 2009.
- [87] B. Berenbach, D. J. Paulish, J. Kazmeier, and A. Rudorfer, *Software & Systems Requirements Engineering: In Practice*: McGraw-Hill/Osborne, 2009.
- [88] R. O. Briggs, A. Davis, D. D. Mittleman, and J. Murphy, "Classification Method for Collaborative Technology," 2007.
- [89] R. F. Ter Bush, "Silence, Attribution Accuracy and Virtual Environments: Implications for developers and facilitators," presented at 39th International Conference on System Sciences Hawaii, 2006.
- [90] D. D. Mittleman, R. O. Briggs, and J. Nunamaker, Jay F., "Best Practices from Experiences in facilitating Virtual Meetings," *Group Facilitation: A Research and Applications Journal*, vol. 2, pp. 5-14, 2000.
- [91] R. Turner and B. Boehm, "Using Risk to Balance Agile and Plan-Driven Methods," *Computer*, vol. 36, pp. 57-66, 2003.

- [92] J. O. Coplien and N. B. Harrison, *Organizational Patterns of Agile Software Development*: Pearson Prentice Hall, 2005.
- [93] D. Damian, I. Kwan, and S. Marczak, "Requirements-Driven Collaboration: Leveraging the Invisible Relationships between Requirements and People," in *Collaborative Software Engineering*, I. Mistrik, J. Grundy, A. van der Hoek, and J. Whitehead, Eds.: Springer-Verlag 2010.
- [94] T. Wolf, A. Schroter, D. Damian, L. Panjer, and T. H. D. Nguyen, "Mining Task-Based Social networks to Explore Collaboration in Software Teams," *IEEE Software*, vol. 26, pp. 58-66, 2009.
- [95] S. Marczak, D. Damian, U. Stege, and A. Schroter, "Information Brokers in Requirement-Dependency Social Networks," presented at International Requirements Engineering Conference (RE '08), Barcelona, Spain, 2008.
- [96] K. Ehrlich, M. Helander, G. Valetto, S. Davies, and C. Williams, "An Analysis of Congruence Gaps and Their Effect Distributed Software Development," presented at 30th International Conference on Software Engineering, Socio-Technical Congruence Workshop, 2008, Leipzig, Germany, 2008.
- [97] P. Grunbacher, B. W. Boehm, and R. O. Briggs, "EasyWinWin: A Groupware-Supported Methodology For Requirements Negotiation."
- [98] D. D. Mittleman, R. O. Briggs, J. Murphy, and A. Davis, "Toward a Taxonomy of Groupware Technologies."
- [99] O. Gotel and P. Mader, "How to Select a Requirements Management Tool: Initial Step," presented at 17th IEEE International Requirements Engineering Conference (RE09) Atlanta, GA, 2009.
- [100] D. E. Damian and D. Zowghi, "Requirements Engineering challenges in multi-site software development Organizations," *Requirements Engineering*, vol. 8, pp. 149-160, 2003.
- [101] A. M. Davis, Ó. D. Tubío, A. M. Hickey, N. J. Juzgado, and A. M. Moreno, "Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review," presented at IEEE International Requirements Engineering Conference (RE06), Minneapolis, MN, 2006.