UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: IRUM INAYAT [redacted]

Registration/Matric No: WHA110012

Name of Degree: PHD

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

Framework to Study the Requirements-driven Collaboration in Agile Teams

Field of Study:

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature　　　　　　　　Date

Subscribed and solemnly declared before,

Witness's Signature　　　　　　　　Date

Name:
Designation:

Abstract

Requirements engineering requires intensive collaboration among team members. The importance of collaboration in agile methods is also undeniable. Due to their emphasis on collaboration, agile methods and requirements engineering activities seem to mutually support each other in software development. However, very little is still known about the "agile way" of dealing with requirements and how collaboration driven by requirements takes place especially among distributed team members. The main aim of this research is to investigate the socio-technical aspects of requirements-driven collaboration in agile teams. Firstly, this research identified the most relevant socio-technical aspects of requirements-driven collaboration among agile teams through an online survey conducted on industry practitioners, as communication and awareness. Secondly, a framework was proposed to study the identified socio-technical aspects of requirements-driven collaboration among agile teams and a prototype was developed to partially automate the framework. Thirdly, an empirical investigation was conducted by studying four IT-based projects carried out in four different organizations. This empirical investigation led to the practical implementation of the proposed framework to study the requirements-driven collaboration among agile teams. This research was validated from two perspectives. From the academic perspective, the results show that the framework is structurally acceptable. From the industrial perspective, an applicability validation was performed to assess the application of the proposed framework while a utility validation was conducted to gauge the usefulness of the proposed framework. The study provides implications for both research and industry practitioners in the form of further research and tool development for agile teams collaboration and performance analysis underlying the concepts proposed in this study.

Abstrak

Kejuruteraan keperluan memerlukan kerjasama yang intensif di kalangan ahli pasukan. Kepentingan kerjasama dalam kaedah tangkas juga tidak dapat dinafikan. Oleh kerana penekanan mereka kepada kerjasama, kaedah tangkas dan aktiviti kejuruteraan keperluan dilihat saling menyokong satu sama lain dalam pembangunan perisian. Walau bagaimanapun , amat sedikit yang diketahui mengenai "cara tangkas" menangani keperluan dan bagaimana kerjasama yang didorong oleh keperluan berlaku terutamanya di kalangan ahli-ahli pasukan yang diedarkan. Tujuan utama kajian ini adalah untuk menyiasat aspek-aspek sosio- teknikal daripada keperluan yang didorong oleh kerjasama dalam pasukan tangkas. Pertama , kajian ini mengenal pasti aspek-aspek sosio- teknikal yang berkaitan dengan keperluan yang didorong oleh kerjasama antara pasukan tangkas melalui kajian dalam talian yang dijalankan bersama pengamal industri seperti komunikasi dan kesedaran. Kedua, rangka kerja yang dicadangkan untuk mengkaji aspek sosio- teknikal yang dikenal pasti daripada keperluan yang didorong oleh kerjasama antara pasukan tangkas dan prototaip dibangunkan untuk mengautomasikan sebahagian rangka kerja tersebut. Ketiga, sesuatu penyiasatan empirikal dijalankan dengan mengkaji empat projek berasaskan IT yang dilakukan dalam empat organisasi berbeza. Siasatan empirikal membawa kepada pelaksanaan yang praktikal bagi rangka kerja yang dicadangkan untuk mengkaji kerjasama yang didorong oleh keperluan di kalangan pasukan tangkas. Kajian ini telah disahkan dari dua perspektif. Dari perspektif akademik, keputusan menunjukkan bahawa rangka kerja adalah struktur yang boleh diterima. Dari perspektif industri, pengesahan kebolehgunaan dilakukan untuk menilai penggunaan rangka kerja yang dicadangkan.Pengesahan kebergunaan dijalankan untuk mengukur kegunaan rangka kerja yang dicadangkan. Kajian ini memberi implikasi untuk kedua-dua penyelidikan dan pengamal industri dalam bentuk penyelidikan lanjut dan pembangunan

alat untuk kerjasama pasukan tangkas dan analisis prestasi berasaskan konsep-konsep yang dicadangkan dalam kajian ini.

ACKNOWLEDGEMENTS

Prior to all, I recompense my gratitude to Allah Who sanctified me with the strength to carry out this thesis and potency to fulfill this responsibility.

I am highly beholden to my Supervisor Prof. Dr. Siti Salwah Salim for her guidance and support. Through the years I was lucky enough to learn from your experience. I have learned from you to work consistently with full dedication right from day one. Today when I look back I realize that all your hard words were only for good. Thank you for helping me to adapt to the new realm of research and making tough goals achievable for me.

I am hugely grateful to Dr. Sabrina Marczak for her utmost support, guidance, time and for making this expedition possible for me. Your motivation and friendly advice at the time of distress helped me to cope with the pressure and to fight back the things with zeal. This work was not possible without your help and guidance.

I am thankful to my co supervisor Dr. Zarinah M. Kasirun for being very kind and gentle throughout. I also want to thank Dr. Maya Daneva for her guidance on the way.

I recompense my regards to my family for their prayers and motivation to get through. I miss my father hugely with his blessings I started this journey but he did not live up to see me accomplish; though I know his prayers are still there with me making my path easier. I would also like to pay my genial gratitude to my husband who suffered the most during this time. Your love and support helped me to believe in myself and be where I am today.

I am grateful to Atika Qazi for being a great friend throughout this time and supported me through thick and thin. Last but not the least I am grateful to Comsats Institute of Information Technology, Pakistan and University of Malaya for providing me with the opportunity to carry out this research.

To

My Mother and Grandmother for being the embodiment of strength

for me

&

My Father who is smiling down at me from heaven

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

This chapter provides an introduction of the research problem, research objectives, research questions, scope, limitations, and significance of the research.

1.1. Background

Traditional software development models like the waterfall are phase driven. Requirements are defined in the early stage of the software development life cycle and then implemented, tested and delivered to the customer. In traditional software development models, changes in requirements are considered hard to implement especially at the later stages of the project development life cycle. Late identification of changes in requirements lead to misinterpretation and misunderstanding and that can cause possible project failures (Curtis, Krasner, & Iscoe, 1988)(Khaled & Madhavji, 1995).

Agile methods, being collaborative in nature, entail an organic management of requirements, unlike traditional software development methods (Cao & Ramesh, 2008) and serve as an alternative to the challenges posed by traditional software development (i.e. communication gaps, over scoping (Bjarnason, Wnuk, & Regnell, 2010)(Bjarnason, Wnuk, & Regnell, 2011a)). Agile methods emphasize on extensive collaboration between customers and developers, and encourage small, self-organized and collocated teams (Sharp, Robinson, & Hall, 2003). In such a dynamic software development process, requirements are highly volatile and constant collaboration is essential to cope with the ever changing requirements for risk mitigation due to dependencies (Martakis & Daneva, 2013). Developer collaboration is dependent on the communication of changes and of new tasks, as well as on the awareness of what others are doing, whether they are available to help, or what they know that can help one's work (Damian, Izquierdo, & Singer, 2007).

Empirical evidence of patterns of collaboration in the activities of requirements engineering for traditional software development teams exists in the literature (e.g., (Damian, Marczak, & Kwan, 2007; Damian, Kwan, & Marczak, 2010a; Marczak & Damian, 2011a)). Collaboration that is driven by software requirements is named requirements-driven collaboration (RDC) by Damian et al., (2010). However, only a small number of studies have explored collaboration patterns among agile teams so far (e.g. (Cataldo & Ehrlich, 2011) (Abdullah, Sharp, & Honiden, 2011) (Ehrlich, Cataldo, & York, 2014). Unfortunately, there is not much work related to the collaboration patterns of agile teams in the literature. Therefore, there is a need to identify the most relevant socio-technical aspects of RDC among agile teams and to formulate a framework to pave ways for devising a scheme to study RDC among agile teams.

In this research, requirements-driven collaboration (RDC) among agile teams is investigated. through team members communication (within allocated team members and with the people or teams who emerge throughout the lifecycle of the project called "emergent members") and their awareness of each other (i.e. awareness of each other's presence (Ehrlich & Chang, 2006), general traits (Ehrlich & Chang, 2006), current task (Ehrlich & Chang, 2006) and work status). Social network analysis (SNA) measures were used to unveil the collaboration patterns of agile teams and the results were empirically supported by four case studies. The results revealed interesting collaboration patterns of agile teams and determines the effect of collaboration on iteration performance.

1.2. Problem statement

In a distributed agile team, it is challenging to collaborate effectively. The distance between the teams contributes more to the collaboration challenges faced by the teams. However, effective collaboration is necessary for software development teams working together on a set of certain interdependent requirements. Moreover, very little is still known about the "agile way" of dealing with requirements and how collaboration driven by requirements in distributed agile teams takes place. Researchers and practitioners lack knowledge on how distributed, cross-functional and self-managed agile teams manage collaboration among each other while dealing with requirements. Unlike traditional software development teams, agile cross-functional teams are closely knit and highly interactive. Communication is constant and free of imposed organizational barriers. Team members are self-manageable and empowered to make decisions on their own in contrast to the traditionally centralized structures dependent on project managers. Therefore there is a need for a better understanding and scoping of collaboration among agile teams in the requirements engineering process. This would enable the intensification of practices, tools and methods to support collaboration among cross functional agile teams.

The focus of this study is to identify the most relevant socio-technical aspects of requirements-driven collaboration among agile teams and to propose a framework comprising a set of steps to study the requirements-driven collaboration among agile teams. Software development teams develop social and technical dependencies on each other while working on certain set of interdependent requirements. These social and technical dependencies give rise to socio-technical relationships such as team members communicating for requirements changes. Therefore, communication, knowledge sharing, teamwork, coordination etc. are the socio-technical aspects of collaboration among teams. The main aim is to deepen the understanding of the most relevant (i.e. the most important socio-technical aspects of requirements-driven collaboration for agile software development practitioners) socio-technical aspects of requirements-driven collaboration (i.e. communication and awareness) among agile teams by studying their structures, and to study their impact on the iteration performance of agile teams.

1.3. Research Objectives

The basic theme of this research is to device a guideline to investigate collaboration among agile teams while dealing with requirements.

In this research following research objectives are aimed to be fulfilled:

- To identify the relevant socio-technical aspects of requirements-driven collaboration among agile teams.

Socio-technical aspects that underlie collaboration in agile teams are still not known in the literature. Several of the above quoted researchers have studied communication among agile teams but there is still a need to identify the relevant aspects and investigate them in detail. The detailed investigation includes probing the understanding of agile practitioners regarding collaboration and their perception of the most relevant socio-technical aspects in their day to day working. Therefore, the first research question is posed in order to identify these relevant aspects.

- To formulate a framework for studying requirements-driven collaboration among agile methods.

The aim here is to device a formal strategy and to provide a certain set of steps for studying the socio-technical aspects of RDC among agile teams and measure their effects on iteration performance. Although, there are a handful of studies that focus on socio-technical aspects of collaboration in agile teams especially communication (e.g. (Cataldo & Ehrlich, 2011) (Bjarnason, Wnuk, & Regnell, 2011a)). The need to have a systematic set to steps to methodologically support the study of socio-technical aspects of RDC in agile teams is required based on the concept of collaboration study among teams in traditional teams (Damian et al., 2010).

- To investigate the requirements-driven collaboration among agile teams and find its impact on the iteration performance.

The aim is to study the RDC among agile teams through empirical investigation following the proposed framework and finding its impact on the iteration performance.

- To evaluate the applicability and utility of the framework.

The aim is to assess the applicability and utility of the proposed framework.

1.4. Research Questions

From the above discussion, it can be clearly seen that requirements-driven collaboration in agile methods need to be investigated for its non-traditional and fluid nature. The research questions (RQ) posed to address the research goal are described below. The mapping of research questions to their respective research objectives is shown in Figure 1.1.

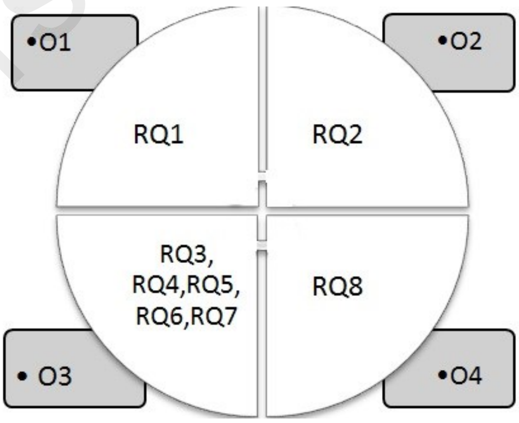


Figure 1.1: Mapping of research questions with research objectives

RQ1. What are the most relevant socio-technical aspects of requirements-driven collaboration among agile teams?

RQ2. How to investigate the relevant aspects of requirements-driven collaboration for agile teams?

RQ3. What are the characteristics of requirements-centric communication and awareness patterns among agile teams?

SRQ1. Who are involved in the networks?

SRQ2. What is the size of the network?

SRQ3. How dense are the networks?

SRQ4. How much communication took place between team members across and within roles, across and within allocation (i.e. assigned and emergent members), and across and within sites?

SRQ5. How much awareness was there among team members across and within roles, across and within allocation (i.e. assigned and emergent members), and across and within sites?

RQ4. What kind of structures do requirements-centric communication and awareness networks have for agile teams?

SRQ6. Do agile teams have centralized communication and awareness networks structures?

SRQ7. Do virtual agile team members communicate in an equal manner with each other?

SRQ8. Do agile teams have members working more closely together than to others?

SRQ9. Are there lone or isolated team members or groups of members in agile teams?

SRQ10. Do agile teams have the tendency to cluster with the increase in communication?

RQ5. Which information flow patterns do agile teams follow?

SRQ11. Do agile teams have some actors playing a central part in information flow?

SRQ12. Do agile teams have such member(s) who play a pivotal part in information exchange?

RQ6. How do communication and awareness networks impact the team performance in agile teams?

RQ7. What are the practical implications of this study for the industry and research fraternity studying requirements-driven collaboration among agile teams?

RQ8. How do potential users perceive the applicability and utility of the proposed framework?

1.5. Research Methodology

This study employed a constructive research approach for achieving all of the objectives explained above. The selection of this approach was based on the structure and requirement of this research. The constructive research approach uses artefacts (both practical and theoretical) to provide solution to a particular problem. The results achieved by the constructive approach have both practical and theoretical relevance (Gordana, 2008). Therefore, in this research, with the constructive method as the core, the emphasis was on utilizing pre-existing knowledge and building a viable solution. The main phases of the research mapped with the research objectives are shown in Figure 1.2.

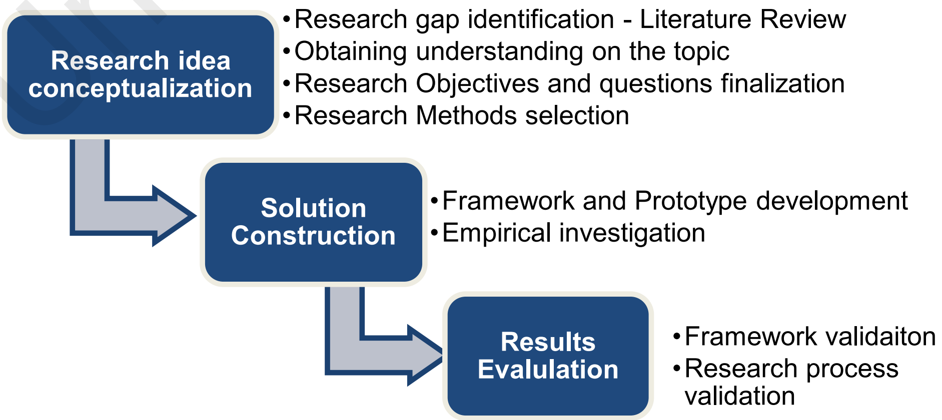


Figure 1.2: Research activities to achieve proposed research objectives

1.6. Research contributions

The contributions of this research are:

a) Identification of the most relevant socio-technical aspects of RDC among agile teams

This study identifies the most relevant socio-technical aspects of RDC for agile teams through an online questionnaire based survey. The survey was conducted among agile methods practitioners using email and online professional network forums. The survey results showed awareness and communication as the two highly recognized socio-technical aspects by agile practitioners. The survey results provided the basis for this research.

b) A systematic literature review of the identified socio-technical aspects of requirements-driven collaboration

This research presents a systematic literature review (SLR) that explored and furthered the knowledge on the two identified aspects of RDC, i.e. communication and awareness. The SLR was based on the guidelines provided by Kitchenham & Charters (2007). Findings from the SLR helped to better understand both of the identified aspects.

c) A framework for studying requirements-driven collaboration in agile teams

The proposed framework helps to study the socio-technical aspects of collaboration for requirements-centric agile teams and their impact on the outcome and performance.

d) Framework Prototype Collab_Tool

A part of the framework was automated through a prototype called Collab_Tool to make it more practical and useable by the industry and research practitioners. The prototype can be extended further in order to fully automate the proposed framework.

e) Empirical Findings

The empirical results from multiple case studies are another contribution of this thesis. The empirical results can be used by researchers to conduct a similar kind of studies in different team settings in order to corroborate or contradict the findings. In addition, they are useful as the literature lacks empirical evidence on the socio-technical aspects of RDC among agile teams.

f) Evaluation results

The framework has been evaluated to assess its applicability and utility for the practitioners. The evaluation results establish how much the proposed framework can be useful for industry and researchers.

1.7. Significance of Research

The result of this research has implications for researchers and industry practitioners. The framework provides a pedestal for future studies on the socio-technical aspects of agile teams. The framework can be used to conduct more empirical studies for other agile methods and their variants, for different team settings and for other socio-technical aspects. The prototype lessens the manual work involved in analysis and provides an automated platform for industry practitioners to assess their teams. Likewise, researchers can make use of the prototype to lessen their labor while dealing with collaboration data of large teams.

The empirical investigation of multiple cases sheds light on the collaboration patterns of agile teams. The results can be generalized for agile teams facing a more or less similar kind of set up and circumstances. The findings help the managers to assess the collaboration patterns of their teams in connection with the team performance. Moreover, the results help the managers to invest in having a well-defined infrastructure to allow team members to contact their remote colleagues and practices to allow everyone to know how to work in order to achieve similar situations.

1.8. Scope and limitations

In this research, collaboration was studied among agile teams following the scrum methodology. Therefore, the scope of this study is limited to agile teams following the scrum methodology only. It cannot cover agile teams following newly emerged methods such as Scrum-ban etc. that comprise of different iteration structure i.e. swim lanes.

The performance factor that was taken into account also does not cover each and every performance aspect of the agile teams. Moreover, quality was also not included in the scope of this research

The sensitivity of this research lies with the fact that the data collected to build the networks, through the questionnaire, counted along with the recollections of team members on what had happened in the recent past. However, we took actions to minimize the impact of self-reported data. Firstly, we deployed the questionnaire at the end of each iteration and before the team members started working on new user stories. We also followed-up on missing answers as the questionnaires were filled out, thus reducing the effort a participant had to make to provide clarifications. Secondly, we triangulated data through interviews in order to learn how the participants perceived their collaboration with others in the team. Interviews were transcribed and further analyzed in comparison to the questionnaire responses.

Social networks are dynamic. Therefore, we designed a longitudinal study with two distinct data collection points to construct the networks and observe their behaviour over time. More data points could indicate the stability of our findings. However, by contrasting two iterations, indications of changes on the collaboration patterns were discussed and one of the main limitations of Damian et al.'s previous work (Damian et al., 2010) (Damian, Marczak, & Kwan, 2007) was overcome.

Generalizability of findings is another concern of software engineering empirical studies. The multiple case studies of four IT-organizations increase the likelihood of having results that represent a large sample of the population. This fact, in conjunction with the longitudinal study, contributes to a broader contribution than typically seen in software engineering empirical studies.

1.9. Structure of Thesis

There are eight chapters and four appendices in this thesis as explained below.

Chapter 1- Introduction- Introduces the research topic and provides an overview of the dissertation by briefly discussing the research problem, research objectives and questions, research methodology, research contribution, significance and limitations. It also presents the structure of the thesis.

Chapter 2- Literature Review – Gives an introduction of requirements engineering (RE), collaboration in requirements engineering, socio-technical aspects, and methods to study the socio-technical aspects. In addition, a systematic literature review was conducted to further the knowledge on Agile RE and to find out the practices of Agile RE and the challenges that Agile RE practices pose to the industry.

Chapter 3- Research Methodology- Provides an introduction of the research methodology carried out to fulfill the objectives of this research.

Chapter 4- Identification and review of the most relevant socio-technical aspects of Requirements-driven collaboration among agile teams- Provides an overview of the identification process of the most relevant socio-technical aspects of requirements-driven collaboration among agile teams, followed by a systematic literature review to study those aspects in detail.

Chapter 5- Framework for studying socio-technical aspects of requirements-driven collaboration among agile teams- Presents the proposed framework and the prototype developed to study communication and awareness in requirements-driven collaboration for agile teams. In addition, this chapter presents a prototype called Collab_tool developed to automate some parts of the proposed framework.

Chapter 6- Empirical Investigation - Presents case study 1 in detail and discusses the results of all the four cases that implemented the framework with respect to the research objectives and questions.

Chapter 7- Validation - Presents the validations performed to validate the applicability and utility of the proposed framework.

Chapter 8- Conclusion- Gives a review and conclusion of the work, provides implications and future work.

CHAPTER 2: LITERATURE REVIEW

In this chapter introduction to agile requirements engineering is provided by conducting a systematic literature review. Also, a general overview of requirements engineering process and collaboration in requirements engineering is provided in this chapter. This chapter summarizes the related work in the field and presents the results of a systematic literature review conducted to deepen the understanding of agile requirements engineering. At the end of this chapter the findings and implications of the systematic literature review are presented indicating the research gap.

2.1. Requirements Engineering in software development process

Requirements engineering is a crucial part of a human-centric social activity-software development. The requirements team is seen as a social organism and a requirement as a social entity (Goguen, 1993a). It is a process which determines the requirements (Cheng & Atlee, 2007); describes system behavior and its attributes (Nuseibeh & Easterbrook, 2000). The improvement in requirements engineering process improves other processes and ultimately the project planning and negotiation (Damian, Lanubile, & Mallardo, 2006). The requirements engineering process involves modeling, analyzing, negotiating and documenting stakeholder's demands (Cheng & Atlee, 2007). Successful requirements engineering means to have built what stakeholder needs (Macaulay, 1996); however poor requirements process leads to projects failures (Curtis et al., 1988) (Khaled & Madhavji, 1995).

2.2. Requirements Engineering perspective of collaboration

In this section underlying concepts such as collaboration, requirements-driven collaboration, and socio-technical relationship are explained.

2.2.1. Collaboration

Collaboration is defined in multiple terms according to the particular situation researcher is handling with. In literature it is defined in terms of mental aspects (Noble, Letsky, & Street, 2002), joint problem solving for the purpose of achieving a shared understanding, making a decision, or creating a product; task coordination , information sharing, actors actively sharing data, information, knowledge, perceptions, or concepts when they are working together toward a common purpose and how they might achieve that purpose efficiently or effectively (Alberts, 2001); and socio-technical aspects i.e. communication (Marczak & Damian, 2011b; Mishra & Mishra, 2009).

2.2.2. Requirements-Driven Collaboration

The requirements are taken as a unit of work around which collaboration occurs in terms of stakeholders coordination (Damian, Kwan, & Marczak, 2010). The requirements team is defined as a social organism (Goguen, 1993a) and requirement as a social entity (Goguen, 1993b). Requirements-Driven Collaboration (RDC) is a concept introduced by Damian and her colleagues as the collaboration compelled by the requirements during their development and management to the downstream artefacts i.e. code, design etc. (Damian et al., 2010). This collaboration drives on coordination needs in software development and based on social factors like communication, awareness, trust and knowledge sharing among many. The team members coordinate with each other during the requirements process by two methods (Ven, Delbecq, & Koenig, 1976) such as : (i) communication, and (ii) predefined processes using source code management, tracking tools etc. (Whitehead, 2007)(de Souza & Redmiles, 2007).

2.2.3. Socio-Technical Relationships

Requirements engineering being a part of human-centric social activity i.e. software development (Kim, Chan, & Keith, 2008) and accumulates many social and technical dependencies. The term socio-technical was originally defined by Trist and Bamforth while describing a psychological analysis of workmen in relation to their social structure and technical system for coal collection (Trist & Bamforth, 1951). The term was later adapted to be used in various dimensions, using the word social to denote "people/community" and the word technical to denote "technology/machinery"; in fact, these two words were considered inseparable whilst studying and analysing an organisation (Coombs, Knights, & Willmott, 1992).

Socio-technical aspects were studied to evaluate the relationships between actors, software systems and their environment, from social, organisational, psychological and technological perspectives e.g.(Olerup, 1989; Robinson & Sharp, 2005b; Rong, Li, Shao, & Chen, 2008; Russell, Drews, & Sue, 2002). It is a study of "the functions of the system" and "the functions of human cooperation" (Shneiderman & Rose, 1996). The technical context deals with tools adopted, artefacts generated and their interdependency; meanwhile, the social context deals with organisational structure, conventions and team structures as well as the social relationships among team members (Sarma, Herbsleb, & Hoek, 2008). Team members become dependent on each other and this dependency maintains socio-technical relationships as illustrated in Figure 2.1.

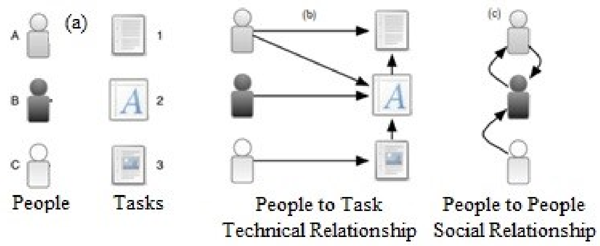


Figure 2.1: An example of socio technical relationships (Kwan, 2011)

The people to task dependency is shown is Figure 2.1(a), task to task dependency is shown in Figure 2.1(b) and social interaction between people working on interdependent workable items is shown in Figure 2.1(c).

The **social relationship** is the people to people relationship as if they communicate with each other (Cataldo, Wagstrom, & Carley, 2006). The **technical relationship** means people who have a work dependency as a relationship between people who contribute to a file that was a part of modification request (Cataldo et al., 2006) .

2.3. Methods to study Socio-Technical Aspects of Requirements-driven Collaboration (RDC)

Several methods are used in literature to study socio technical aspects of coordination in software development described below:

2.3.1. Social Network Analysis

Social network analysis is an approach that makes collaboration and communication more tangible and visible to everyone on the team (Wasserman & Faust, 2009). Difference SNA measures are used to study diverse behaviors of networks. For instance, SNA measures have been used to identify the leaders in communication network (Joshua Tyler, 2005), to analyze the organizational structure considering the community structure and interconnections (Ryan Rowe, 2007), used to study the collaboration patterns of teams (Ehrlich & Chang, 2006), the role of brokers (Marczak, Damian, Stege, & Schröter, 2008)(Marczak & Damian, 2011b), to identify emergent people in requirements-driven collaboration among software development teams (Kwan & Damian, 2011), communication among teams (Ehrlich, Valetto, & Helander, 2007) and impact of distance of team's communication (Damian, Marczak, & Kwan, 2007), information flow among software development teams (Marczak et al., 2008), impact of communication structures on software teams performance and outcome quality (Cataldo & Ehrlich, 2011), and team performance (Ehrlich et al., 2014).

The inter relationship between members are represented by Sociograms. The sociograms are maps, graphic pictures or images, of a kind of relationship they are the illustrations of a relationship at a point in time. A sociogram consists of actors as nodes and their relationships as ties between them. The sociograms are used to visualize the relationships between teams as a group of nodes and strength of ties among them. There are several commercially available software packages equipped with network visualization along with analysis and simulations i.e. Gephi (www.gephi.com), UCInet (Borgatti, Everett, & Freeman, 2002), etc.

Collaboration among agile teams has been studied using SNA measures (for example (Cataldo & Ehrlich, 2011)). Cataldo and Ehrlich investigated the impact of the role of hierarchy and small-world communication structures on iteration performance and quality in a large distributed agile team. They found that there was a strong positive effect for hierarchy but a marginal negative effect for small-worlds on team performance, and a negative effect for hierarchy but a very strong positive effect for small-worlds on quality.

Therefore, it can be seen that usage of SNA measures to assess the collaboration behaviors of agile teams is still an under researched area and that is focused in this research.

2.3.2. Socio Technical Congruence

Another approach to investigate coordination among teams is by examining the congruence between the technical dimension of work and the social relationships team members establish, called socio-technical congruence (STC) (Cataldo, Herbsleb, & Carley, 2008). The mismatch is defined as "gap" that indicates poor coordination. STC is described as a relationship between social and technical dependencies with a team working on certain set of interdependent requirements. In (Kwan & Damian, 2011) awareness is studies in an organization having distributed culture using STC and results showed an improvement in the proposed STC model whereas the empirical results concluded how awareness can be achieved among team members. This congruence from literature is used to study socio-technical aspects i.e. Communication among cross functional RCTs (Marczak, Kwan, & Damian, 2009). The basic purpose was to indentify the gaps their characteristics and roles involved in those gaps.

The examples from literature show that to study collaboration among teams and to find out the gap in coordination among teams working together on interdependent workable items Socio-Technical Congruence (Kwan & Damian, 2011)(Marczak et al., 2009) and social network analysis measures can be used.

2.3.3. Approaches and frameworks in literature

There are a handful of studies in literature in which a set of steps were coined to study socio-technical aspects among software development teams. Such studies from literature include: (i) Damain and her colleagues work (Damian et al., 2010) in which an approach was proposed to study requirements-driven collaboration based on the concepts of Social Network Analysis for traditional software development teams; (ii) Pikkarainen, Haikara, Salo, Abrahamsson, & Still (2008) presented a theoretical framework to study impact of communication in agile teams. The literature provides a set of steps to study socio-technical aspects of collaboration i.e. communication and fleeting knowledge supported by empirical investigations from software industry for traditional teams (Damian et al., 2010) and empirical analysis of agile team to find impact of communication (Pikkarainen et al., 2008).

Unfortunately, not much about studying socio-technical aspects in agile teams is available till date. Therefore, the proposed framework is an initiative to provide a formalized set of steps to study socio-technical aspects (i.e. communication and awareness) among agile teams and empirically supported by four exhaustive case studies. The studies from literature remained silent on several issues like performance of teams and its applicability in variable development atmospheres like agile methods with high rate of change acceptability. Hence, it justifies the need of proposing this framework for agile teams with high collaboration, requirements volatility and small teams. Therefore, a formal set of steps were introduced to study collaboration among agile teams in terms of their communication and knowledge of each other.

2.4. Agile Software Development Methods

Agile software development is based on a set of principles. It comprises a group of several iterative and incremental software development methods with focus on collaboration between cross-functional and self-organising teams. Unlike traditional software development methods, agile methods focus on collaboration and interactions rather than processes (Highsmith & Fowler, 2001).This approach enables agile software development to be able to cater to today's fast-growing industry needs by having short development lifecycle, speedy development process and constant interaction with customers (Xianfeng, Kun, & Xiu, 2008).

2.4.1. Scrum

Scrum was introduced in 1995 as a lightweight, project management-oriented method based on several theories of system dynamics and complexity (Nonaka & H. Takeuchi, 1995). In Scrum, sprint is the basic unit of work spanning from one week to one month. The self-organising teams work in iterations called sprints with high degree of self-management and decision-making power vested in operational level, unlike traditional control and power- oriented methods (Moe, Dingsøyr, & Dybå, 2010). The project is divided into several sprints or iterations depending upon its complexity. During each sprint, the team works on several product features, i.e. user stories or use cases and deliver a shippable product at the end of that iteration. Sprint planning is done at the start in order to specify and prioritise the features to be worked on. The team members discuss their progress and problems in daily stand-up meetings called daily scrums. The work flow of scrum methodology is illustrated in Fig. 1 below. Scrum offers project management, versioning and requirements traceability through certain artefacts such as product backlogs, burn-down charts and sprint backlogs.

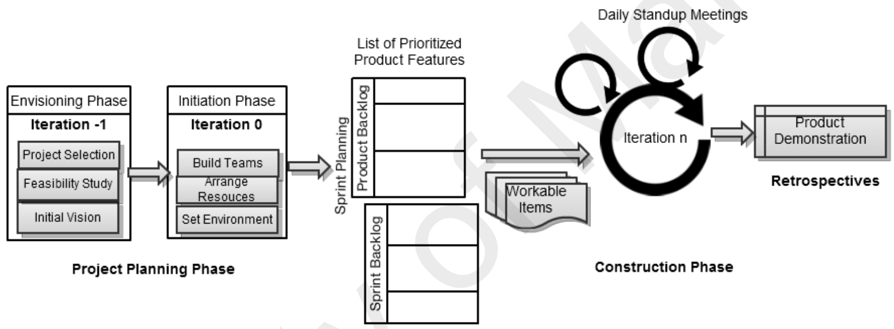


Figure 2.2: Illustration of scrum methodology workflow

2.5. Agile Requirements Engineering

The term 'agile requirements engineering' is used to define the "agile way" of planning, executing and reasoning behind requirements engineering activities. A systematic literature review was conducted to understand how traditional requirements engineering issues have been resolved through the agile software development approach and what new challenges have been created through the adoption of agile requirements engineering. The review consisted of main steps including planning, conducting and reporting the review results based on the guidelines proposed by Kitchenham & Charters (2007) .

2.5.1. Planning the systematic literature review

The review was planned by proposing research questions relevant to the topic of interest and search strategy, search string and inclusion/exclusion criteria were defined. The details are presented below.

2.5.1.1. Review objectives and questions

Although there have been several contributions to the field of incorporating traditional RE activities into agile software development, there is a lack of a coherent and consolidated views on the topic. Therefore, the main goal of this work is to develop an understanding of agile RE including the practices and challenges. The research questions guiding this review are:

RQ1. What are the adopted practices of agile requirements engineering according to published empirical studies?

RQ2. What are the practical challenges of agile requirements engineering?

2.5.1.2. Search strategy

After defining the research goals and questions, the search strategy was formulated to analyze all available materials specific to the objective of this review. The plan involved defining the search space, which included electronic databases and printed proceedings including as shown in Table 2.1. Moreover, reference search (snowballing) was performed to identify other meaningful studies and DBLP database was also searched.

Table 2.1: Search characterization

Electronic Databases	ACM digital library IEEE xplore, springerlink, EI compendex, Inspec ISI web of science, ScienceDirect, Taylor & Francis Online	**Search Items**	Journals, workshops and conference papers
		Search applied on	Full text
		Language	English
		Publication period	Up to June 2013

2.5.1.3. Search string

An example of a search done in the electronic databases is shown below:

Software AND (agile OR agility OR scrum OR “XP” OR “extreme programming” OR fdd OR “feature-driven development” OR “feature-driven” OR tdd OR “test-driven development” OR “test-driven” OR lean OR kanban) AND (“requirements engineering” OR “requirements” or “user story” OR “feature” OR “prioritisation”)

2.5.1.4. Inclusion and exclusion criteria

The following inclusion and exclusion criteria were used for the review:

Table 2.2: Inclusion and exclusion criteria for the review

Inclusion criteria	Exclusion criteria
(I1)The study is a peer-reviewed publication	(E1)*S*tudies that do not focus explicitly on agile methods, but only as a side topic
(I2) the study is in English;	(E2)Studies that do not discuss RE in agile methods;
(I3)It is relevant to the search terms defined in Section 2.4.1.3	(E3)Studies that do not meet inclusion criteria
(I4)It is an empirical research paper, an experience report, or workshop paper;	(E4)Opinion, viewpoint, keynote, discussions, editorials, comments, tutorials, prefaces, and anecdote papers and presentations in slide formats without any associated papers.
(I5)The study is published prior to June 2013.	

2.5.2. Conducting the review

In this section, findings of search from relevant sources are presented.

2.5.2.1. Search and study selection

In round 1, 543 studies were retrieved as shown in Table 2.3. As a result of applying the inclusion criteria of classification, 51 candidate studies were selected. Then, in Round 2, the pre-selected studies were assessed in order to apply the exclusion criteria. Out of the 51 studies pre-selected, 23 were excluded on the ground that they did not discuss any topic directly related to the scope of investigation (E1 to E4). Therefore, final selection consists of 21 studies.

Table 2.3: Identified studies during the distinct rounds of systematic search

		Round 1		**Round 2**	
Database	**Retrieved**	**Included**	**Excluded**	**Included**	**Excluded**
ISI Web of Knowledge	63	10	53	3	7
Wiley	12	4	8	4	0
Emerald	19	3	16	1	2
Springer Link	27	5	22	4	1
Taylor & Francis Online	12	1	11	1	0
Science Direct	63	7	56	3	4
IEEE Xplore	168	14	154	4	10
ACM	179	7	172	1	6
Total	**543**	**51**	**492**	**21**	**30**

2.5.2.2. Data extraction and synthesis

According to the guidelines provided by Kitchenham & Charters (2007), a data extraction process was defined to identify relevant information from the 21 included studies that pertain to the research questions. The data extraction process includes the following: First, a form was set up to record ideas, concepts, contributions, and findings of each of the 21 studies. Using this form ensures subsequent higher-order interpretation. The following data were extracted from each publication: (i) review date; (ii) title; (iii) authors; (iv) reference; (v) database; (vi) relevance to the theme, i.e. agile requirements engineering issues, challenges, practices, models, methods, techniques; (vii) methodology (interview, case study, report, survey); (viii) data analysis; (ix) validation techniques; (x) future work; (xi) limitations; (xii) country/location of the analysis; and (xiii) year of publication.

2.5.3. Findings of Our Review

In this section, the findings of the review in light of proposed research questions are presented.

2.5.3.1. Overview of Studies

Of the 21 studies, about 57% (12 of them) were published in conferences, 19% (4 of them) in journals, 19% (4 of them) in workshops, and 5% (1 only) in a magazine. Regarding the topics that formed the focus of the 21 studies, 29% are on agile RE practices, 28% are on newly proposed ideas in the form of methods. With respect to techniques and models for agile RE, only 5% are based specifically on the comparison of traditional RE with agile RE while the remaining 38% of the studies discuss agile RE in general.

2.5.3.2. Agile RE practices

While answering (RQ1) 16 RE practices adopted by agile methods were identified. It is important to note that the list below is inclusive, i.e. it reflects what have been collectively found in the 21 studies. Frequency of occurrences and the studies reporting each of the practices can be found in Table 2.4.

Table 2.4: Summary of the frequency of practices in the selected studies

Practice	Studies that Reported the Practice	Freq.
1.Face-to-face communication	(Cao & Ramesh, 2008)(Ramesh et al., 2010) (Jun et al., 2010)	3
2.Customer involvement	(Daneva et al., 2013) (Cao & Ramesh, 2008)(Ramesh et al., 2010)	3
3. User stories	(Paetsch, Eberlein, & Maurer, 2003)(Bjarnason, Wnuk, & Regnell, 2011a)	2
4.Iterative requirements	(Cao & Ramesh, 2008)(Ramesh et al., 2010) (Jun et al., 2010)	3
5.Requirements prioritisation	(Cao & Ramesh, 2008)(Ramesh et al., 2010) (Daneva et al., 2013) (Jun et al., 2010)(Racheva, Daneva, & Buglione, 2008)	5
6.Change management	(Cao & Ramesh, 2008)(Ramesh et al., 2010)	2

7.Cross-functional teams	(Bjarnason et al., 2011a)	1
8. Prototyping	(Cao & Ramesh, 2008)(Ramesh et al., 2010)	2
9.Testing before coding	(Cao & Ramesh, 2008)(Ramesh et al., 2010) (Jun et al., 2010)(Haugset & Stalhane, 2012)	4
10.Requirements modelling	(Boness & Harrison, 2007)(Ernst, Borgida, Jureta, & Mylopoulos, 2014)	2
11.Requirements management	(Cao & Ramesh, 2008)(Ramesh et al., 2010)	2
12.Review meetings & acceptance tests	(Cao & Ramesh, 2008)(Ramesh et al., 2010)	2
13.Shared conceptualisations	(Nik Nailah Binti Abdullah et al., 2011)	1
14.Pairing for requirements analysis	(Yu & Sharp, 2011a)	1
15. Retrospectives	(Cao & Ramesh, 2008)(Ramesh et al., 2010) (Jun et al., 2010)	3
16. Continuous planning	(Jun et al., 2010)	1

1. **Face-to-face communication** between team members and client representatives is a characteristic of agile methods. Frequent face-to-face communication leads to informal communication among stakeholders, which aids in the evolution of the requirements.

2. **Customer involvement and interaction** were declared the primary reasons for project success and limited failure (Eberlein & Julio Cesar, 2002). Agile methods rely on frequent collaboration (e.g. face to face (Cao & Ramesh, 2008; Ramesh, Baskerville, & Cao, 2010)) with an accessible and available onsite customer (Beck et al., 2001).

3. **User stories** emphasise “user goals”, briefly explain the user perception, focus on “what” is needed to be done, and support collaborative and iterative development (Carlson & Matuzic, 2010). User stories are created as specifications of the customer requirements. User stories facilitate communication and better overall understanding among stakeholders (Daneva et al., 2013).

4. **Iterative requirements,** requirements unlike in traditional software development methods, emerge over time in agile methods (Ramesh et al., 2010). Frequent interaction among stakeholders leads to this iterative requirements approach. Such approach makes requirements clearer over time, strengthens relationships with

customer and allows requirements to evolve with less investment of time (Cao & Ramesh, 2008).

5. **Requirement prioritisation** is part of each iteration in agile methods. In traditional RE, prioritisation is performed only once before development commences; in contrast, in agile methods, requirements are prioritised continuously in each development cycle by customers who focus on business value (Cao & Ramesh, 2008), or on risk (Daneva et al., 2013).
6. **Change management** has proven to be a significant challenge for traditional approaches thus far. For agile RE, its dynamic nature offers the greatest benefit. The main reported changes in requirements are to add or to drop features (Cao & Ramesh, 2008). Frequent face-to-face communication between development teams and clients preclude the need for changes in subsequent stages. The clarity gained by clients helps them to refine their requirements, which contributes to less rework and changes in subsequent stages.
7. **Cross-functional teams** include members from different functional groups who have similar goals. Agile teams work together and make their own decision without being dependent on the central members i.e. project manager in traditional methods.
8. **Prototyping** is perceived as a simple and straightforward way to review requirements specifications with clients and to gain timely feedback prior to moving to subsequent iterations. It promotes quicker feedback and enhances customer anticipation of the product.
9. **Testing before coding** means to write tests prior to writing functional codes for requirements e.g. Automated acceptance test-driven development (ATDD) (Haugset & Stalhane, 2012) that combines features of both agile RE practices and traditional RE.

10. **Requirements modelling** is a technique used to model requirements such as goal-sketching, which intends to provide intuitive and easy-to-read goal graphs for stakeholders (Boness & Harrison, 2007). This technique empowers decision-making while requirements negotiating process is carried out.
11. **Requirements management** is performed by maintaining product backlog/feature lists and index cards (Cao & Ramesh, 2008)(Ramesh et al., 2010). In the Scrum method, product backlog can be used to keep track of requirement changes.
12. **Review meetings and acceptance tests** are the developed requirements and product backlogs that are constantly reviewed in meetings (Carlson & Matuzic, 2010); they are a form of checks and balances of the user stories completed and still in hand. Similarly, acceptance tests are just like unit tests, resulting in binary results of "pass" or "fail" for a user story. These acceptance tests increase team, customer and domain expert collaboration as well as reduce the severity of defects and regressions.
13. **Shared Conceptualisations** is a supporting concept to carry out RE activities related to gathering, clarifying and evolving for agile methods (Abdullah, Honiden, Sharp, Nuseibeh, & Notkin, 2011). The concepts are built and stored in each individual's memory through communication and collaboration during RE activities. The co-located agile teams constantly rearticulate their shared conceptualisations during development, which helps in problem solving.
14. **Pairing for requirements analysis** is a practice that encourages the stakeholders to perform multiple roles as well (Yu & Sharp, 2011a). A single stakeholder playing different roles can introduce efficient task sharing due to minimal communication delay.

15. **Retrospectives** are the meetings held after completion of an iteration (Carlson, Matuzic, & Simons, 2012). These meetings often review the work completed so far and determine future steps and rework.

16. **Continuous planning** is a routine task for agile teams (Jun, Qiuzhen, & Lin, 2010). The team never sticks to fixed plans and adapts to the upcoming changes from customers as the project progresses. This flexibility facilitates changing requirements in later stages of projects.

2.5.3.3. Challenges of Agile RE

To answer the (RQ3) What are the practical challenges of agile requirements engineering? the identified challenges of agile RE are described below.

1. **Minimal documentation** is a vital challenge that agile methods pose to development teams (Cao & Ramesh, 2008)(Ramesh et al., 2010). Whenever the requirements are supposed to be communicated to customers at distributed geographical locations and not collocated or onsite, it becomes cumbersome to tackle such a situation with little or no documentation (Goetz, 2002).

2. **Customer availability** is assumed and advocated by agile methods. However, this assumption is often unrealistic as empirical studies confirmed customer availability and access to be overall a challenge (Ramesh et al., 2010)(Pichler, Rumetshofer, & Wahler, 2006). In practice, most of the agile teams have surrogates or proxy customers to play the role of a real customer (for example, product owners (Daneva et al., 2013).

3. **Budget and schedule estimation** is a challenge for organisations that follow agile methods. Although practising agile methods enables the initial valuations of a project, it is not possible to make upfront estimations due to volatile requirements, dynamic planning and design (Ramesh et al., 2010).

4. Inappropriate architecture finalised by the team in earlier stages of the project becomes inadequate in later stages with new requirements (Ramesh et al., 2010). On the other hand, refactoring is an ongoing activity among agile teams, which keeps on changing the code.

5. Neglecting non-functional requirements It is considered as a major challenge for agile methods (Ramesh et al., 2010) that can cause rework.

6. Customer inability and agreement are the two main issues apart from availability according to (Daneva et al., 2013). The disagreement between customer groups affects the performance, especially in short development cycles (Ramesh et al., 2010).

7. Contractual limitations and requirements volatility are important by not allowing changes in requirements after the signing of contract; the changes can cause an increase in cost and sometimes failure of projects. Therefore, legal measures should be taken to avoid such a situation and appropriately handle the flexible nature of agile RE.

8. Requirements change and change evaluation is an important aspect of agile methods. The flexible nature of agile methods welcomes changes, but it can create trouble when evaluating the consequences of these changes.

2.5.3.4. Discussion of the results

An important aspect highlighted in the analysis of our 21 selected studies is the geographic locations of authors. It is observed that nearly 1/3 of all contributions were from North American countries (based in the US and Canada). This is unsurprising, considering the fact that the Agile Manifesto was created by North-American software development practitioners. The uneven distribution of authors across geographic regions means that the empirical evidence reported by the 21 studies could not be considered generalisable. As most of the evidence is provided through empirical research in organisations in North America and Western Europe, it is hard to predict the similarity in results if agile RE is practised in Asian or South American organisations. Based on our findings, it can be concluded that many organisations in Asia are still in the process of adopting agile methods and are in maturation. Therefore, it is recommended that researchers conduct more empirical studies on Asian organisations to report the findings from different parts of the world, where the culture in each location varies. In addition, this opens up an avenue to conduct comparative studies on the agile RE implementations across the continents.

Most of the studies reported in this review have used a case study-based research approach. This establishes the fact that researchers acknowledge agile methods as a social process and thus investigate it in its real-life context (e.g. through case studies) and not reproducing it in classrooms.

Furthermore, the current practices of agile RE in response to RQ1 were identified, which ensure the effectiveness of agile ways of dealing with requirements. However, it was noticed that these findings could not be traced to particular project contexts, for example large and very large projects versus small projects, or projects in specific industry sectors, e.g. government, healthcare. Hence, it proves that more research is needed to define specific practices for specific industry-based scenarios. An increased number of empirical studies are required to implement these practices and contribute findings to the existing body of knowledge.

Eight challenges of agile RE practices in response to RQ2 were identified. The challenges that agile RE poses to project organisations include minimal documentation, budget and schedule estimation, inappropriate architecture, neglect of non-functional requirements, waste management, customer unavailability and contractual issues.

2.5.3.5. Implications of the study

This review has several implications for both researchers and practitioners. The compensations and flexibility of the agile methods are also expected from the agile way of managing requirements. Thus, this should prompt the industry to adopt agile RE practices for even distributed large-scale projects. Active research can only be possible with industry participation. Thus, the industry should participate in research related to agile RE and target research goals that are highly related and beneficial for the future of the software industry. The empirical results generated from communication patterns and collaboration studies among agile teams should be utilised by project managers to orient their teams towards the implementation of certain collaboration structures, i.e. hierarchal or centralised among them for maximum performance.

In terms of research, the review shows that there is a need for more empirical studies that incorporate agile RE approaches using various variants of agile methods, e.g. Scrum, XP, lean, and Kanban. With the advancements in global software development, the examination of the performance of agile RE in outsourced projects with distributed teams and customers remains challenging. Thus, there is a need to implement agile RE in large distributed and outsourced projects to gain more insights.

2.5.3.6. Limitations of the study

To eliminate the bias and ensure precision and accuracy in study selection, following steps were implemented while developing research strategy. First, the search-string-building process was treated as a learning process that included experimentation. As Dybå & Dingsøyr (2008) indicate, search strings in software engineering are language dependent; thus, there is a possibility of missing relevant studies during each search.

Next, this review encompassed only articles that primarily focused on the agile method of addressing requirements and not on the following issues: (i) studies of requirements engineering in organisations that follow agile methods; and (ii) agile studies for the complete software development lifecycle, which include requirements as a small part. In that context, it might have been possible to include more data and draw more conclusions.

2.6. Summary

In this chapter, Agile RE and its underlying concepts are discussed. Agile RE has been investigated through a systematic literature review. The review was conducted by following available guidelines (Kitchenham & Charters, 2007). Of the 543 initial papers located in well-known electronic research databases, 21 relevant papers were extracted through a multistage sifting process with independent validation in each step. These papers were then further analysed and categorised into the following thematic groups based on the research questions: (i) commonly used practices of agile RE; and (ii) practical challenges of agile RE. This research provides future dimensions to industry and research practitioners for further work on agile RE.

CHAPTER 3: RESEARCH METHODOLOGY

This chapter explains the set of steps involved in achieving the proposed research objectives. Each phase of the research is explained in detail in this chapter.

3.1. Introduction

The process of seeking a solution to a research problem in software engineering consists of several steps. First, the research gap or problem is identified from existing literature based on its relevance with the research idea. The novelty and relevance of the research problem is determined in this phase. Then, problem understanding is gained and research issues are conceptualized, through brain storming and literature review. Research conceptualization gives face to thoughts and helps in determining the research objectives by breaking them down into research questions. The formulation of the research objectives gives rise to the question of how to achieve them. Therefore, the appropriate selection of the research methodologies is performed to fulfill these research objectives. In the next phase, a real study design is created and implemented to achieve the desired results. The design can be a model, framework, approach or some algorithm which can help to achieve the proposed research objectives. The implementation of the design comprises of step by step execution in order to answer each of the objective. After answering the research questions, validation is performed to check the applicability and utility of the results achieved. Validation is performed from two evident perspectives i.e. industry and research. The results are compared with previous research based on its similarities and differences from previously obtained results. The research is analyzed to determine whether it corroborates with the previous findings or provides some new insights. In addition, the usability and applicability of the research for industry practitioners are measured. The last step is to share the results with the research fraternity and industry practitioners.

This research also followed this general hierarchy of steps explained above, following the constructive research approach to fulfill the proposed research objectives.

3.2. Constructive Approach

This study employed the constructive research approach for achieving all of the set objectives. The constructive approach was used as the overall research approach in which other methods were also employed i.e. systematic literature review, survey, questionnaire and interviews. The selection of this approach was based on the structure and requirement of this research. The key concept behind the constructive method is to construct solutions on the basis of previously available ones and to provide the missing factors to support its construction. Gordana (2008) described some characteristics of the constructive research method as: Feasibility, how feasible it would be to solve a previously existing problem; Novelty, how a new solution can be devised for a pre-existing problem; and improvement, how a better solution can be devised than the previous one.

The basic structure of the constructive approach defined in the literature is (Lukka, 2002):

- Defining research problem that has relevance
- Gaining understanding over the research area
- Construction phase, construct or device a new plan or solution
- Testing and validating the solutions for correctness
- The theoretical connections of the solution and the new knowledge it provides
- Generalization of results

The research process followed in this dissertation is illustrated in Figure 3.1. The research process consists of the following steps: (a) brainstorming research ideas, (b) obtaining an understanding of the literature surveyed, and (c) constructing a solution followed by (d) validation and (e) generalization of results. Each step is explained in detail in Figure 3.1.

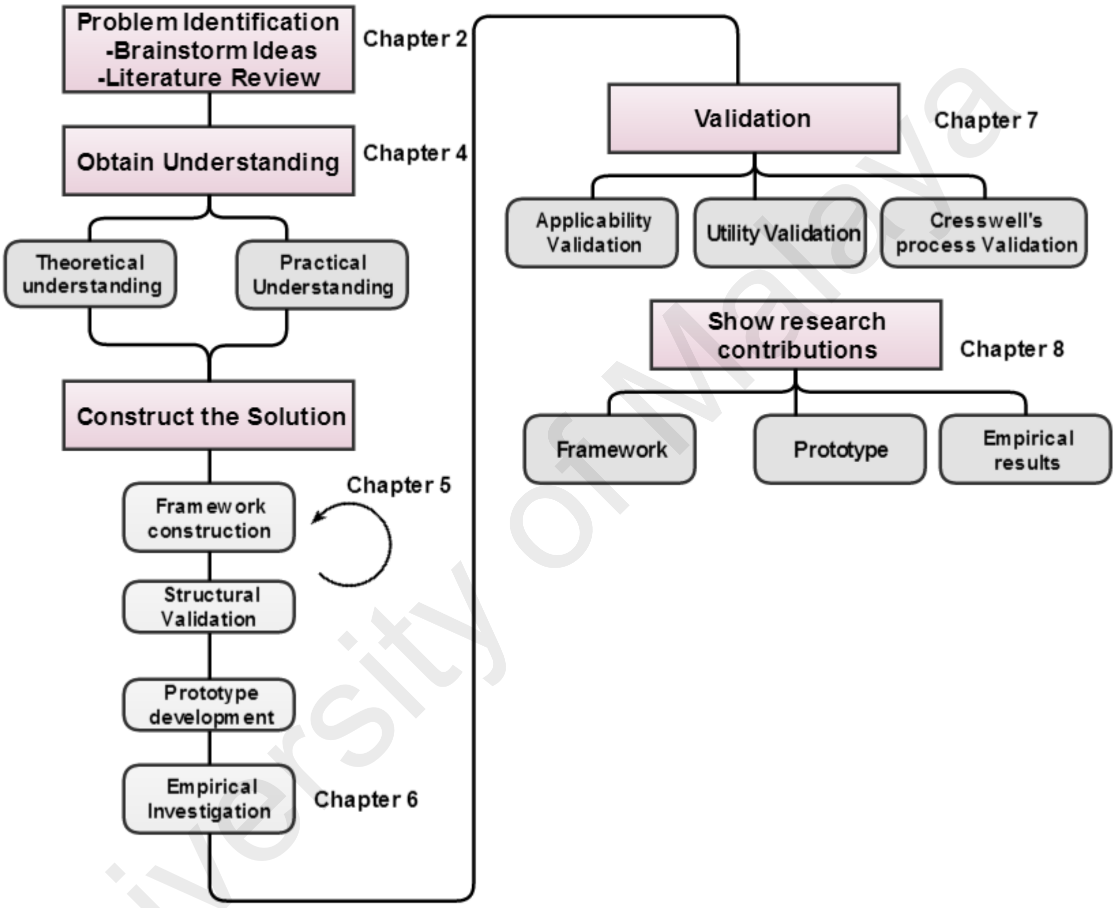


Figure 3.1: Research process followed for this dissertation

3.2.1. Brainstorm research idea

The research problem can come from colleagues, from the literature and from our own past research experiences (Lassenius, Soininen, & Vanhanen, 2001). Therefore, the first step is to brainstorm and find a relevant problem or idea. The process starts with hypothesizing the problem and later on visualizing the solutions. As the first step, the brainstorming of ideas related to requirements engineering in agile methods was conducted with my supervisors. Based on my supervisor's experience in this discipline and my own previous experience of working with agile methods, the study of the collaboration aspects among agile teams was pointed out and became the focus of our research idea. Later, the literature review was built around this theme to find the appropriate research issue for studying collaboration among agile teams.

3.2.2. Obtain understanding

The second step was to obtain an understanding of the problem area through theoretical and practical ways. The theoretical understanding comes from the literature and theoretical sources, known as literature survey of the particular area of research that helps in proving the novelty of the idea (Holmes & Gordon, 2002). On the other hand, the practical means of problem understanding can be a survey, interview etc.

3.2.2.1. Theoretical understanding

An extensive literature review was performed to gain an understanding of the problem area. The literature review was divided into two systematic literature reviews. The first one was focused on agile requirements engineering and its perspectives in order to learn about the currently available knowledge on the topic. The second one was conducted on the most relevant socio-technical aspects of requirements-driven collaboration. This way, it covered the area of interest in broader terms and helped in furthering the knowledge and insights in this area. The first systematic literature review is discussed in Chapter 2 and the second one is discussed in Chapter 4.

3.2.2.2. Practical understanding

After finalizing the theme of the research as "socio-technical aspects of requirements-driven collaboration among agile teams", the next milestone was to identify the most relevant socio-technical aspects to study. In order to obtain this practical understanding, a questionnaire based online survey was conducted to understand the perception of industry based agile practitioners on collaboration and its socio-technical aspects following survey guidelines (Kitchenham & Pfleeger, 2008). The survey implementation consisted of the following steps:

(a) Defining objectives

The objective was to further the understanding of the problem area by including the perception of agile practitioners from within the industry by making use of their practical experience and knowledge. Furthermore, as per the discrepancy highlighted by the literature review, it was the prime objective to identify the important and relevant socio-technical aspects of requirements-driven collaboration for agile methods in order to conduct further research.

(b) Designing survey

The survey was designed in compliance with the objectives. The survey questionnaire consisted of two sections. The first section was the demographics section in which information about the respondents' experience with agile methods and the role performed in the team was elicited. The second section was about the perception of the respondent's on the collaboration among agile teams and on the relevant socio-technical aspects of requirements-driven collaboration among agile teams. Both sections of questionnaire helped to filter out the required information in order to strengthen the understanding of the problem area and to pave the way for future investigation.

(c) Developing the survey instrument (questionnaire)

The questionnaire was divided into two sections to collect the required information as explained above. The questions were mainly based on Likert scale questions to gather quantitative information for analysis and avoid misinterpretation.

(d) Evaluating the survey instrument

The questionnaire was first evaluated using expert opinion from subject matter experts through the implementation of a pilot test, before the real execution of the survey, on ten experts (3 from industry and 7 from academia). The feedback was used to improve the questionnaire and remove any ambiguous questions.

(e) Obtaining valid data

The data obtained was in the form of Likert scale responses so that a lesser chance of misinterpretations and confusions would occur as in the case of open ended questions.

(f) Analyzing the data

The data was later analyzed through simple statistical measures, as required for obtaining the results and to ensure their reliability such as Cronbach's Alpha etc.

3.2.3. Constructing the solution

The solution construction phase is iterative and accumulates in two phases i.e. construction of the solution and iterative validation (Lassenius et al., 2001). For this research, a framework was proposed to study the socio-technical aspects of requirements-driven collaboration for agile teams in order to find their impact on performance. The framework was deduced from the literature mainly based on the work of Damian and colleagues (Damian et al., 2010). The framework was supported by an empirical investigation by conducting four case studies.

For this empirical investigation, a multiple case study method was adopted to gain deeper insights on the collaboration happening among agile teams over a course of time. The multiple case design was selected because it is better than a single case in terms of results generalization (Yin, 2003). In the literature, the multiple-case study method is used for literal replications-different cases can either be selected to produce same results, and theoretical replications- contradictory results can be attained due to change in conditions (Yin, 1994). Therefore, we have used multiple-case study research design in order to deduce more generalizable results by studying the similarities or differences between the cases which can be further implied to numerous settings.

3.2.3.1. Case study design

Multiple cases were selected based on certain criteria: (a) the project should be in the starting phase because the first iteration needs to be recorded, (b) the project life span must not be more than six months, (c) the iteration size must not be more than 1 month, and (d) the team has to be distributed, (e) language constraint-the communication medium of the teams has to be English. However, the optimal convenience of researchers to conduct the required on-site observation, interviews and interaction oriented data collection practices was also kept as a priority. Four projects were studied in four different organizations which are: Case Alpha (Malaysia and Finland), Case Beta (Malaysia and US), Case Gamma (Pakistan and UK), and Case Lambda (Pakistan and Philippines). This was due to the accessibility of physical presence that was made possible with the teams located in Malaysia and Pakistan for data gathering. However, the rest were observed virtually through daily scrum sessions via Skype and large interactive touch screens installed at the sites. After the selection of companies was made, data sharing and security documents were signed to ensure due privacy for the companies and their employees. Later, the data to be collected was finalized and data collection was deployed.

3.2.3.2. Data collection and analysis

Multiple data collection methods were used in this study. Document analysis was used to build the basis of this research. It helped to go through the feature list of some projects (only where the administration allowed), user stories, story cards, walls etc.

On-site observation helped to assess the work flow, collaboration practices of the team and their relationship with each other (e.g. it helped to find out for what reasons teams communicate with each other, how they communicate with each other formally and informally etc.). Questionnaire was the main instrument of data collection. It was deployed online and in paper format also. Semi-structured interviews were also conducted with some of the team members mainly the team leader(s) or project manager(s) to gain further information and fill the missing data in the questionnaires. Some of the confusions and ambiguities were later clarified through email responses of the team members.

Data were analyzed using social network analysis techniques and statistical measures. Social network analysis techniques were the main analytical approach adopted in this research supported by statistical measures where required.

3.2.4. Validation process

In this research, validation was performed to assess the applicability and utility of the proposed framework for the practitioners. To evaluate the proposed framework in order to find its potential application in the industry, application validity was performed. To evaluate the utility of the proposed solution, structured interviews were conducted with industry practitioners. In addition, eight validation strategies proposed by Creswell (2009) were used to validate the research process such as triangulation, member checking, rich and thick descriptions, clarify bias, report discrepant information, prolonged contact with participants, peer debriefing, and external auditor.

3.2.5. Examine scope of applicability

Defining the practical implications of the research study for industry practitioners and researchers is the last phase of the constructive research approach. In this phase, research generalization is performed after obtaining consolidated empirical results from case studies. In addition, practical implications are discussed for researchers and industry practitioners.

3.3. Summary

This chapter provides an overview of the research methods used for carrying out this research. The constructive approach was followed, in which, other research methods were used. The overall research approach is presented in this chapter along with the introduction of subsequent research methods used. A detailed description of all of the research activities performed is described in the forthcoming chapters.

CHAPTER 4: IDENTIFICATION AND REVIEWING THE MOST RELEVANT SOCIO-TECHNICAL ASPECTS IN AGILE TEAMS

In this chapter, there are two phases (shown in Figure 4.1). In phase 1, the most relevant socio-technical aspects of agile teams were identified through an online survey. In phase 2, the identified aspects are reviewed in literature with regards to requirements-driven collaboration through a systematic literature review.

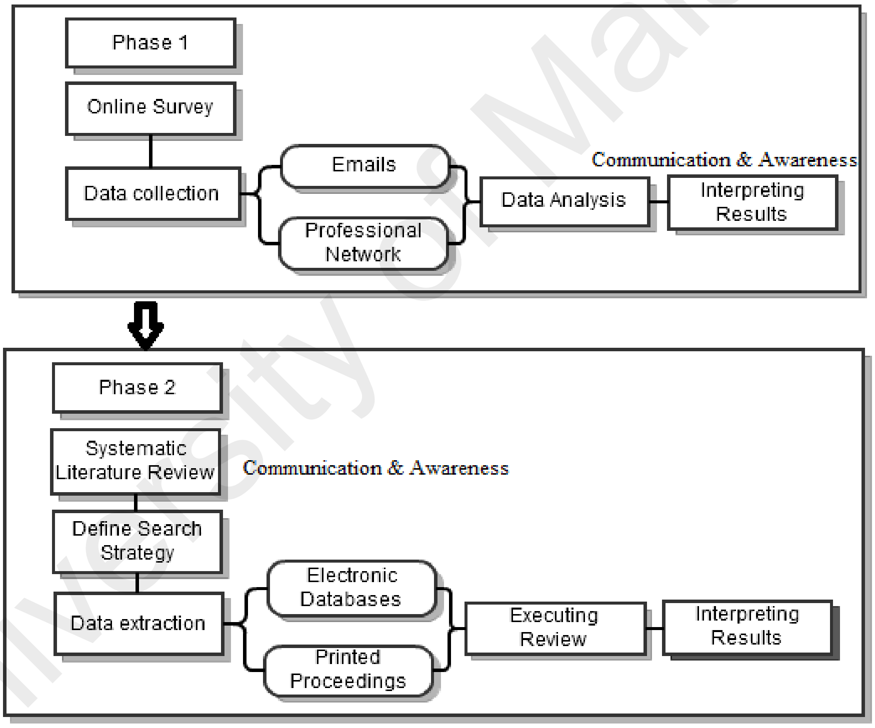


Figure 4.1: Research activities performed for identification and review of the most relevant socio-technical aspects of RDC among agile teams

4.1. Phase 1- Identification of the most relevant socio-technical aspects

The are several socio-technical aspects focused in the literature in particular for agile teams i.e. communication (Cataldo & Ehrlich, 2011)(Bjarnason et al., 2011b)(Ehrlich et al., 2014), trust (BredeMoe & Smite, 2008; Tjørnehøj, 2012), coordination (Mishra & Mishra, 2009), culture (Robinson & Sharp, 2005a), knowledge sharing (Dorairaj & Noble, 2013; Downs, Plimmer, & Hosking, 2012), and teamwork (Moe, Dingsøyr, & Dybå, 2010). The main objective of this survey study was to identify the important socio-technical aspects according to the perception of software professionals. The software professionals can better demonstrate their observations as per their experience regarding socio-technical dependencies whilst working in teams. This survey study helped in narrowing down the scope of investigation regarding socio-technical aspects to only two important aspects. This study helped us to define those socio-technical aspects considered by software professionals as the most relevant ones according to their perception of collaboration (Inayat, Salim, Marczak, & Kasirun, 2014). To achieve this specific objective answers to following questions were sought:

1. How do agile practitioners perceive collaboration?
2. What are the most relevant socio-technical aspects of requirements-driven collaboration for agile teams?

4.1.1. Survey

To fulfil the study aims questionnaire based survey method was adopted. The questionnaire consisted of both open and closed ended questions in combination to incorporate the survey design tactics suggested in literature (Royse, 2008). While designing the web-based questionnaire it was intentionally chosen to keep the number of open-ended questions lower to avoid missing data (Reja, Manfreda, & Hlebec, 2003).

4.1.2. Data collection

A twofold approach was used for data collection: (i) through emailing targeted communities and companies that use agile methods and its variants worldwide and (ii) by uploading questionnaire to related groups at a professional social network website, LinkedIn *(www.linkedin.com).* LinkedIn is a professional social network which owns millions of professional groups, communities, and users. It has been used for professional discussions, knowledge sharing, and job listings since early 2002. The survey questionnaire was posted to specific groups such as Agile, Agile and Lean, Agile CMMI, Agile Bangalore, Agile Project Managers, Extreme Programming (XP), Group Lean Brazil, Agile Project Managers Extreme Programming (XP), QA in an Agile World, Requirements Engineering, Scrum Alliance Inc., Scrum China, Scrum Gathering Orlando 2010, Scrum Manager, Scrum Practitioners, and Software Engineering Professionals.

The target population was agile practitioners working in industry and using agile methods, i.e. Scrum, Lean, Kanban, XP etc. Data was collected for 4 entire weeks in starting from May 2012 to June 2012. A total of 103 responses were collected from both of the sources. About three-quarters of the responses were gathered through the LinkedIn network while only about one-quarter were collected through e-mail. A total of 50 email invites were sent to agile practitioners from various parts of the world and 30 questionnaires were replied, out of which 4 were incomplete thus only 26 were considered.

4.1.3. Population

The survey was emailed to several software development companies including IBM (Pakistan, Malaysia, Sydney) Microsoft (Malaysia), Systemtech (Pakistan) and got (26%) responses from there. Rest of the (74 %) responses were gathered by uploading the survey on LinkedIn in software development Professional groups which congregated responses from all over the world. Figure 4.2 shows the distribution of the respondents with respect to their location.

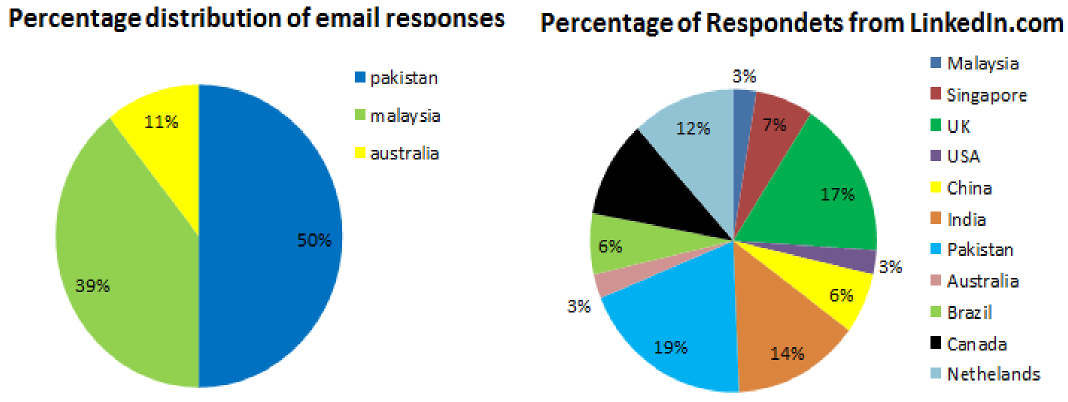


Figure 4.2: Percentage distribution of respondents' location

4.1.4. Questionnaire

The questionnaire investigated:

1. Respondent's background and work experience
2. Respondent's perception towards socio-technical aspects
3. Indication towards the socio-technical aspects (i.e., communication, organisational culture, etc.) that the respondents find promising in influencing software process, requirements or product quality.

First section was about demographic data, which included respondent's role in team, experience in the role, software development method in use and team structure (either distributed or collocated). The second section was about the perception of practitioners regarding collaboration among teams and socio-technical aspects of collaboration. In addition to this acuity about coordination with distributed and collated teams was also sought. The third section was to identify the important socio-technical aspects of collaboration in agile RE. This questionnaire had both close ended and likert scale questions. A 5 point likert scale was used to reflect the level of observation.

4.1.4.1. Procedure

Prior to the actual implementation of questionnaire (see Appendix A), a pilot survey was sent to three subject matter experts and 7 software industry practitioners working in industry to check the validity of content. The suggestions and reviews were incorporated in the survey before opening it to the targeted population. The main purpose behind was to improve the questions and avoid ambiguities in understanding of questions.

4.1.5. Survey results

The survey was distributed among respondents through two sources i.e. email and Linkedin.com a professional social network. The response rate of email based survey responses remained 57.7%. Total 46 emails were sent to different agile following companies' employees and 28 responses were received, 2 were discarded due to incomplete and inadequate information. The detailed discussion of responses received is described in this section.

4.1.5.1. Demographics

In this survey, the demographics section was about participant's present role and work experience in that role. The responses showed that most of the participants were experienced; 30% of the people have work experience of less than two years, about 45% fall in midlevel career between three to seven years and 25% of people are highly experienced, having more than 9 years in the field (shown in Figure 4.4). This indicates that the people participating in this survey were well versed in the field and that their opinions were worthy of consideration in this research.

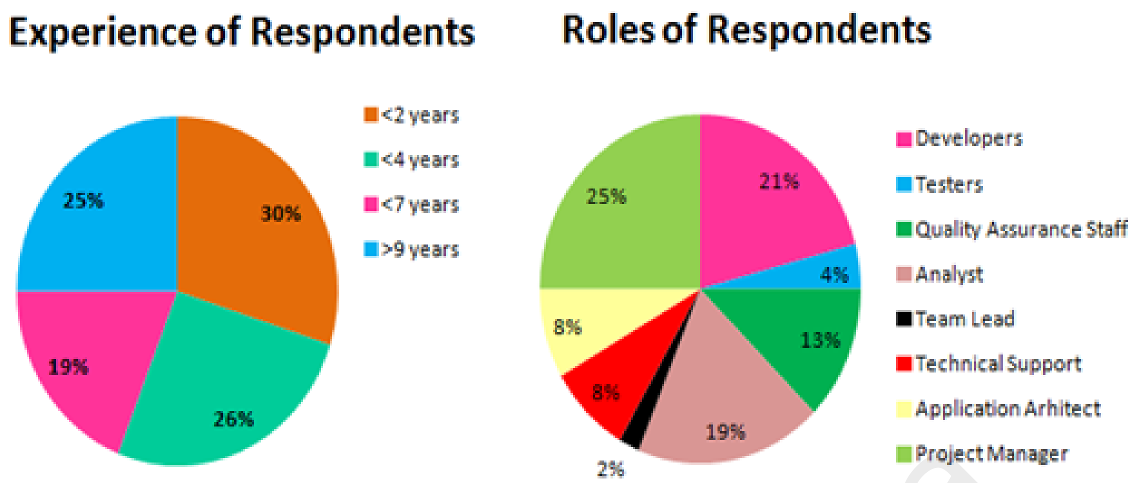


Figure 4.3: Role and experience of the respondents

The percentage of the roles of the participants are: 24% project managers, 18% analysts, 20% developers and the remaining 38% comprise team leaders, testers, technical support, application architects and QA staff, as shown in Figure 4.3.

4.1.5.2. Respondent's perception towards collaboration and socio-technical aspects

The perception of agile practitioners regarding collaboration and socio-technical aspects is tabulated and shown in Table 4.1. The responses are measures on a 5 point likert scale ranging from 1: strongly agree 2: agree, 3: neutral, 4: disagree, 5: strongly disagree. The results showed that agile practitioners found it easy to collaborate due to several interdependent factors, namely communication (Mean (M) =1.05, Standard Deviation (SD) =0.73), knowledge sharing (M=2.44, SD=1.21), previous work experience (M = 2.35, SD = 1.86), strengthened trust with each other (M = 1.84, SD = 1.25) and awareness of peers (M = 1.08, SD = 0.74), organizational structure (M= 2.11, SD= 0.36) and culture (M= 2.13, SD= 0.25). This gives rise to awareness about each other's work, expertise and status. It can be concluded from the results that agile team members find it easier to collaborate with their colleagues or counter parts with whom they can communicate easily, they are aware of previously and they can trust on to share information.

Table 4.1: Scale reliability and descriptive statistics of responses

Questions	Mean	SD	Cronbach's Alpha (α)
Collaboration is easier with people you can easily communicate with (Communication)	1.05	0.73	0.70
Collaboration is easier with people you can easily share knowledge with (knowledge sharing)	2.44	1.21	0.76
You can easily collaborate with people you can trust to share information (trust)	1.84	1.25	0.88
Collaboration is easier with people you are already familiar with (awareness)	1.08	0.74	0.91
Collaboration is easier with people following same organization hirer achy as yours (organizational structure)	2.11	0.36	0.77
Collaboration is easier with people following same culture as yours (culture)	2.13	0.25	0.79
(5 point Likert scale:1-strongly agree-5 strongly disagree) SD=standard deviation			

Cornbach's Alpha was used as a measure to check the reliability and internal consistency of data. Cornbach alpha is known as the most popular statistical measure for data reliability and internal consistency (Cronbach, 1951). The Cornbach's Alpha values depict reliability and internal consistency shown in Table 4.1. The respective values of Cornbach's Alpha ranges from (min 0.70 to max 0.91) acceptable to excellent levels of internal consistency (Table 4.1, last column).

4.1.5.3.Identification of the most relevant socio-technical aspects

The practitioners view regarding socio-technical aspects of collaboration whilst working in a team are shown in Table 4.2. The respondents were asked about each socio technical aspects i.e. communication, organization structure, knowledge sharing etc. individually; that how important role the socio-technical aspects play while collaborating with their colleagues. The respondents were asked to rate their perceived importance on a scale of 5 ranging from highly important to highly unimportant. The survey results show that two highly recognised socio-technical aspects among software development teams are awareness and communication (M = 1.21 and 1.06), where mean for both lies in between strongly agree=1 and agree=2, as shown in Table 4.3. The successive aspects of collaboration such as knowledge sharing (M = 1.80, SD = 1.06) was the third most relevant according to the agile practitioners. Among the rest of aspects, organisational structure, culture and trust agile practitioners considered trust and culture far more important than organizational structure. However, these priorities of respondents widely depended upon their personal experiences and could vary generally with change in circumstances.

Table 4.2: Scale reliability and statistical analysis of responses

Question	**Mean**	**SD**	**Cronbach's Alpha (α)**
Which of the following are most important aspects of collaboration among distributed teams?(5 point Likert scale:1-Most Important -5 Highly Unimportant)			
Communication	1.06	0.83	0.87
Knowledge Sharing	1.80	1.06	0.77
Awareness	1.21	0.88	0.89
Organisation Structure	3.45	1.10	0.79
Trust	2.20	0.60	0.75
Culture	2.93	0.96	0.81
Mean	2.10	17.52	0.81

The survey results show that most of the software professionals perceive collaboration as communication among roles and their awareness of peers. They view collaboration as how they communicate during work and the knowledge of with whom they need to communicate and collaborate, according to work status and availability. The survey results also depict that among teams of software professionals, communication and awareness are the two most recognised socio-technical aspects which are focused in this study. This shows that collaboration among teams is deemed as when teams communicate with each other and when they are aware of each other's general traits, availability, current task (Ehrlich & Chang, 2006) and work status. To check the reliability and consistency of collected responses the value of Cronbach's Alpha was calculated for all the questions ranging from (min 0.72 to max 0.90) acceptable to excellent levels of internal consistency (Table 4.2).

4.1.6. Threats to validity

While ensuring the reliability of responses gathered through survey, 100% accuracy cannot be claimed as the data is collected from sampled population. To ensure the construct validity a pilot survey was performed with 10 respondents (3 researchers and 7 industry practitioners). The aim was to improve the ambiguities in the questionnaire and make it understandable for the targeted participants. As the result of pilot survey questionnaire were improved and irrelevant or ambiguous questions were discarded.

To ensure internal and external validity, it was ensured that the participants were all experienced agile practitioners, with at least two years of experience (see Figure 4.4). Moreover, the respondents played diverse roles in agile teams i.e. 25% were developers, 21% were project managers, 19% business analyst and so on. Needless to mention here that most of the respondent's job nature was directly related to agile software development.

The scale reliability was measured to determine the internal construct validity by assessing the extent to which a set of questions measures a single latent variable. It can be seen that basic statistical measures were used on raw data obtained from the responses of questionnaires. Cronbach's alpha coefficient, the most-widely used index of internal consistency in social sciences was used to determine the consistency of responses.

4.2. Phase 2- Reviewing Communication and Awareness as the socio-technical foundation of RDC

Although socio-technical aspects of software development have been largely investigated in previous research, (e.g. (Damian, Izquierdo, & Singer, 2007; Dourish & Bellotti, 1992; Gutwin, Penner, & Schneider, 2004)), yet they have not been widely discussed in relation to requirements-driven collaboration. Therefore, collaboration driven by requirements, particularly on the relevance of socio-technical aspects, still warrants further investigation. One of the goals of this research was to fill this gap by identifying the relevant socio-technical aspects of collaboration driven by requirements in the context of agile development and conducting a systematic literature review of the identified aspects. The understanding about the identified aspects i.e. communication and awareness was deepened from the requirements engineering perspective by conducting a systematic literature review.

4.2.1. Systematic Literature Review

Phase 2 aimed at reviewing the identified relevant socio-technical aspects. It covered a systematic literature review about the two identified socio-technical aspects, i.e. communication and awareness. More specifically, the review was conducted to identify: (i) current knowledge about the incorporation of communication and awareness in requirements-driven collaboration; (ii) interdependence between communication and awareness; and (iii) implications for agile practitioners and the research community. These research questions were devised after identification of important socio-technical aspects in Phase 1 (section 4.1).

4.2.2. Review outline

The standards for conducting a systematic literature review (Kitchenham & Charters, 2007) define a set of steps such as planning the review by identifying research questions and defining research protocol; executing the review by literature search and extracting data; these steps are followed by quality appraisal, and reporting the results through analysis. These guidelines are duly followed in this review.

4.2.3. Planning and executing the systematic literature review

In this section, planning and execution of this systematic literature review are explained.

4.2.3.1.Research Questions

The questions that form the basis for this review are as follows:

1. What is currently known about communication as a relevant socio-technical aspect of requirements-driven collaboration?
2. What is currently known about awareness as a relevant socio-technical aspect of requirements-driven collaboration?

4.2.3.2.Search strategy

A proper study strategy was used to search for the targeted literature of interest. The first step was to identify the search space, i.e. electronic databases, printed proceedings, etc. The complete list is shown in Table 4.3.

Table 4.3: Data sources used for conducting systematic literature review

Search Space	**Search Item**	**Searched Items**	Journal, workshop and conference papers.
Electronic databases	Electronic Databases		
	ACM Digital library		
	IEEE Xplore		
	SpringerLink	**Search applied on:**	Full text
	EI Compendex	**Language:**	English
	Inspec		
	ISI Web of Knowledge	**Publication period:**	Up to June 2013
	ScienceDirect		
	DBLP		

These databases were explored to retrieve a number of studies related to the objective of this review through the titles and abstracts of articles. A reference-based search approach was then applied to find related materials. Subsequently, possible authors of retrieved studies were searched further using DBLP to find more relevant studies.

4.2.3.3.Search and selection of studies

Studies were searched, retrieved and categorized. At the beginning, 340 articles were retrieved from search engines. Many of the articles returned were redundant. The query included articles with full texts in most of the search engines; therefore, the number of articles retrieved was large. Later, 10 studies were selected by searching the authors and references of the retrieved studies (e.g. (Ehrlich & Chang, 2006) study was found through this method). A total of 167 redundant results (articles) were rejected out of 340 studies (details are shown in Table 4.4). The rest of 173 studies were parsed through titles, abstracts and conclusion with irrelevant studies discarded, leaving only 16 relevant studies**.** First a form to record the ideas, concepts, contributions, and findings of each of the 16 studies was set up (references shown in Table 4.5). Brief information contained in this form ensured subsequent higher-order interpretation. The following data were extracted from each publication: (i) review date; (ii) title; (iii) authors; (iv) reference; (v) database; (vi) relevance to the theme, i.e. agile requirements engineering issues, challenges, practices, models, methods, techniques; (vii) methodology (interview, case study, report, survey); (viii) data analysis; (ix) validation techniques; (x) future work; (xi) limitations; (xii) country/location of the analysis; and (xiii) year of publication.

Table 4.4: Results of the systematic search for primary studies

Database	**Retrieved**	**Included**	**Excluded**
ISI web of knowledge	19	2	17
Wiley	10	0	10
Emerald	19	0	19
Springer Link	27	0	27
Tylor&Francis Online	12	0	12
Science direct	63	2	61

IEEE Xplore	70	10	60
ACM	120	2	118

When the extraction process was completed, content analysis was used (Hsieh & Shannon, 2005), (Elo & Kyngäs, 2007) to characterise the focus (e.g. shared recommendations, lessons learnt) of each study. Lastly, the results of data extraction were assessed by using an inter-rater agreement among the researchers (Fleiss, Levin, & Paik, 2003).

Table 4.5: Overview of the Selected Studies

Publication Source	**Type**	**No.**	**Ref.**	**Focus of study**
ACM Communication	Journal	1	(Cataldo & Ehrlich, 2011)	Communication
IEEE Software		1	(Damian, Eberlein, Shaw, & Gaines, 2000)	Communication
IEEE Transactions on SE		1	(de Souza & Redmiles, 2007)	Awareness
4th IEEE International RE Conference	Conference	1	(Damian, Eberlein, Shaw, & Gaines, 2000)	Communication
International Conference on Global SE		1	(Ehrlich & Chang,2006)	Communication
15th IEEE International RE Conference		1	(Damian, Marczak, & Kwan, 2007)	Awareness
International Conference on Global SE		3	(Calefato, Damian, & Lanubile, 2007) (Kwan, Damian, & Marczak, 2007)(Damian et al., 2007)	Awareness Communication
16th IEEE International RE Conference		1	(Marczak et al., 2008)	Communication
19th IEEE International RE Conference		2	(Bjarnason et al., 2011b) (Marczak & Damian, 2011b)	Communication
IEEE International Symposium on Computers and Informatics		1	(Ahmad, Tahir, & Kasirun, 2012)	Communication
1st Workshop on Agile RE	Workshop	2	(Abdullah et al., 2011) (Bjarnason et al., 2011a)	Communication
Institute for Software Research Carnegie Mellon University	Technical report	1	(Cataldo & Ehrlich, 2011)	Communication

Studies were initially examined through their abstracts and titles according to the inclusion criteria. 167 redundant studies were excluded from the total of 340 studies retrieved, leaving a total of 173 studies; this was followed by the exclusion of 157 studies that were found irrelevant and not in line with the aims of this study. Hence, the remaining 16 studies were selected based on the relevancy of the contents to the answers of the research questions. The records of these studies were kept in a spreadsheet for easy access and retrieval.

(a) Inclusion and exclusion

Sources for the relevant research were queried and searched. The inclusion criteria adopted in our review is represented by the following Boolean expressions:

A1 = Requirements Engineering OR Requirements OR Requirements Engineering process A2 = Communication, A3 = Awareness, B1=Interaction, B2=Collaboration

A1 AND (A2 OR A3) AND (B1 OR B2) Equation 1

It was then determined whether the selected studies fulfilled the criteria set for inclusion. The inclusion criteria are as the following: (I1) the study is a peer-reviewed publication; (I2) the study is in English language; (I3) the study meets the mandatory requirement of A1 and B1 or B2; the study satisfies the requirement of A2 or A3, meaning the paper could include the topic of either communication or awareness; (I4) the study is an empirical research paper, an experience report, or workshop paper; and (I5) the study was published between 2000 and June 2013. The exclusion criteria adopted were the following: (E1) the studies discuss communication and awareness as a sub-stream; (E2) the studies discuss communication and/or awareness for other phases of software development, such as design, coding, testing, etc. and (E3) articles that consist of opinions, viewpoints, keynote, discussions, editorials, comments, tutorials, prefaces, and anecdotes as well as presentations in slide formats without associated papers.

4.2.4. Results and discussions

In this section, the overview of the selected studies and review the studies related to both communication and awareness is discussed.

4.2.4.1.Overview of the studies

It is observed that most of the studies were papers presented in conferences (10 out of 16), a few were published in Scientific Journals (3 out of 16) and delivered in workshops (2 out of 16) and 1 was published as a technical report. Out of the conference studies, a quarter (4 of 12) of papers were presented during the International Conference on Global Software Engineering 2006 and 2007; a quarter (4 out of 12) of papers were presented during the IEEE International Conference on Requirements Engineering in 2007, 2008, and 2011. The geographic distribution of authors shows that most of the selected studies were written by Canadians (57%) and Americans (11%). The percentage of authors who originated from Europe (Italy, Sweden and UK) is around 24%; and only 8% of the authors were from Asia (Malaysia and Japan). These statistics show that particular interest groups in the USA and Canada had worked on certain research issues related to our research theme for this review.

It is noted that all of the selected studies used case study as the main research method. The secondary research methods used for further data collection include observation (7 of 16), interviews (4 of 16), document analysis (4 of 16), questionnaires (3 of 16), online survey (1 of 16) and ethnography (1 of 16). 3 of the 16 studies did not explain the secondary research method explicitly. Almost 69% of the cases (9 of 16) in the selected studies were industry based and 31% of them (5 of 16) were conducted on groups of university students. 6 groups of students were selected in those 31% of studies for the purposes of recording data and conducting experiment. There is only one study (de Souza & Redmiles, 2011) with multiple cases involving three companies. It can be seen that the researchers had observed the social aspects using appropriate research methods to encompass the collaboration between team members with time.

4.2.4.2.Communication in RDC

(RQ1) What is currently known about communication as a relevant socio-technical aspect of requirements-driven collaboration? It is observed that communication has been discussed in the literature based on the following two paradigms in requirements engineering: (i) the effects of communication media in requirements engineering; and (ii) the communication patterns among global teams. Requirements engineering is considered as the most communication-intensive activity in software development (Calefato et al., 2007). The dominant reasons for the necessity of communication among teams, as reported in the literature, are discussions about changes in requirements, clarification of ambiguities, coordination of activities, requirements negotiation and synchronisation of codes (Marczak et al., 2008).

(a) Communication Media

In the literature, communication is discussed from dual perspectives in the context of requirements engineering, i.e. communication media and communication patterns or structures as well as their effects on other socio-technical aspects, such as group performance, awareness, etc.

The importance of virtual means of communication, including video resources for requirements negotiations, was discussed by Damian and colleagues in two of their studies. Damian and colleagues conducted a study focusing on the effects of communication media, i.e. rich or virtual, on a group's performance in both co-located (face-to-face) and distributed requirements negotiations (Damian et al., 2000). Both studies used the following five different settings: (i) all team members were co-located; (ii) the two customers interacted remotely; (iii) the two customers were co-located; (iv) the developer interacted remotely; and (v) the facilitator interacted remotely. It was concluded in the first study that among virtual means of communication, video channel remained the most useful and promising method, as it enabled the users to be aware of the emotional state of other team members and provided interpersonal awareness (Daft & Lengel, 1984). In the second study, they reported that groups using rich communication media did not perform better than those using leaner communication media due to the lack of trust and interpersonal relationships.

Requirements communication among six teams of multi-cultural and geographically dispersed students was studied in (Damian et al., 2006). The teams of students were dispersed among universities in Canada, Italy, and Australia. Each team was distributed in two countries. It was concluded that collaboration became more effective with asynchronous discussions or communication between the teams, which was evidenced by the lower number of open issues after the discussions. The decrease in the number of open issues led to conflict resolutions and forging of agreements, which were achieved as effective communication was maintained among the geographically dispersed teams.

An empirical study was conducted to compare the results of using text-based communication in distributed requirements workshops with the results of face-to-face communication (Calefato et al., 2007). The effects of computer-mediated communication for requirements negotiation and elicitation were also discussed. Six teams of students were involved for the directed lab experiment. The students were located in distributed locations. The results revealed that computer-mediated communication was more effective for requirements elicitations than negotiations in regard to stakeholder's satisfaction with performance. On the other hand, users were more inclined to resort to face-to-face communication for requirements negotiations. Later, an empirical study was conducted to identify the effectiveness of communication media with respect to information richness needs for requirements elicitation and negotiations (Calefato, Damian, & Lanubile, 2012). Contrary to common belief, it was concluded that preference for face-to-face communication does not apply to all cases, and group performance is not affected by communication means. Requirements documents, the end product for directed lab experiments incorporating students as actors in this study, were produced for users of all kinds of communication media. Recently, another study was conducted based on Calefato's work in which face-to-face, text-based and rich communication media were tested for participants' performance and satisfaction (Ahmad et al., 2012). Directed lab experiments were used and involved student teams. The study provides evidence that face-to-face is the most preferred mode of communication among users for requirements communication and negotiation. This finding does not agree with the results of (Damian et al., 2000), which deviates from the traditional myth that groups perform better with rich means of communication, i.e. face-to-face rather than leaner media. The causes and effects of communication gaps in large-scale software development organisations were explored in (Bjarnason et al., 2011b). An exploratory study was conducted in a large-scale market-driven software development company. Findings revealed that there are four main factors affecting requirements communication, namely scale, size and complexity of software project; temporal aspects, lack of continuity of requirements awareness; common view, common goals; vision and decision structures, unclear or weak goals leading to unstable requirements. Findings suggest that communication gap results in failures to meet customer requirements, quality issues and rework, all resulting in waste of effort. Later on, agile requirements engineering approach was found viable to solve problems like communication gaps (Bjarnason et al., 2011a).The suggested solutions are as follows: i) gradual detailing of requirements, i.e. finalising the requirements only when they are about to be implemented makes them more stable; ii) representing requirements as user stories, i.e. enhancing communication among stakeholders and clarifying user's perception; iii) cross-functional development teams; and iv) integrated requirements process.

(b) Communication patterns of Requirements Engineering teams

The study of structure of communication patterns through social network analysis reveals the dynamics of collaboration within a team. Here, studying communication patterns and examined their effects on awareness, team dynamics, social climate, and group performance was emphasized.

It was noticed while studying communication patterns among global software development teams that the people who are central in communication and peripheral networks are more productive and their perception sets the tone for a positive environment in teams (Ehrlich & Chang, 2006) . Moreover, communication patterns study revealed positive effect of communication over awareness among team members (Kwan et al., 2007). The communication patterns of software development teams help to identify the information flow among distributed teams, role of brokers (Damian et al., 2007) and major reasons of communication among teams (Marczak & Damian, 2011a) through studying the Requirements Centric Social Networks (RCSN). However, communication patterns were studied to find their impact on iteration performance and outcome quality for agile teams (Cataldo & Ehrlich, 2011).

4.2.4.3.Awareness in RDC

(RQ2) What is currently known about awareness being a relevant socio-technical aspect of requirements-driven collaboration? Awareness is discussed in the literature from two main perspectives in the paradigm of requirements engineering, which are: (i) factors that affect awareness; and (ii) factors that are affected by awareness.

(a) Factors that affect awareness

In this section, the studies found in literature based on the factors that affect awareness in RDC were described. The findings are organised according to the factors rather than listing of studies with those factors; this arrangement facilitates identification of the factors mentioned in literature.

1. Communication structure, distance and experience of team members In (Kwan et al., 2007), factors affecting awareness among distributed teams were studied. The factors identified were communication structure among teams, distance and experience of team members. A case study was conducted in a US-based software organisation operating in Brazil with teams distributed in both countries. To study the factors responsible for disrupting awareness among teams, a group of 39 people were interviewed and the email inboxes of 5 people were studied. It was observed that the following factors affect awareness: (1) distributed development; (2) experienced team members; and (3) communication structure. It is suggested that a centralised communication system can prevent awareness problems.

2. Organisational culture Damian et al., (2007) studied task-level awareness in globally distributed software development teams. The aim was to study the effects of organisational culture on awareness while having distributed teams spread across three continents. It was concluded that organisational structure has significant effects on awareness of team members regarding certain tasks. In addition, awareness needs to be maintained throughout the process, as social networks revolve around requirements, which are dynamic and change with time.

3. Project growth The awareness networks are considered fluid, whereby components and size keep changing as a project evolves (de Souza & Redmiles, 2011). With the age of project, the actors involved develop knowledge about it and gain understanding about other actors, resulting in growth of the awareness network. Thus, the age of a project is also one of the important factors influencing awareness.

(b) Factors affected by awareness

These are the factors that are affected by the existence or lack of awareness. In this section these factors as per our review of literature were described.

1. **Frequency of communication** In (Ehrlich & Chang, 2006), awareness affects the frequency of teammates communicating with one another in a distributed software development. The investigation focused on different types of awareness, such as availability of other persons, awareness of their current task, general awareness about them and importantance placed on communicating with them. A case study was conducted among three teams working at distributed locations (n1 = 19, n2 = 39 and n3 = 79). The findings report that people communicate more with colleagues whom they have already known, provided that this relationship is positive. Similarly, it was observed that people communicate more with co-located teammates than the ones present at remote sites, due to the informal communication links that have been established. Thus, communication frequency is affected by team members' awareness of one another and the roles of people inside and outside the team.

2. **Knowledge acquisition** In (Damian et al., 2007), the impact of distance and communication on awareness among distributed teams working on an interdependent

set of requirements was investigated. Analysis of the teams was done using SNA measures. The aim of the investigation was to identify the aspects of awareness among distributed teams; specifically, it was to find out how awareness evolves, how it contributes to the requirements and how it is affected by communication and distance. The types of awareness considered were current awareness, general awareness and availability as well as their correlation with communication patterns in the project. The findings show that people who are familiar with one another communicate more often.

4.2.5. Discussions of the findings

About 67% of the studies were published as conference papers. This raises the question of validation of the results as in most of the conference papers, ideas are proposed and validation is not a must. Therefore, attention is drawn towards the fact that although social aspects like communication are discussed extensively in literature, the need of more empirical studies with proper validation and verification of results is still there. The geographic distribution shows that about 68% of the authors are from North America, i.e. Canada and USA. About 24%, the second highest percentage of the authors, are from Europe and only 8% of them are from South-East Asia and Far East Asia. There are significant cultural differences between Asian and European/American organisations and working styles differ as well. Therefore, more empirical studies from the Asian region were needed to generalise the results.

Case study was the widely used research method to explore the social aspects, i.e. communication and awareness among teams. This depicts the right choice of research method to investigate projects over the span of time.

The analysis of studies shows that past research was more inclined to explore communication as a social aspect of collaboration among software development teams. Approximately 69% of the studies focused on communication, 19% discussed awareness and the remaining 12% discussed both communication patterns and awareness. In answering RQ1 (Section 4.2.4.2), it is observed that the groups did not always use rich communication media such as face-to-face to achieve better performance.

Furthermore, in answering (RQ1), it was found that communication patterns are another popular dimension discussed in the selected studies. The study of communication patterns leads to very interesting results in terms of information flow, tendency of team members to communicate with their colleagues (Ehrlich & Chang, 2006), leading by awareness (Kwan et al., 2007), reasons of communication among teams (Marczak & Damian, 2011a) and their effects on teams' performance (Cataldo & Ehrlich, 2011).

The communication pattern studies reveal that team members are more likely to communicate with colleagues whom they have certain degree of awareness. This shows the interdependence between communication and awareness and vice versa. The centralised communication structure was suggested for effective information dissipation in order to update the knowledge of work status of all members among the teams; it can also be used to keep new entrants informed. It is necessary to conduct more empirical studies with different scenarios to confirm these findings. A centralised network structure may not be always effective as in the case of (Kwan et al., 2007). Therefore, more experiments with variable settings should be conducted to propose ideal network settings for certain teams.

In a communication network, the role of brokers cannot be ignored (Marczak et al., 2008). The information flow examined through communication pattrens determines the presence of information brokers in the networks and proves their significane. It can be seen that in geographically dispersed teams, brokers can be a good channel to transmit information among teams. This opens doors to analyse the functions of brokers based on various paradims in global and distributed software development with special focus on requirements management.

The communication patterns tend to provide useful information for dynamic teams (e.g. agile teams). In (Cataldo & Ehrlich, 2011), communication structures had been studied for the first time for agile teams. The study projected the evolution of communication patterns and their effects on iteration performance in agile teams. The hierarchical communication structure was found to be positively associated with iteration performance. The case report clearly indicated the narrowing of communication gaps, which led to reduced rework and improved quality of product. However, small world communication structures between closely knitted teams were not promising. Only one case was reported in this study. Therefore, more empirical evidence was needed to generalise the results at large. Moreover, this opens up to further investigation in certain other interesting cases of variable agile methods like Scrum, Kanban etc. The study of communication patterns of closely knitted dynamic agile teams will help further our knowledge about agile teams' behaviours and collaboration states over the growth of project.

In answering (RQ2), it was observed that software development teams, either distributed or co-located, are comfortable in communicating with people they are aware of. Increase in awareness helps in reducing communication breakdowns. It improves work practices, reduces cost by avoiding unnecessary communication and contacts, as well as leverages existing knowledge in addition to innovating new knowledge (Ehrlich & Chang, 2006). Thus, the main challenges of awareness in requirements-driven collaboration are scalability of coordination of requirements with respect to time and project size, fluidity of awareness networks, distributed teams, physical distance, communication infrequencies, organisation hierarchy and unavailability of required roles. Awareness is somehow built among members who communicate with one another or had worked together previously.

4.2.6. Implications for practitioners and research community

This review has several implications for software practitioners and researchers working in this field. The implications for industry practitioners are:

1. Affective means of communication: The review suggests that the software industry should use rich media of communication, i.e. virtual means of communication for attaining affectivity in requirements negotiations among distributed teams to reduce the communication gap, enhances awareness among teams and builds trust.

2. Hierarchical communication networks and hierarchical communication structures are linked to fewer coordination problems among distributed teams (Hinds & Mcgrath, 2006). Hierarchical communication structures measured for agile teams show positive effects on iteration performance by reducing communication overheads (Cataldo & Ehrlich, 2011). It induces a positive climate in teams and avoids unnecessary outflow of information. Moreover, small world communication structures have positive effects on quality (Cataldo & Ehrlich, 2011). This has implications for managers, who need to recognise evolving communication patterns and introduce the right kind of communication. Furthermore, this situation also encourages managers to assess communication patterns and to work on the missing links as per coordination needs.

3. Tools development: Comprehensive tools should be developed for management and collaboration of cross- functional teams with further features of traceability and automatic RCSN generation through pruning repositories. This would definitely help managers to identify communication lapse and powerful people in networks at any phase of project development.

4. For researchers, it is still a challenge to conduct more empirical studies based on real-world industrial cases, with the aim of observing other socio-technical aspects. Other socio-technical aspects like teamwork, organisational structure as well as trust in variable environments need to be examined. In addition, it is also novel to investigate the interdependencies of the social aspects of requirements-driven collaboration in variable scenarios.

4.3. Summary

This chapter has two phases. In phase 1 the most relevant socio-technical aspects of requirements-driven collaboration among agile teams were identified through an online survey and then the identified aspects were reviewed through a systematic literature review in phase 2. The survey results revealed that communication and awareness are the most relevant socio-technical aspects. However, the review findings highlight various aspects from which communication and awareness are discussed in literature.

CHAPTER 5 - A FRAMEWORK FOR STUDYING REQUIREMENTS-DRIVEN COLLABORATION AMONG AGILE TEAMS

This dissertation introduces the concept of studying the socio-technical aspects of requirements-driven collaboration (RDC) among agile teams. A framework is proposed followed by an empirical investigation to study requirements-driven collaboration among agile teams. This chapter explains the proposed framework.

5.1. Introduction

A conceptual or theoretical framework is described as a visual or written narration of the main things to be studied in the form of key factors, variables and presumed relations between them (Miles & Huberman, 1994). In the literature, four main sources are described to construct a framework i.e. (1) experimental knowledge, (2) existing theory and prior research, (3) pilot and exploratory research, and (4) thought experiments (Maxwell, 2005). In this research, the initial version of the proposed framework sought its basis from two sources i.e. extensive literature review (existing theory and prior research) and preliminary survey (pilot research results).

The literature study revealed that currently there are just a few empirical research studies that focus on the social aspects of agile requirements engineering (e.g. communication (Pikkarainen et al., 2008)(Cataldo & Ehrlich, 2011)). Hence, it can be concluded that unfortunately, not much literature on studying the socio-technical aspects in agile teams are available to date. Therefore, the proposed framework is an initiative to provide a formalized set of steps to study these socio-technical aspects (i.e. communication and awareness) among agile teams empirically supported by four exhaustive real world case studies. Moreover, studies from the literature remained silent on several issues such as team performance in variable development atmospheres such as agile methods with a high rate of change acceptability. Hence, this justifies the need in proposing a framework for agile teams with high collaboration, requirements volatility and small teams. Therefore, a formal set of steps was introduced to study the collaboration among agile teams in terms of their communication and knowledge of each other. Hence, this framework was proposed to study the socio-technical aspects of requirements-driven collaboration among agile teams (fulfilling research objective 2 and research question 2 explained in Chapter 1).

5.2. Purpose of the proposed framework

Agile teams are a closely knit, highly interactive set of 5 to 9 people working closely on volatile user stories in small iterations called sprints. The information is believed to be highly dispersed in the agile teams due to cross functional and self-organizing members (changing roles constantly) working closely. Unlike traditional software development teams, the chain of command and collaboration patterns of agile teams are different. Knowledge distribution in agile teams is among multiple roles and due to the close interaction with customers or proxy customer (i.e. product owners, project managers), it makes them less dependent on seeking information from outside (unlike traditional software development teams). These differences make agile teams collaboration patterns different from the traditional teams. This difference needs to be investigated by studying requirements-driven collaboration patterns among agile team members that communicate with each other all the time and hold most of the information within the team in order to investigate the (i) communication patterns among agile teams, and (ii) awareness patterns of agile teams.

The proposed framework helps to (i) study the socio-technical aspects of requirements-driven collaboration among requirements-centric agile teams and (ii) find their impact on performance. This framework provides an organized format to study the social aspects among agile teams and highlights the important aspects of the teams' collaboration to be studied. This framework integrates and furthers the findings and results of previous studies that focused on collaboration driven by requirements (e.g. Damian et al., 2010)). Also, it strengthens the conceptualization of the socio-technical aspects of collaboration driven by requirements engineering phases and provides assistance to researchers for designing their studies aiming at diverse results and strengthening concepts by interpreting those results. However, this framework differs from previous work (i.e. Damian et al., 2010)) on requirements driven collaboration in such a way that:

- It integrates and furthers the findings and results of studies focused on collaboration driven by requirements.
- It addresses a particular type of situation – agile teams with high flexibility towards welcoming changes in requirements at any stage.
- It investigates the effect of agile teams' collaboration in terms of the team member's communication and their awareness, on their performance.

5.3. Limitations of previous work

There are several limitations in the current conceptualizations of studying RDC based on the investigation of available literature. The limitations to which solutions are proposed in this framework are as under:

1. **The identification of most relevant socio-technical aspects of requirements-driven collaboration**

There are several socio-technical aspects studied among agile teams i.e. communication (Cataldo & Ehrlich, 2011)(Bjarnason et al., 2011a)(Pikkarainen, Haikara, Salo, Abrahamsson, & Still, 2008)(Licorish & Macdonell, 2013), trust (Al-ani, 2013)(Tjørnehøj, 2012), culture (Robinson & Sharp, 2005a) etc. However, there is still a need to identify the most relevant socio-technical aspects of requirements for agile teams in order to study them further in detail.

2. The requirements driven collaboration has been studied for traditional software development

The framework development by Damian and colleagues (Damian et al., 2010) studied the socio-technical aspects of collaboration for traditional software development teams. Thus, the iteration based small agile teams working on volatile requirements was an interesting dimension to be studied. Likewise, Cataldo & Ehrlich (2011) studied communication patterns for agile teams and calculated their impact on outcome quality and iteration performance. However, there is still a need to further these investigations and develop a systematic way of studying requirements-driven collaboration among agile teams which is proposed in this framework.

3. The impact of socio-technical aspects of requirements driven collaboration on team performance

Another challenge is to study the impact of socio-technical aspects of requirements-driven collaboration on the quality and performance issues. Among the previous work available in literature i.e. (Cataldo & Ehrlich, 2011) observed the impacts of communication patterns of small world agile team for IBM Jazz® project iteration performance by studying hierarchical and small world network structures of agile teams. However, there is still a room to further his findings with more empirical results. However, the study conducted to study the socio-technical aspects of collaboration driven by requirements among traditional software development teams by Damian and her colleagues' (Damian et al., 2010) also remained silent on performance issues. Thus there is need to focus on agile team's performance during iterations based on their collaboration patterns.

5.4. Construction of the framework to study RDC among agile teams

The proposed framework sought its basis from: (i) the systematic literature review (SLR1) (discussed in Chapter 2) conducted to deepen the understanding of agile requirements engineering and its underlying concepts, and (ii) the online survey (discussed in Chapter 4) conducted to define the most relevant socio-technical aspects of requirements-driven collaboration among agile teams with agile-based industry practitioners followed by a systematic literature review (SLR2). The main aims of the sources (i.e. SLR1, online survey and SLR2) that contributed to the construction of the proposed framework are shown in Figure 5.1.

The results of SLR1 helped to deepen the understanding of Agile RE and became the basis for the research motivation to develop a framework to study RDC among agile teams. Moreover, the results helped in defining the underlying concepts for the proposed framework.

The results of online survey identified the most relevant socio-technical aspects of RDC among agile teams i.e. communication and awareness. The results of the online survey provided the aspects to be studied among the agile teams.

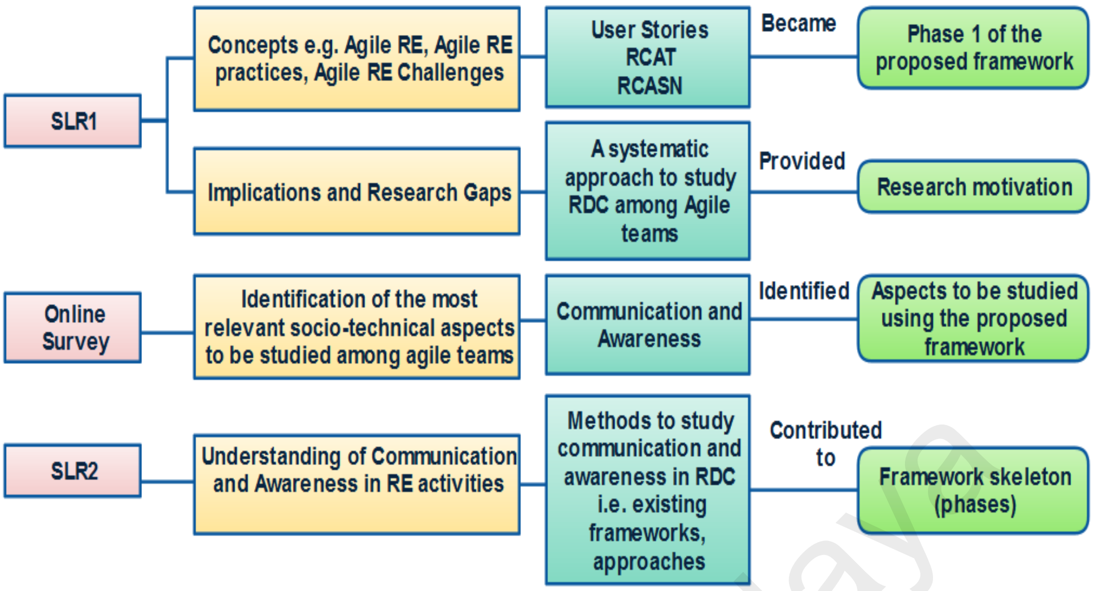


Figure 5.1: Aims of data sources contributing to the proposed framework

The results of SLR2 have, firstly, helped to gather more information on the two identified socio-technical aspects and secondly, defined the methods and approaches used in the literature to study them. This helped in forming the basic structure of the proposed framework based on the studies like (Damian et al., 2010).

Figure 5.2 shows the process of framework development. After the development of the initial version of the proposed framework from the above mentioned sources, a structural validation was applied to evaluate whether the constructs of the proposed framework to study communication and awareness among agile teams were appropriate to be used by the software teams for collaboration patterns and performance analysis. Expert opinion from well versed academicians and researchers were used for this purpose. According to the feedback, the framework was improved in terms of (i) the addition of several SNA measures into the revised version, (ii) the construction of a prototype to automate the functioning of the framework for ease of use, and (iii) industrial deployment was performed through empirical investigation (case studies). The revised version of the framework was evaluated after the empirical investigation in order to assess its applicability and utility through industry practitioners' feedback (explained in Chapter 7).

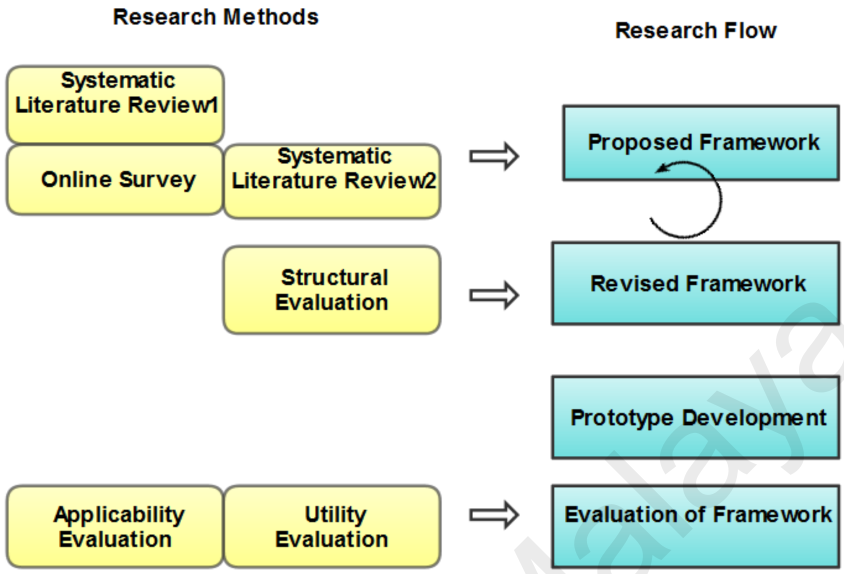


Figure 5.2: Framework development approach

5.5. Framework to study RDC among agile teams

Figure 5.3 shows the initial version of the proposed framework devised to answer the research questions (discussed in Chapter 1). Phase 1 of the framework is 'Conceptualization' in which the underlying concepts like user stories, requirement centric agile teams and requirements centric agile social networks are defined. Phase 1 provides the conceptual basis to the proposed framework.

Phase 2 of the framework is 'Visualization' in which communication and awareness networks of requirements-driven collaboration among agile teams is visualized in the form of networks called sociograms. The sociograms help to define certain characteristics of the communication and awareness networks i.e. members involved in networks, size of the networks, and density of the networks. In addition, it also helps to estimate the communication and awareness among the teams.

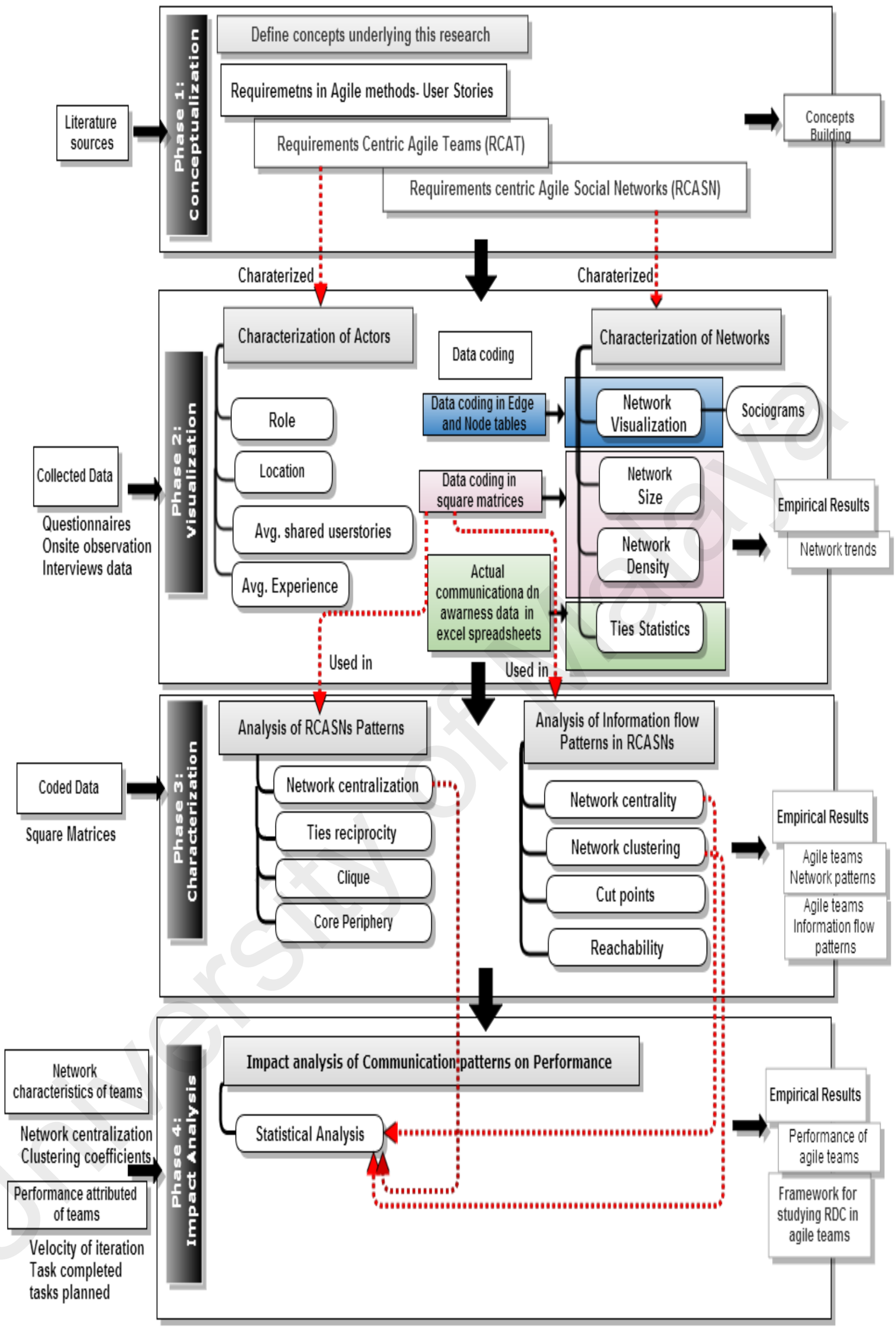


Figure 5.3: Initial version of the proposed framework

Phase 3 is 'Characterization' in which the structures of requirements-centric communication and awareness patterns among agile teams are studied to determine teams' centralization, reciprocity, clustering etc. In addition, characteristics of the information flow patterns of the agile teams are studied to find out the central and pivotal members of the teams.

Phase 4 is 'Impact analysis' in which the impact of communication and awareness networks is measured on agile teams' iteration performance. The detailed description of each phase is as under.

5.5.1. Phase 1: Conceptualization

The main underlying concepts in this study are defined in this section.

5.5.1.1. Defining User stories

A user story is a high level requirements artifact (Yin, 1994) and a simple description of the desired functionality from user's perspective (Cataldo & Ehrlich, 2011). User stories, as shown in Figure 5.4 are slimmer than normal usage requirements artifacts written by stakeholder (Yin, 1994). A well written user story follows the invest model (Wake, 2001) which states that a user story has to be independent, negotiable, valuable, estimable, small, and testable.

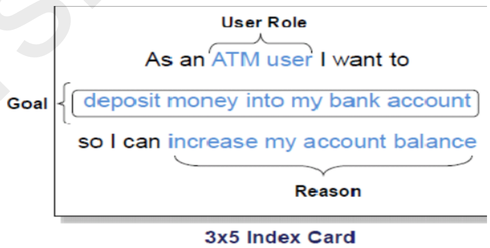


Figure 5.4: User story example (Cohn, 2004)

5.5.1.2. Defining Requirement Centric Teams in Agile Methods

A common definition of a team is ''a small number of people with complementary skills who are committed to a common purpose, set of performance goals, and approach for which they hold themselves mutually accountable"(Katzenbach & Smith, 1993).

Agile teams comprise of self-managing group of people, who coordinate, work together and make decisions of their own (Beedle, 2001).

First, we explained the concepts and definitions that we used in our framework. In our work, we have introduced the concept of a **Requirements-Centric Agile Team** (RCAT) (Inayat, Marczak, & Salim, 2013) based on the Requirements centric teams (RCT) concept from (Damian et al., 2010). RCAT is a group of cross-functional and self-organizing members working on a certain user story or a set of interdependent user stories, broken down into tasks as well as on downstream artifacts. This set of members includes anyone that is assigned to work on the project. It also includes anyone that 'emerges' during the life cycle of the project development, called 'emergent members' (Inayat et al., 2013).

5.5.1.3. Defining Requirements Centric Agile Social Network

Requirements-Centric Agile Social Network (RCASN) in this research is based on the concept retrieved from Damian et al., (2010). It represents cross-functional, self-organizing agile team members as actors and their interrelationships as ties. Such network is defined as a Requirements-Centric Agile Social Network (Inayat et al., 2013). For instance, a tie in a certain RCASN can represent a team member's communication regarding user stories or a team member's awareness of the other members etc.

The RCASN(s) created for RCAT(s) determines questions such as "who communicated for a certain user story?", "who was informed about any change in a certain user story?" and "who else needs to be contacted for a certain user story?"

(a) Communication RCASN

It consists of members technically dependent on each other i.e. contributing to certain interdependent user stories and their downstream artifacts and are assigned to work on certain interdependent user stories, communicating with each other. The reasons team members communicate with each other can be for user story negotiation, user story clarification, management and sprint planning etc. (Damian et al., 2010). Likewise, in this study, thirteen possible reasons for agile team communication were identified during data collection i.e. user story clarification, user story negotiation, changes communication, code synchronization, quality issues, management, support issues, sprint planning, and some code related issues, through onsite observations.

(b) Awareness RCASN

Awareness RCASN is the requirements-centric agile social network formed on the basis of the perception of awareness. Four kinds of awareness has been considered such as: (i) availability, how easy it is for one to reach a person when one needs help about the project (Ehrlich & Chang, 2006); (ii) general awareness, how aware one is of a person's professional background and how his/her skills could help one with his/her work on the project (Ehrlich & Chang, 2006); (iii) current awareness, how aware one is of the current set of tasks that a certain person is working on (Ehrlich & Chang, 2006); and (iv) work status awareness, how aware one is of a colleague's current work progress that is related to the project. The fourth type of awareness has been defined for this study given the relevance of constant progress report in agile teams.

To construct awareness-based networks, data was collected through questionnaires using questions like: "Are you aware of this project member's professional background" or "Are you aware of how these project members can help you in your work?", "Is it easy for you to reach this project member when you need help on certain user story?".

5.5.2. Phase 2: Visualization

This part of the proposed framework deals with the characterization of communication and awareness in agile teams. It is aimed to elaborate on (i) how do the communication and awareness networks built for the agile teams look like, (ii) how many people collaborated, (iii) what kind of collaboration patterns can be seen in the networks, (iv) how dense are the networks, and (v) who are the external or emergent members in the team. To answer all of these questions, the measures used are explained below:

5.5.2.1. Characterization of Actors

Characterization of actors means the identification of important demographic and project related details about them such as their experience level, geographical location, average shared user stories etc. The actor analysis helps in describing the fine details about networks in depth, dependencies between actors, and the frequency of their communication to determine their interest and level of awareness. Furthermore, a fine grain analysis of network actors helps in grouping actors according to their traits to better understand the information flow among them. For instance, the role of emergent members in networks, the role of experienced actors, etc.

5.5.2.2. Characterization of Networks

(a) Network Visualization

The networks can be visualized using a node and tie representation of an actor and his relationships, called sociograms. Sociogram is an old technique used to measure social relations using graphs, proposed by Jacob Levy in 1934. The data collected through questionnaires was transformed into matrices, called socio-matrices (Hollander, 1978) representing source and target nodes with direction (e.g. A communicated with B for communicating changes in user story) and weight (e.g. A communicated with B more than 4 times a day for user story clarification). These socio-matrices were then imported into a social network analysis visualization tool i.e. Gephi that results into sociograms. An example of user story clarification communication and availability awareness sociogram is shown in Figure 5.5. The weight is represented through variable colors of ties or edges (from red to black) and the color and size of nodes (from green to yellow) represent the degree of node.

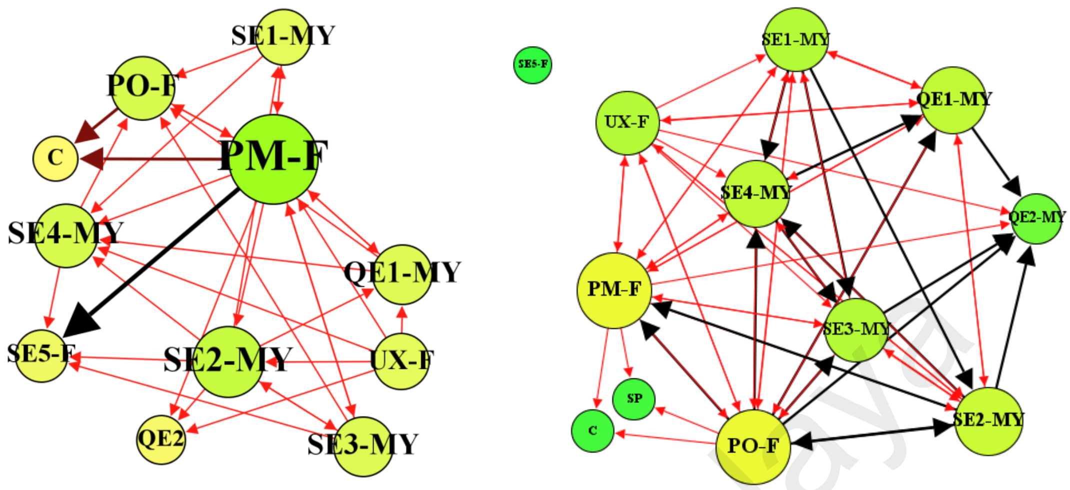


Figure 5.5: Examples of (a) communication and (b) awareness sociograms

(b) Network Size

Network size determines the number of actors involved in each RCASN. For instance, the RCASN size for the network presented in Figure 5.5. (a) is 11. It helps in determining: (i) the number of actors involved, (ii) the participation trends of users (e.g. most of the developers communicated for code issues discussions), (iii) the amount of collaboration required for a certain kind of user story.

(c) Network Density

Network density is defined as the proportion of existing ties with the total number of possible ties. Density helps to assess how tightly coupled the team is. If the density value is low, it shows that less communication happened between members than expected. The ideal network has a density value of 1. The networks with less participation have low densities (close to 0). The density of network can be interpreted as the speed with which information travels in a network or as the social constraint of the actors (nodes). Similarly, for RCAT, a high density of change communication in

RCASN is interpreted as a frequent communication of members with each other for communicating information about changes in user stories. On the other hand, a low density for management related communication means that fewer actors participated in management related communication. Density is a very important network structure measure which has been interpreted in several ways in the literature such as, a high density indicates ease of coordination among geographically dispersed software development teams (Hinds & Mcgrath, 2006) and volatility of requirements for traditional software development teams (Damian et al., 2010) etc. Therefore, along with network size, network density is also an important aspect to consider in studying the collaboration patterns of teams. Nevertheless, network size and network density are not correlated since network density depends on the number of ties between the nodes while network size is determined by the number of nodes or actors in a network. Therefore, if a network has a large number of nodes but less number of ties between them, the density will be low (Scott, 2012). If the number of ties an actor can have is limited, then the density will be low with the growth of the network. So, for a social network with limited node size, there is no correlation between network size and density.

(d) Network trends

The analysis of network structure, i.e. the number of ties linking the actors, helps in finding the collaboration patterns of teams. Each actor is examined with respect to its role in the team, location, and task assignment. The collaboration that happened within and cross –site can also be analyzed using network structure analysis. The analysis of network structures involves a two-step process. Firstly, a ties analysis named 'ties statistics' is performed by taking into account the number of ties between each pair of roles for all the communication and awareness networks. Using the pair wise coding format, data sets were imported to Excel to calculate the ties statistics for all of the communication and awareness networks. For each pair of members with existing link, we counted one tie. The communication and awareness networks were considered directional, i.e. if a pair 'source-target' reported communication and this same pair 'target-source' also reported communicating, two instances of communication are considered. This involved a visual inspection of the networks and a manual calculation process for filtering the dataset. Secondly, the data is recoded into a matrix format with a distance matrix (explaining the locations of actors per network) and a correlation was calculated between communication and awareness. This pair-wise coding format helps in determining the actors in pivotal position (with maximum number of ties), actors with maximum inter-role communication and awareness, and actors with maximum cross-site communication and awareness. In addition, the correlation helped in understanding the effect of distance on awareness and communication in distributed agile teams.

5.5.3. Phase 3: Characterization

To study the characteristics of the communication and awareness networks, a detailed analysis was conducted using SNA measures i.e. Network Centralization, Clique, and Ties reciprocity.

5.5.3.1. Characterization of Communication and awareness RSASNs patterns

(a) Network Centralization

The centralization value for networks determines the dispersion of ties around nodes. If a network is centralized, the value of the centralization lies closer to 1. This means that all nodes communicated with the central node the most. On the contrary, if the nodes have dispersed ties around multiple nodes, then the network is decentralized and the value of the centralization is lesser than 1. The centralization measure gives an idea on the chain of command in a team. If the centralization value for a network is high, this shows that the network was centralized and decentralized otherwise. In a centralized network, most of the information is centered and emitted from one node. However, in decentralized networks, information is distributed among variable actors in the team. Centralization is an important trait to study collaboration patterns among software developments teams. Previous studies used centralization for the analysis of software development teams traits, such as centralization is negatively correlated with the project size for open source software development teams (Crowston & Howison, 2006)(Amrit, 2005), and a low centralization indicates better performance (Sparrowe, Liden, Wayne, & Kraimer, 2001). Unlike traditional teams, RCATs are cross-functional and self-organizing. Therefore, it was highly interesting to study which collaboration among them was exhibited.

(b) Ties Reciprocity

The ties reciprocity measure helps to calculate the reciprocity of ties i.e. if A is connected to B; B is also connected to A. There are two variants in calculating reciprocity; i) **Dyad based** in which the number of reciprocated dyads is divided by the number of adjacent dyads, and ii) **Arc based** in which the number of reciprocated arcs is divided by the total number of arcs. Both dyad and arc based ties reciprocity were calculated for the communication and awareness networks in order to discover the actors involved in a reciprocal relationship (Dyad based) and the percentage of reciprocated ties (Arc based). This is used to characterize the balance in relationships (Hammer, 1985). Ties reciprocity has been studied in the literature in relation to several aspects such as: (i) the reciprocity in social networks gives rise to trust and strength of social ties (e.g. common friends measure in Facebook) (Bapna, Gupta, Rice, & Sundararajan, 2011); and (ii) reciprocity in ties mediated personal relationships and enhances performance etc. Therefore, it is highly interesting to study ties reciprocity among agile teams with members collaborating with each other all the time. Moreover, it is equally important to investigate mutual communication and awareness among actors in agile teams.

(c) Clique

A clique is defined as a subset of a network in which members (nodes) are more closely tied to one another than with the rest of the members of the network (Hanneman & Riddle, 2005). It is a subset of a graph in which all of the nodes are connected to each other (Mchugh, 1990). For instance, in real life social networks, friendship networks such as cliques are formed within members on the basis of common interests, age, gender and ethnicity etc. A clique test is helpful in identifying the least number of actors connected with each other (i.e. no null ties exist) among communication and awareness networks. A clique is used to study several collaboration phenomena in the literature such as, the number of cliques increases with the increase in friendship among a collaborative learning students group (Fong Boh, Slaughter, & Espinosa, 2007). Therefore, it is interesting to identify cliques in closely collaborating agile teams. These cliques help in understanding which roles were closely communicating for which communication reasons and which actors tend to communicate more with each other.

(d) Core Periphery

Core Periphery describes the network structure in terms of two sets of nodes i.e. core and periphery. Core has nodes that are closely connected to each other while periphery has nodes that are loosely connected to each other. The core-periphery measure is used in several variable aspects in the literature such as to measure the impact of leadership development in leadership networks (Hoppe & Reinelt, 2010). Therefore, it helps in identifying the core members and loosely connected periphery members among highly collaborative agile teams.

5.5.3.2. Characterization of Information flow among Agile teams

To identify the patterns of information and knowledge exchange among agile teams, several measures determining the information flow were used such as degree centrality, component, reachability, and cut points.

(a) Degree Centrality

Degree centrality is defined as the number of ties a node has. The number of ties a node emits determines its 'out-degree' while the number of ties it receives determines its 'in-degree'. Degree centrality along with Eigenvector centrality are considered highly used and appropriate measures for affiliation network (Faust, 1997). The centrality measure is used in the literature to determine: (i) the power of actors in a social network (Gomez et al., 2003), and (ii) the activity of actors (Freeman & Mulholland, 1979). For an agile team, the centrality measure helps in determining the actors with the highest degree values and helps in identifying the most active, powerful, pivotal and central actors. The information is distributed in agile teams due to role sharing and cross-functional responsibility sharing; therefore, it is an interesting aspect to study in agile teams. The more central actors tend to emit more information and vice versa, which determines the information flow among agile teams.

(b) Component

The component measure is defined as the maximal set of nodes connected to each other through some path (of any length). Component measure is used in the literature to determine: (i) outliers detection from insurance related social networks, and (ii) retrieving relevant users in a social network (Canali, Casolari, & Lancellotti, 2010).

There are two kinds of components: (i) strong components; node A and B need to have a directed path, and (ii) weak components; Node A and B are connected to each other regardless of the directed path. Therefore, component offers a fine grain analysis of an actor's tendency to collaborate with his colleagues in agile teams. It defines the actors that are in connection with each other directly and those who are communicating but not directly.

(c) Reachability

To further the information gained from components on connectivity of nodes, reachability is used. It identifies any existing path between two nodes no matter how many other nodes fall in between them. Reachability is used in the literature for computing the complexity of an algorithm for providing suggestions to users in a social network website. Therefore, it determines that if some nodes are not reachable in a network, it has the propensity to fall apart into small subgroups. To investigate the tendency of RCASNs to fall apart in subgroups, reachability serves this purpose. This helps in analysing which actors tend to stay close to each other.

(d) Cut point

Cut point determines the actor or actors (called cut sets) in the networks who, if removed, can cause information loss due to lost connections with other nodes. So, it is important to identify such important nodes that, if removed, can cause information lapse in the networks. After investigating the tendency of a network to split (reachability) and into which components (strong and weak components), it is equally important to identify the connecting nodes between subgroups. These nodes are pivotal in information transfer between the subgroups they link. Cut points are used to detect the key players in a social network (Borgatti, 2006) and information brokers in traditional software development teams (Damian et al., 2010).

(e) Clustering Coefficients

Clustering coefficients is one of the most important properties of a graph or network which determines the degree or extent of nodes to stay closer or connectivity of neighbouring nodes. In other words, in the context of friendship clustering, coefficient determines the measure of the extent to which ones nodes' friends are each other's friends as well (Watts & Strogatz, 1998). After studying the split in networks and nodes or groups that have the tendency to split (reachability and component) and nodes that are the foci of attention between two split sub groups (cut points), it is appropriate to find out the tendency of nodes to stay close to each other. In addition, clustering coefficient helps in understanding small world effect (two nodes connected to each other through the smallest path) and clustering among agile teams (Watts & Strogatz, 1998).

5.5.4. Phase 4: Impact analysis

An agile team's performance closely depend on close-loop team communication (Dingsøyr & Lindsjørn, 2013) ,mutual communication patterns (Cataldo & Ehrlich, 2011) and teams' coordination (Burke et al., 2006). The performance of agile teams has been described, with respect to several factors in the literature, as the number of defects and rework per iteration (Cataldo & Ehrlich, 2011), the time to complete a certain task (Espinosa, Slaughter, Kraut, & Herbsleb, 2007), the time to complete the rework on requirements changes (Cataldo & Herbsleb, 2008), and the quality, dividing work, iteration amendments and teams' satisfaction (Drury-Grogan, 2014). However, identifying an agile team's productivity factors in a socio-technical system are considered to be under-researched and challenging (Melo, Cruzes, Kon, & Conradi, 2013). Therefore, this study aims to find the impact of the agile teams' collaboration i.e. their mutual communication and awareness of each other's general traits, availability, current task and work status on iteration performance by keeping in view the team's velocity and the number of unfinished tasks which relates to the 'quality' and 'schedule' category defined by Drury-Grogan, (2014). In practice, there are several metrics used by industry practitioners to measure an agile team's performance such as LOC (line of code), customer satisfaction, defect count, lead time, unfinished stories etc.

The impact of collaboration driven by requirements on the performance of agile teams was sought in terms of the number of user stories completed per iteration, the number of user stories repeated per iteration (amount of rework), and the velocity per iteration. Moreover, we have explored the following issues: (i) what impact awareness has on distributed agile teams' communication, (ii) what is the impact of distance on awareness and communication of distributed agile teams, (iii) what impact small worldliness has on iteration performance, and (iv) how network centralization is linked to team performance.

5.5.4.1. The interplay between distance, communication, and awareness

The relationship between distance, communication and awareness is analyzed in order to determine how much these factors are interdependent on each other. The main aim was to identify whether the result of this empirical research corroborates with the previous findings or deviates from them. The quadratic assignment procedure (QAP) (Hubert & Schultz, 1976) was employed to find out the mutual correlations between communication, awareness and distance. QAP is an unbiased technique to analyze dyadic social network based data to answer questions on observed relationships (Krackhardt, 1987). Multiple correlation operations were performed on coded awareness, communication, and distance matrices to assess the team's behaviors.

5.5.4.2. Iteration performance

Iteration performance is calculated by keeping in view the number of tasks allocated, completed and not completed at the end of each iteration. A task is defined as a unit of work for each iteration. Each task is later broken down into further downstream artifacts such as specification writing, design work, coding, testing, and debugging. The agile teams allocate points to the tasks and use these arbitrary values to measure the effort required to complete a task or user story called story points. These points can be allocated in many ways based on the team's preferences. In most cases, project managers define a story point complexity range as a Fibonacci series (for example, 1, 2, 3, 5, 8). The story points produced per iteration is called the team's velocity. To determine the team's performance, several details were kept in consideration such as team size, maximum tasks allocated per person and iteration size. To measure the iteration performance, several attributes of networks were considered i.e. small worldliness and centralization. Small world network structures are defined as small and highly dense clusters of nodes connected to a few other nodes. In the context of collaboration translated to communication between software development teams, it means that a team member is surrounded by dense networks of other team members all communicating with each other. Small world network structures for communication between agile teams is considered detrimental to the iteration performance in the context of iteration planning and execution by reducing the time spent on real work (Cataldo & Ehrlich, 2011). However, small world network structures are found well suited for interdependent development tasks which, in return, enhances the iteration quality (Cataldo & Ehrlich, 2011). Small world network structures have (i) high clustering coefficients, and (ii) low average path lengths. Likewise, network centralization was used to determine the effects on agile team performance. For this purpose, in-degree centralization and out-degree centralization of all of the networks was considered.

5.6. Structural Validation

The objective of the structural validation process is to gather feedback on the constructs of the proposed framework and to improve according to the suggestions. Expert opinion of well-versed academicians and researchers were used for this purpose. The GQM statement to perform this validation is shown in Table 5.1.

Table 5.1: GQM statements to perform validation with researchers

To Analyze	The processes to study the socio-technical aspects of RDC in agile teams
In order to Evaluate	The constructs of the proposed framework
From the perspective of	Researchers in the field of requirements engineering and agile software development methods
In the context of	Face-to-face and online questionnaire based interview
Because	The suggestions of researchers and academicians help to improve and validate the results

A semi structured interview strategy was chosen to gain information from the experts. The interview technique was intentionally chosen over others, such as questionnaire based survey, so as to gain more of the expert's opinion which would not have been possible in the case of a Likert scale questionnaire. Moreover, interactive sessions can lead to a more constructive feedback which is impossible to gain through an open ended questionnaire. However, this technique was time consuming right from its preparation to sending invites and waiting for responses and then having interviews conducted at the agreed time for each of the expert. Nevertheless, it provided some useful suggestions which led to slight modifications in the proposed framework making it more applicable for the industry.

5.6.1. Participants

The participants of this questionnaire based interview were selected on the basis of their professional background, research interest and their publications after pursuing their due willingness to participate. The invitations, along with the details of the validation study, were sent to 15 experts within the field. However, only six experts were willing to participate. The research interest and country wise distribution of the participants are shown in Table 5.2. The rest of the details of the experts are kept hidden in order to maintain the privacy of personal data.

Table 5.2: Area of expertise of the experts

No.	**Experts' Research interests**	**COUNTRY**
1.	Agile software development methods	Malaysia
2.	Requirements engineering	Saudi Arabia
3.	Social aspects in software engineering and Empirical studies in Requirements Engineering	Sweden
4.	Agile software development methods especially Scrum	Pakistan
5.	Requirements Engineering, Requirements prioritization in agile methods	Malaysia
6.	Agile software development methods	Malaysia

5.6.2. Research Instrument and Procedure

A questionnaire was prepared (shown in Appendix B, Table B.1) and used to conduct the interviews. The questions were not asked in the same order as they were written in the questionnaire. A copy of the questionnaire was also sent to the participants to avoid any kind of miscommunication. The interviews were conducted online through Skype chat service while face-to-face interviews were conducted with the participants from Malaysia. A handout comprising the necessary information to be shared was prepared and sent beforehand to the participants. In addition, the interviews were followed by a short 10-15 minutes oral presentation to avoid any ambiguity and for a clear explanation of the results to gain a concise and appropriate feedback. The interviews were semi structured and the sequence of questions was changed according to the direction of the discussion. The interviews were recorded and field notes were taken so as to avoid any missing information. Later, while transcribing the interviews, the field notes were matched and linked to the discussion. Post interview emails were also sent to the experts to clarify on certain aspects.

5.6.3. Analysis and results

First, the interview recordings were transcribed carefully by replaying them at least twice to avoid any missing information. Each of the transcribed file is saved separately with the interviewee's identical code allotted for privacy purpose i.e. A, B, C etc. So, there were a total of 6 transcribed files from A to F. Then, the coding process was started. During the transcription process, a few minor ideas on the categories of comments emerged. From here, the main categories were later on formulated such as feasibility, measures, and steps. These main categories were looked into in the transcribed files for another three sub categories which are suggestion, disagreement, and agreement. The descriptions of these sub categories are such that (i) suggestion is any action required for the improvement or change in the proposed framework, (ii) disagreement is any concern of the experts on the current proposed framework, and (iii) agreement is the acceptance by the experts of the proposed framework steps and measures with no further action required in terms of any change or deletion. The detailed coding of the experts interviews is shown in Table 5.3.

Based on the suggestions, agreement and disagreements shown in Table 5.4, the necessary actions were taken before validating the suitability and applicability of the study by industry practitioners. The list of the necessary actions taken on the experts advices is shown in Table 5.4.

Table 5.3: Categories and their descriptions from the transcribed interviews

Category	Sub category	Description	Interviewee
Feasibility	Suggestion	Endorsement from industry practitioners will add to it	B
	Agreement	A feasible approach for studying collaboration	A
		An approach that provides insights on the teams' collaboration	E
		It is feasible to study collaboration of agile teams	F
		It can be followed by studying other social aspects in the future by researchers	D
	Disagreement	Difficult to evaluate without industry endorsement	B
SNA and statistical Measures	Suggestions	More SNA measures can be added based on the intention of the manager and situation of the team	D
		Some SNA measure that determines the position of a particular node in the network can add to the analysis	C
	Agreement	Adequate SNA measures for network analysis	B
		Enough SNA and statistics for finding detailed insights on a team's collaboration attributes	A
		Covers almost every aspect of team's collaboration	D
	Disagreement	Applying all of them would be too much for a Manager to do	C
Improvement	Suggestions	Deploy in industry and gain feedback on the results	C
		Ask the industry practitioners if they are interested to use these insights	D
		Deployment in industry should be made easier	E

Table 5.4: Concerns and necessary actions taken to mitigate them

Concerns	Action taken
Endorsement from industry practitioners will add to it	Validation from industry practitioners was already planned and conducted after this phase. The detailed description of industry validation is shown in Chapter 7 in Section 7.3 and 7.4
Difficult to evaluate without industry endorsement	Industry evaluation is described in Chapter 7 in Section 7.3 and 7.4
More SNA measures can be added based on the intention of the manager and situation of the team	The SNA measures were added after industry validation based on the practitioners explanation of the situations of their interest.
Some SNA measure that determines the position of a particular node in the network can add to the analysis	Eigenvector centrality and Betweenness were added to the SNA measures set in order to measure the position of a node in a network.
Deploy in industry and gain feedback on the results	Industrial validation is performed and described in Chapter 7 in Section 7.3 and 7.4
Ask the industry practitioners if they are	Industrial validation is performed and described

interested to use these insights	in Chapter 7 in Section 7.3 and 7.4
Deployment in industry should be made easier	A prototype was developed to enhance the data collection and data translation process in applying the SNA measures explained in Section 5.8.

The researchers and academics focused on the industrial evaluation of the framework. The industrial evaluation was conducted after this step. Based on the experts' suggestions, a prototype was developed to enhance the data collection and data translation process for the managers to run the SNA measures. The proposed set of steps were automated through this prototype and thus, made it more useable for the researchers and industry practitioners. In a nutshell, from the structural validation performed, confidence was gained on the proposed steps for studying collaboration and finding its impacts on performance.

5.7. Revised Framework

The framework was revised based on the suggestions of the experts' panel. The revised framework, shown in Figure 5.6 consists of the incorporation of the prototype built to automate the data entry and conversion part of the framework. The manual process of data coding was replaced by the prototype making it less complex and handy. In addition, the data entry process is also automated which doesn't need any manual recording of the data.

In addition, two measures to assess the individual properties of members including Betweenness and Eigenvector centrality were added to the framework. The revised framework is presented in Figure 5.6.

A summarized view of the proposed framework comprising the SNA measures, the research questions each measure answers and the results generated is presented in Table 5.5. The grey shaded rows show the measures that were added to the framework after getting feedback from the expert panel through the structural validation of the constructs of the framework.

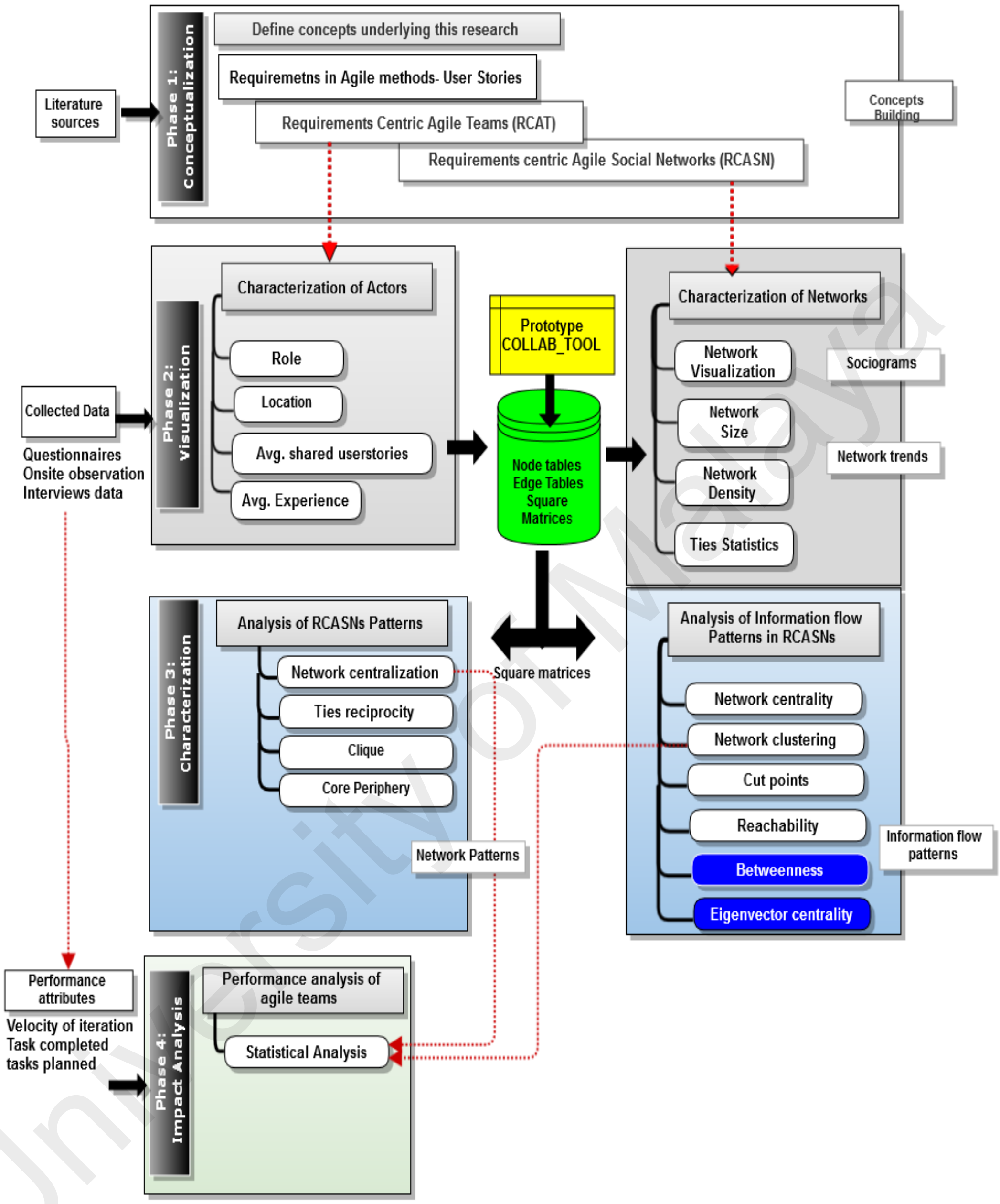


Figure 5.6: Revised version of the proposed framework

Table 5.5: Summarized view of the framework

SNA Measure	Question	Result
Network Visualization		
Sociograms	How do communication and awareness networks of distributed agile teams look like?	Overall view of networks

Size	How many team members collaborated?	Number of team members who participated in each network
Density	How dense are the communication and awareness networks for distributed agile teams?	Network cohesion Ratio of actual ties among members to the expected number of ties.
Ties Statistics	What kind of communication and awareness ties existed among distributed agile team members?	Nature of ties i.e. assigned or emergent Ties among roles Ties among sites
Characterization of networks		
Centralization	How dispersed are the communication and awareness around a node?	Collaboration centralization and knowledge distribution
Core periphery	Do Agile teams being closely knit, have core members?	Members with large social capital
Cliques	Do agile teams have some members working closely to each other?	Member or group of members who worked closely
Ties reciprocity	How mutual and balanced the collaboration is in distributed agile teams?	Reciprocal communication and awareness ties among members
Information flow in networks		
Degree centrality	How much collaboration actually happened among distributed agile team members?	Ratio of emitted and received communication links for each member
Betweenness	How much each node collaborated?	Central members of the network
Eigen vector centrality	Which members influenced the networks?	Influential member of the network
Reachability	Is there any path for information flow among distributed agile team members?	All members can reach each other
Clustering coefficients	What is the connectivity ratio of members in a distributed agile team?	The extent to which members stay close to each other and formed clusters
Cut points	Do some members of a distributed agile team play a central role in information exchange?	The important members who if they go missing can cause a communication gap
Components	Are there any isolated or lone members or groups of members present in distributed agile teams?	The inactive lone and isolated members
Impact Analysis		
Distance vs communication and awareness	Impact of distance among members on their mutual communication and awareness	
Communication vs awareness	Impact of awareness on communication on distributed agile team members	
Small world network structure and iteration performance	Impact of high clustering coefficient index and small average path lengths on network structure on distributed agile teams performance	
Network in degree and out degree centralization and iteration performance	Impact of network in degree and out degree centralization on iteration performance	

5.8. The prototype- Collab_Tool

The objective of the prototype is to provide ease to the industry practitioners in entering their collaboration data and the conversion of these raw collaboration data into a social network analysis tool's acceptable format. Due to the complex and lengthy data collection process employed by the framework, having a prototype would help in getting the data entered and converted into the acceptable format in a systematic way without carrying out the process manually. The literature supports the fact that tools automate proposed methods, simplify them and in some instances, shorten the process (Avison, Golder, & Shah, 1992).

The tool is a Windows based standalone application that partially automates the working of the framework and makes it more practical for industry practitioners and ultimately, for research purposes. The Collab_tool was developed using Labview (Laboratory Virtual Instrument Engineering Workbench), a system-design platform and development environment for visual programming language from National Instruments. Labview was chosen because it combines graphical user interface development within the development cycles which makes it dynamic and interactive. The design diagram of the prototype is shown in Figure 5.7.

The present version of Collab_tool was developed to facilitate data entry and data conversion into an acceptable format for SNA. The prototype accepts raw collaboration data entered by the user in the form of responses to the questions posed to them e.g. have you communicated with Mr. X or are you aware of Mr. X work status etc. The prototype performs two major tasks: (i) converts raw collaboration data into node and edge tables, and (ii) converts the data into square matrices for the application of SNA measures using UCINET tool. Both of the processes are explained as under.

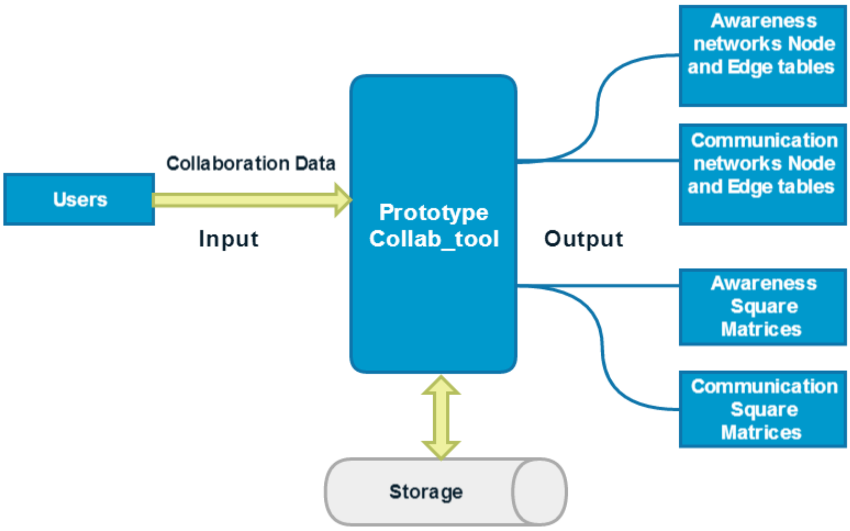


Figure 5.7: The design diagram of the framework prototype

5.8.1. Formation of sociograms

The Collab_tool, a prototype developed to automate a part of the proposed framework, enables the user to generate node and edge tables for the creation of sociograms. The node and edge tables are used to mask the networks by making use of social network visualization tools; in this case, Gephi was used. The node and edge tables facilitate the conversation of raw collaboration data gathered through the deployment of an automated questionnaire by the user into social networks or sociograms.

To achieve this purpose, the prototype was embedded with the questionnaire used for this study. The respondents were asked to fill-up their demographic details under the tab "Add Personal Data" as shown in Figure 5.8 and information regarding the emergent members that they have come across during the iterations under the tab "Add Emergent" as shown in Figure 5.9.

The users can enter the communication and awareness details by mentioning their role, the respective members' roles, and checking the subsequent communication and awareness questions at the front as shown in Figure 5.10. The user needs to choose one team member from the drop down list available under the tab "Team member's Role" and then check the relevant boxes on the right side displaying the questions regarding communication and awareness of the team members. After checking all the suitable boxes for the selected team member, the user can click "submit" to continue on with the next team member. In the end, the user will click "submit" and the results will automatically get stored at the already specified target location as node and edge tables for each of the communication reason and awareness type.

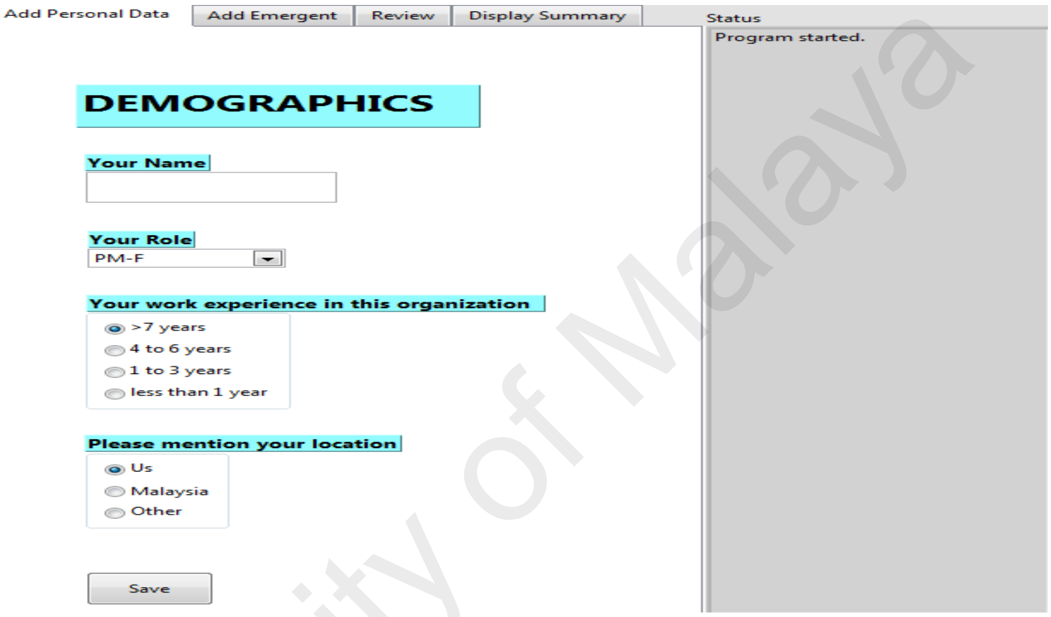


Figure 5.8: Demographic information of the respondent

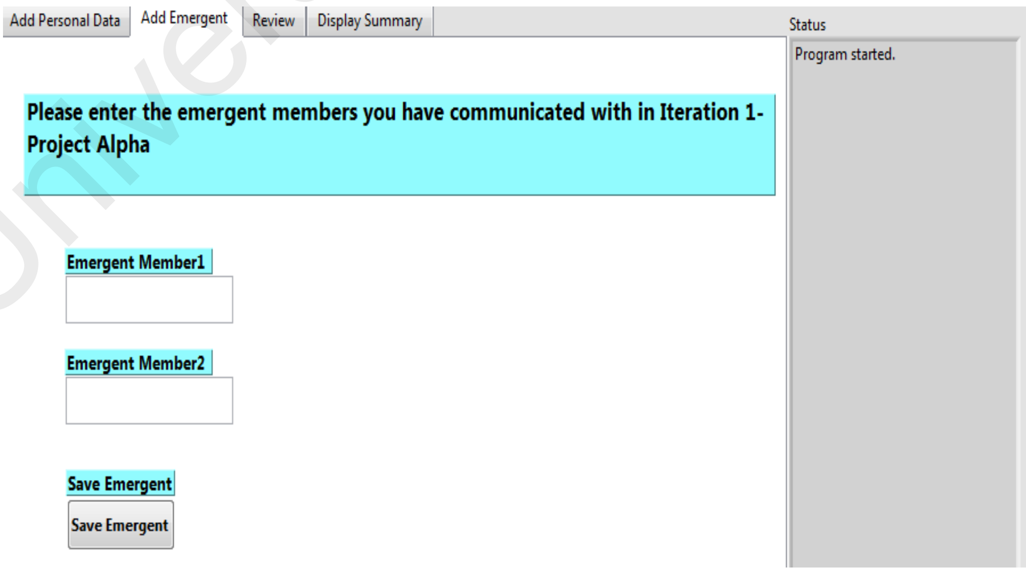


Figure 5.9: Emergent members' information

The data entry status is continuously updated in the right most panel of the screen as shown in Figure 5.10. The status panel helps the user to review his entered details and to check for missed ones. Moreover, the status prompts a message if the user re-enters details by mistake for a particular team member.

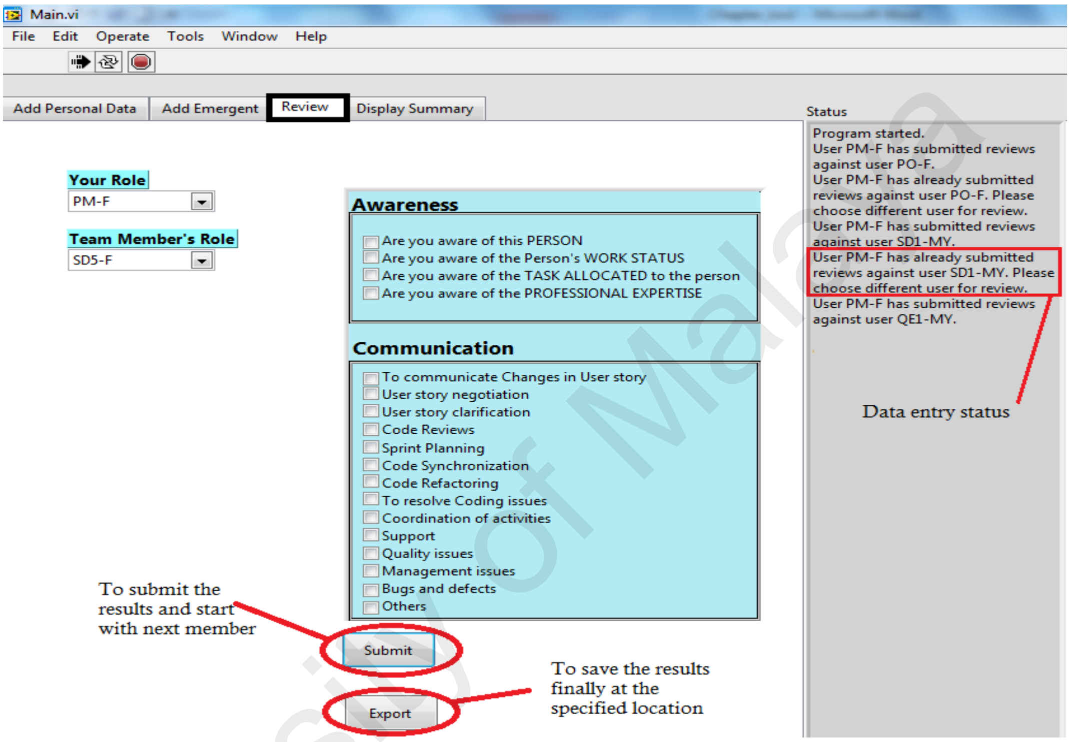


Figure 5.10: Communication and awareness data entry

When the tab "submit" is clicked, the data of all the respondents entered through this form is saved at the target locations in the form of edge and node tables. An example of the node table and demographic details record of the team members is shown in Figure 5.11.

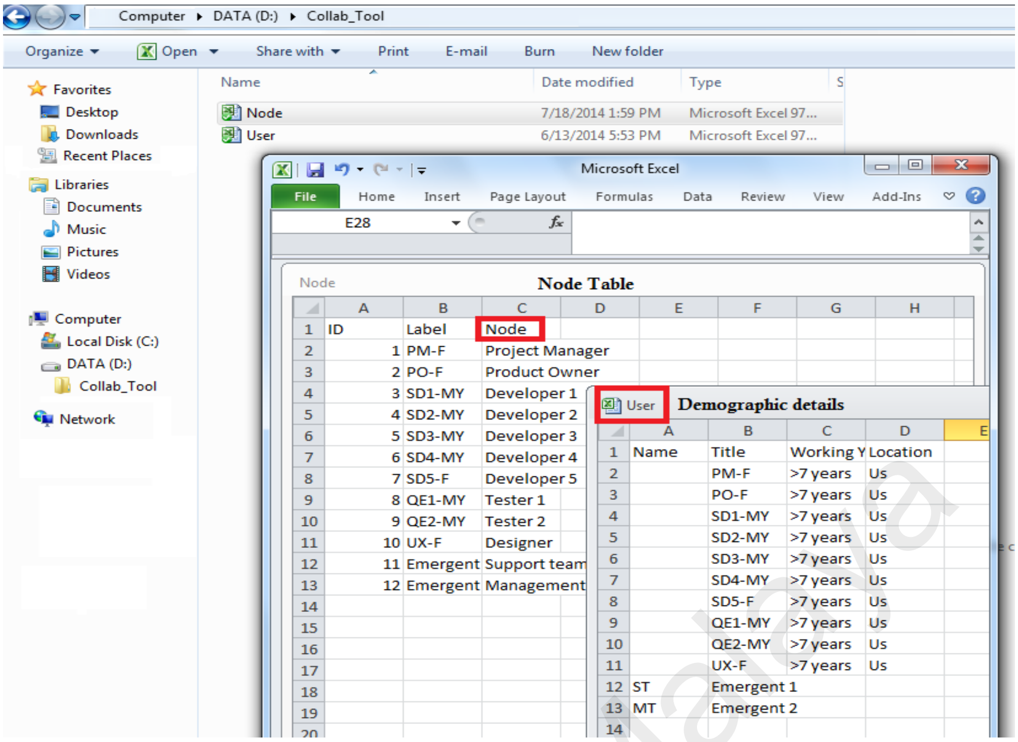


Figure 5.11: Screen shot of the node table and demographic file

The edge tables for all of the communication and awareness networks are shown in Figure 5.12.

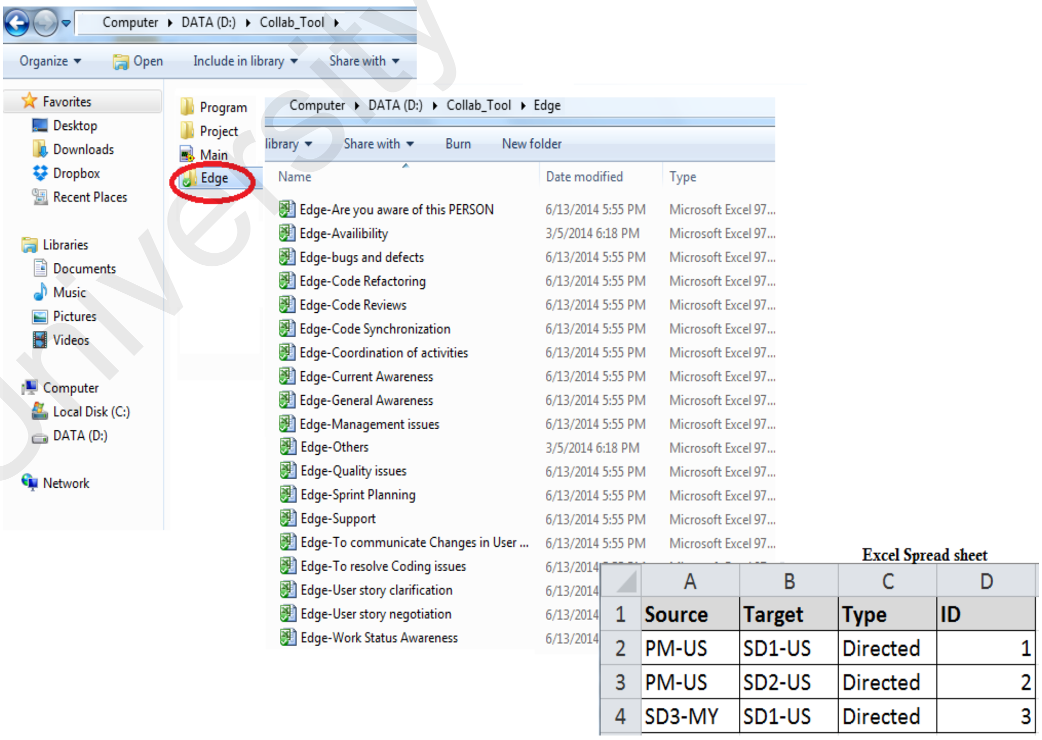


Figure 5.12: Screen shot of edge tables created for communication and awareness

The figure shows that in the tool's system folder "Collab_Tool", another folder named "Edge" has been created containing the excel sheets (edge tables) for all of the communication and awareness networks. The edge files are named as "Edge- User story clarification" and they consist of source, target, type of relationship and id. Node tables are created the same way at the same location along the Edge folder. These excel spreadsheets, i.e. node and edge tables, can later be directly transported to the social network visualization tool to visualize the sociograms.

5.8.2. Conversion of collaboration data in UCInet acceptable format

The tool aims to convert the raw collaboration data entered by the team members through the form (shown in Figure 5.10) into a UCInet readable format, square matrices as shown in Figure 5.13. The UCInet accepts collaboration data in a square matrix format with actors on the x and y axis and Boolean values to represent the relationship among them (e.g. 1 if there is a relationship and 0 if not). The tool inputs the data entered by the team members and saves them into edge and node tables for sociogram formation and simultaneously, creates square matrix files at an already specified location in a folder named "Report" as shown in Figure 5.13. The tool creates separate files comprising square matrices for each of the awareness type and communication reasons. These square matrices can easily be transported to the UCInet tool to assess the networks by applying various social network analysis measures such as centrality, reachability etc.

5.8.3. Report Generation

The tool provides a feature for viewing the data entry report for all of the communication reasons and awareness types in the tab "Display Summary", as shown in Figure 5.14. This feature provides a summary of all of the communication reasons and awareness types. The user can click on any of the communication reason or awareness type to view the corresponding square matrix or the summary of the collaboration data entered.

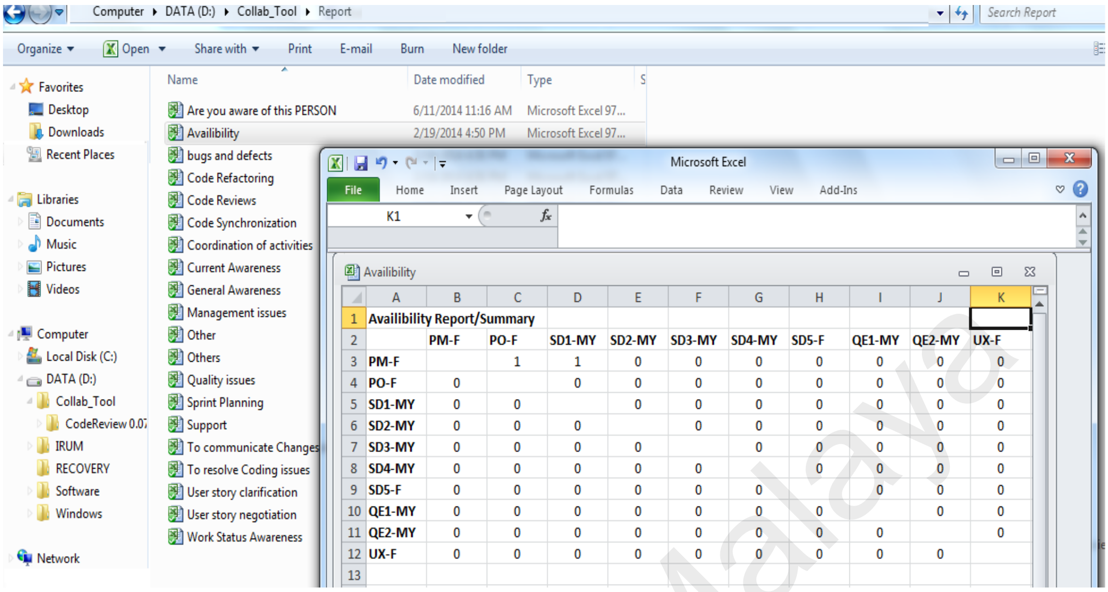


Figure 5.13: Screen shot of the square matrices created for communication and awareness networks

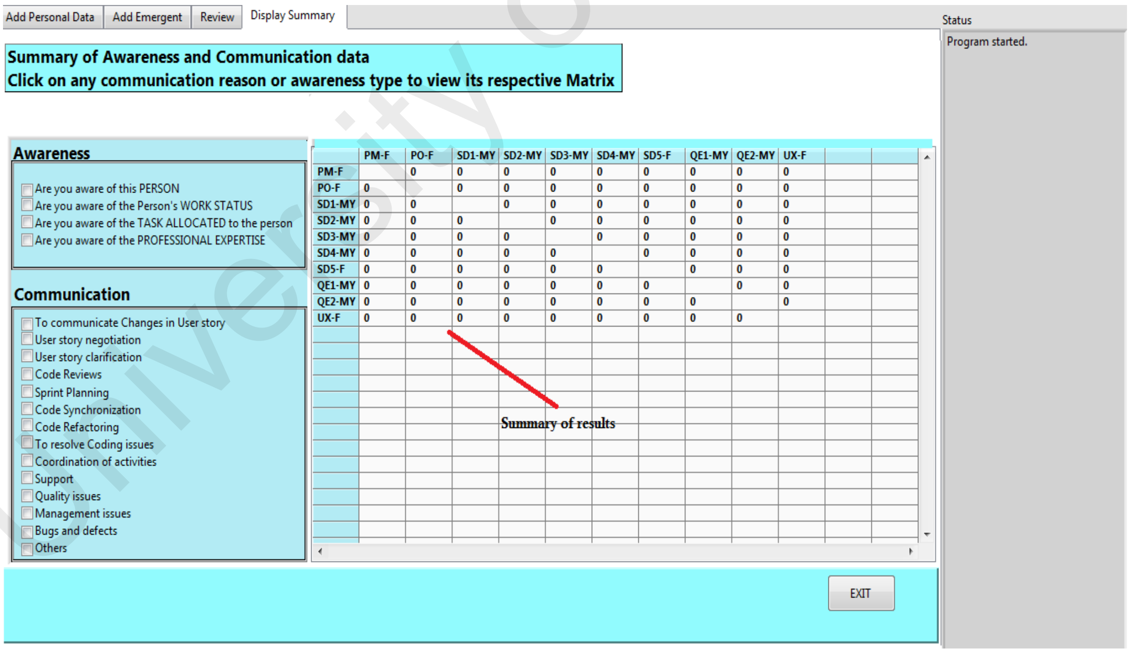


Figure 5.14: Screen shot of communication and awareness data summary

5.8.4. Implementation

The Collab_Tool prototype was developed using Lab view platform. Labview platform is a system design environment that offers productivity by providing abstraction and integrates all the technologies in a unified development environment. Labview provides an opportunity for visual instrumentation and enables users to design in a graphical environment which is highly suitable for developing prototypes in research and development. The graphical design environment is more intuitive than text based coding and offers a variety of functions, tools, and visual palates known as Virtual Instruments (VIs) to maneuver with. The VI hierarchy of the Collab_tool prototype is shown in Appendix B, Figure B.1.

5.9. Summary

The initial version of the proposed framework is based on the data sources i.e. two systematic literature reviews (discussed in Chapter 2 and Chapter 4) and an online survey (presented in Chapter 4). Structural validation was used to gather feedback on the initial version of the framework. The expert panel's suggestions were implemented to develop a revised version of the framework and prototype was developed to partially automate the framework. The prototype helps to gather data from the teams automatically converts and saves it in the required formats to be used for further analysis. The prototype makes this framework more practical and usable for the practitioners.

CHAPTER 6: EMPIRICAL INVESTIGATION

In this chapter, the empirical investigation performed through the multiple case study method to implement the framework for studying the socio-technical aspects, i.e. communication and awareness, of requirements-driven collaboration among agile teams, is discussed. To keep the content manageable, only Case 1 is described in detail based on the step by step execution of the proposed framework. However, the details of the rest of the case studies (i.e. Case 2, 3 and 4) are shown in Appendix C. The results of all the cases are discussed in this chapter in order to answer the proposed research questions and fulfill the research objectives.

6.1. Case 1 -Alpha

6.1.1. Company introduction

Alpha is a product development organization that follows in-house product development of internet security applications. The main buyers of the products are internet service providers (ISPs). Product ideas are gathered from potential customers and users through surveys and workshops. From the feedback gathered, new and improved product ideas are generated on which the company develops their new products. A product is launched to the potential buyers after completion and then customized according to the buyers' company environment specific requirements.

6.1.1.1. Company workflow

The company focuses on in-house product development and follows agile methods i.e. Scrum and Kanban. The project that we have studied followed the scrum method. The workflow that is followed companywide is as below:

- Envisioning Phase: The Product Managers (PM) proposes ideas to the leadership team. Product ideas are gathered through workshop sessions and customer surveys with potential users of the product. The ideas, when presented to the leadership team, are approved and then the team prepares the initial vision of the project by allocating resources, deciding the time to complete and selecting the teams for the project.

- Initialization Phase: Then, teams are put together consisting of 5-9 people. The number of iterations is decided, with each one being 2 weeks long.
- Construction Phase: In each of the iteration, day 1 is planning/documentation while day 10 is demo. It is not necessary to have a deliverable at every demo, e.g. in the 1st iteration, the teams normally come up with the plan and documentation.

6.1.1.2. Project Introduction

Project Alpha is a security related project which ensures the security of user information while being online on a computer or smartphone against any digital or mobile threats. To meet the imminent challenges of cyber security, the company followed a constant enhancement process which offers customizations and enhancements to every released product. The project being investigated is one of the customized or enhanced releases of the Alpha project. The project has 6 iterations, each with a total duration of 3 months, called one business iteration. After every 2 weeks (one sprint), at the end of one scrum iteration, a "preview release" is launched to around 500 users for feedback. When the business iteration completes, the beta version is released to more than 1000 users for feedback. Then, the next three months business iteration is planned for customizations.

6.1.1.3. Project requirements

At company Alpha, the product manager comes up with a set of product features which is later broken down into user stories and tasks. Project Alpha has 6 features, each feature is divided into 2-3 user stories with each story divided into 5-6 tasks depending on their complexity level. This makes a total of 17 user stories and 85 tasks.

6.1.1.4. Project Team structure

There was a total of 10 team members distributed at two locations, Malaysia (MY) and Finland (F), including: 1 Project Manager (PM-F), 1 Product Owner (PO-F), 5 Software Developers (4 Dev-MY, 1 Dev-F), 2 Quality Engineers (Tester-MY), and 1 User Experience Designer (UX-F). The UX was not a dedicated member of the team since he was attached to three projects (at most) simultaneously. The emergent members recorded during data collection for the team interaction include customers, support team and sales team.

At the Malaysian site, the team used to work from home once every week. That day was decided to be on Tuesdays, when the team would work from home and conducted their daily scrum through Skype video call and chat.

6.1.1.5. Artefacts used

Product burn down charts, story boards, product backlogs and user stories were used in the project. Altersian JIRA was used as a tool for iteration management including bug tracking, issue tracking, user story management and overall project management. Documents were maintained in terms of the PM's initial version of features and as time goes on, changes are made to the document. A storyboard is a readily available visual record of the teams' activities, performance and project progress.

6.2. Data Collection

In this section, the two phases of data collection are described in detail. In addition, the data collection methods employed are also discussed.

6.2.1. Phase 1

Phase 1 of data collection comprised of three steps. The first step dealt with information regarding the projects, while the second step was about the teams and the third was about the processes, practices and tools used.

6.2.1.1.Step 1: Data gathering on Projects

To start off, the elementary information on the projects was gathered. This involved the nature of a project i.e. whether it is an innovative project or an old project with maintenance. The rest of the important factors regarding the project are given below:

- Project nature i.e. innovative, old, maintenance , Project terms, history, scope
- Life cycle span of the project, Iteration size i.e. duration of iteration
- Agile method in use Life i.e. Scrum, Lean, etc.
- Number of iterations i.e. the total life span of the project divided into how many iterations, Number of work items (broken down user stories) per iteration, Number of tasks completed per iteration, Number of tasks left over and repeated in the next iteration

6.2.1.2.Step 2: Data gathering on Project teams

A team is a set of people working together on certain interdependent work items which includes developers, testers, analysts, designers, customer representative and any other person allocated any custom role in the team. A project team's related data involves the team's distribution (if any) and the team's collaboration practices. Furthermore, it involves gathering information on the emergent people who were not formally part of the team but were contacted in several cases. The details are as below:

- Team size, team structure i.e. distributed or collocated.

- To learn about the people involved and their roles in action, Responsibilities of roles i.e. team members

6.2.1.3.Step 3 Process and Practice data

Step 3 was aimed at collecting detailed information on the development process. Some organizations follow basic agile principles while others follow tailor made customized approaches. Therefore, it was necessary to focus on the process and artifacts in use, in the form of practices and tools. It also involved an in depth study of collaboration practices used between teams especially distributed. In this phase, we also aimed to focus on project documentation i.e. requirements documents, user stories, product backlogs, burn down charts etc. These documents help to provide insights into the project such as the adopted processes, collaboration practices followed, collaboration tools used, organizational structure, work flow, and project documentation.

6.2.2. Phase 2

In the second phase of the data collection process, the focus was more on gaining an understanding of roles, their dependencies and their communication patterns. In addition, it aimed at gathering awareness information. The awareness data was gathered by parsing project documents to find requirements allocation to roles and their interdependencies, questionnaires responses and interviews with members. Communication data was more rigorous and were based on observation and questionnaires. Team members were asked questions regarding their communication with other colleagues working in a team through personalized questionnaires.

- Awareness data: Project task allocation scheme i.e. Who is working with whom and work items dependencies
- Communication data: Observation on how team members collaborate, questionnaires and interviews

6.2.2.1.Data Collection Methods

The data collection methods should cover all of the possible information from the subjects. Case study was selected to be the method in this study and for this purpose, methodological triangulation (Guion, Diehl, & Debra, 2011) was used to collect data from various sources such as observation, document analysis, questionnaires and focus group discussions. Case studies are not controlled by the researcher which makes it difficult to claim the validity of the results. Therefore, multiple case studies or triangulation methods for data collection was suggested in order to have cooperative evidences (Yin, 1994). The use of multiple data collection methods within the same domain increases confidence in the results and helps in the better understanding of concepts (Thurmond, 2001). Therefore, this research opted to use several methods for data collection (summarized in Table 6.1) as described below:

Table 6.1: Summarizing the results of data collection methods

Data collection Method	Result
Observation	Workflow of the teams, Teams' collaboration practices, Teams interaction with each other
Document Analysis	Functionalities list, User stories, Number of tasks, Teams' interdependencies
Work Diaries	On spot collaboration data recording
Questionnaires	Communication patterns of teams Team members' awareness of each other
Semi structured interviews	Project details, Team details , Processes and practices details, Missed out information in questionnaires

(a) Observation

Observation was used as a mean of understanding the workings of teams, their collaboration and daily practices. It is used as an additional source to enhance the findings from other data collection sources (Yin, 1994). What was observed are how team members communicate with each other, how they collaborate with their distributed partners, and how they develop awareness of their surroundings? What was also probed is how team members communicate with people who are not part of the team but are involved in the process (emergent people).

(b) Document analysis

To analyze the project related documents, requirements specifications, schedules, meeting points and other accessible resources, document analysis was used. Document analysis plays an important role in finding dependencies and traceability between user stories, burn down charts and test documents etc. The list of features or the document bearing features broken into user stories and tasks with the allocation details was an important document for designing questionnaire. The document helped in defining the number of tasks, allocated to which team members and their interdependencies.

(c) Work diaries

The use of a work diary as a data collection tool mitigates the impossibility of physical presence at multiple locations and observing multiple teams simultaneously (Duke, 2012). It is an advantage of the diary method in that it enables the researcher to collect multiple data sets at a time (Lewis, Sligo, & Massey, 2005). It is the research partners own responsibility in the dairy method to write down their experiences, which is time consuming and requires commitment (Bolger, Davis, & Rafaeli, 2003).In this case, the diary method serves the purpose of recording a longitudinal study as things unfold with time, and recording an unusual and often rare phenomena, team's collaboration details and etc. (Bolger et al., 2003). We used event based designs for keeping work diaries in order to get accurate information on the interaction as and when it occurred. The maintenance of work diaries is an important mean of data gathering for ongoing day to day events. It captures on spot data, thus provides better reporting of ongoing events than questionnaires and interviews (Conrath, Higgins, & McClean, 1983). But it was discovered that software developers found it hectic to take time out for this activity in addition to their own scheduled work. Thus, it was planned to be a supporting technique for data collection along with others including questionnaires for communication and awareness social networks.

(d) Questionnaires

Questionnaire was the main source of data collection in order to visualize the communication patterns and awareness structures among agile teams. The questionnaire method is one of the most widely used method for data collection, particularly in social network analysis after work diaries (Gibson, Mathews, Diaries, & Pepys, 1942).

For capturing information from agile teams, it was necessary to collect data at several points of time such as at the start, end and somewhere in the middle of iterations to gain deeper insights on the teams collaboration patterns with respect to requirements maturation.

The questionnaire was designed specific to the team members of the projects under discussion. Each respondent was asked about his awareness levels regarding other team members and their communication with them. Each respondent was provided a list of other team members and tasks. To obtain the details of communication, the respondents were asked about the nature, reason, mode and task for which they communicated. The respondents were asked about their name, work experience and role in the demographic section. The details on the project, user story or task which they worked on and their awareness of their workplace were asked in the next section followed by demographics. The detailed Goal Question Metric (GQM) analysis of the questionnaire (shown in Appendix C) is described in Table 6.2:

Table 6.2: GQM analysis of questionnaire

Goal	**Purpose Issue Object (process) Viewpoint**	**To investigate Communication and Awareness of Agile Teams For requirements-driven collaboration (RDC)**
	User story related questions	
Question	Q1.	Mention the number of tasks you have worked on in this iteration.
Metric	M1	% of tasks allocated to Person X
Question	Q2.	Please mention the number of tasks you have worked on independently in this iteration.
Metric	M2 M3	% of tasks allocation to Person X independently To find the coordination requirements
Question	Q3.	Please mention the number of tasks you needed to rework on after this iteration.
Metric	M4	% of rework
Question	Q4.	If a change occurs in the user stories are you informed about the change.
Metric	M5	Level of communication among teams
Question	Q5.	How you are informed about changes made to User stories.
Metric	M6 M7	Nature of change communication % of most used medium of communication.
Question	Q6.	How often do you face changes in user stories during implementation?
Metric	M8	% of frequency of changes in requirements
	Communication	
Question	Q7.	Have you communicated with Person X.
Metric	M10 M11	% of people communicated with person X. Communication occurrence among team members
Question	Q8.	How often did you communicate with Person X.
Metric	M12	% of frequency of communication
Question	Q9.	Reason of Interaction with Person X
Metric	M13 M14	% of nature of interaction with person X Reason of communication among agile teams
Question	Q10.	Medium of communication you to communicate with Person X
Metric	M15	% of highly used medium of communication
Question	Q11.	The above interaction with Person X was for which task
Metric	M16	% of number of interactions with person X for certain task
	Awareness	
Question	Q12.	Did you know about the Professional Expertise of Person X
Metric	M17	% of people who have general awareness of person X.
Question	Q13.	Did you know about the work status of Person X
Metric	M18	% of people who have current awareness of Person X.
Question	Q14.	Did you know about the task allocated to Person X
Metric	M19	% of people who have task awareness of Person X.

(e) Semi structured interviews

Semi structured interviews were conducted in addition to the questionnaires in order to better understand the company's workflow, project details, and organizational structure. It helps not to miss any useful information. The interviews provided data for gaining basic understanding on the workflow and helped to proceed with data collection in a more structural manner. The semi structured interviews were used to gather information required for the completion of Phase 1 (step 1, 2 and 3). The set of questions is shown in Table 6.3.

Table 6.3: Questionnaire used for semi structured interviews

Project Details	
Q1.	Describe about the nature of Project? (Innovative project, Old project with customized iterations in process or maintenance project)
Q2.	Describe the project scope and domain.
Q3.	What is the overall time span of the project?
Q4.	Which agile development method are you using?
Q5.	Describe the iteration related details i.e. iteration size, number of iterations etc.
Q6.	How many user stories involved per iteration?
Q7.	How many tasks a user story is broken down into?
Q8.	How many tasks completed, left over and repeated in next iteration?
Team details	
Q9.	What is the team size?
Q10.	Describe the team structure?
Q11.	Which are the roles involved and what are their responsibilities?
Process and practice details	
Q12.	Which collaboration practices are being followed by the team members?
Q13.	Which collaboration tools are in use?
Q14.	Which processes are adopted by the team?
Q15.	How project documentation is managed?
Q16.	What is the organization structure and work flow followed by the company?

6.3. Data Analysis

An analysis of the collected data was carried out to draw answers for the research questions. To analyze these data, social network analysis and statistical techniques were used. In the literature, SNA has widely been used for analyzing the structure of social relationships among groups of people or teams e.g. to study informal communication between teams (Boh, Slaughter, & Espinosa, 2007), communication structure of distributed teams (Hinds & Mcgrath, 2006), information brokerage (Marczak et al., 2008), analysis of team performance (Cataldo & Ehrlich, 2011) etc.

In addition, statistical analysis was applied on quantitative data to find the impact of agile team collaboration on team performance. The step by step execution of data analysis is described below.

6.3.1. Constructing communication and awareness RCASN

To investigate the collaboration patterns of agile teams, the data collected from the four study cases were converted into sociograms for further analysis. The data for both communication and awareness were converted into node and edge tables and were saved as Excel spreadsheets. The node tables comprised of source, target, id, and label of nodes. The edge tables comprised of source, target, id, label, direction and weight of the edges. All of the possible communication and awareness links were manually transformed into tables called 'edge tables' in Excel files. For each case study, there were two iterations, 13 communication reasons and 4 awareness types for which a total of 34 edge and node tables were made. These tables were then loaded to Gephi, a network visualization and analysis tool, to transform them into sociograms consisting of actors as nodes and relationships as ties among them.

6.3.2. Preparing data for ties statistical analysis

Using the pair wise coding format, the dataset was imported to Excel spreadsheets in order to calculate some statistics on the communication and awareness reported. This analysis was named as "ties statistics". For each pair of members, one tie was counted if a relationship exists between them. Communication and awareness were considered directional, i.e. if a pair 'source-target' reported communication and this same pair 'target-source' also reported communicating, two instances of communication were recorded. This process was conducted manually by filtering the dataset per the criterion of interest. For example, members from a same location were selected to identify who had communicated with whom from the same site. Members who were not from the same location were selected to identify who had communication with whom cross-sites. We have done the same type of analysis for the roles and for identifying who were the emergent members and their collaboration with others. The ties statistics helped to identify which members have communicated with emergent members.

6.3.3. Constructing networks for social network analysis

The data were converted into square matrices by applying social network analysis measures and saved as separate Excel spreadsheets. Each matrix comprised of actors (team members) on an X and Y axis and populated with relationships between them. For instance, if Project Manager communicated with Product Owner, '1' was placed in the respective row and '0' if otherwise. The diagonal was left empty to ensure no self-communication and awareness is added. These matrices were then loaded into the UCINET tool to perform social network analysis.

6.3.4. Running social network analysis measures on data

Running SNA measures on the data in UCINET was a two-step process. In the first step, after constructing the relationship square matrices in Excel spreadsheets, the files were loaded into the UCINET tool one by one. UCINET then converted all of the files into system useable files with .h and .d extensions. In the second step, all of the SNA measures (defined in Chapter 5 Section 5.4.2 and 5.4.3) were applied to the network files compatible with the UCINET format, i.e. .h and .d extension files. Each measure was applied on all of the networks for all of the four cases and for both iterations, one by one. Side by side, the measures were tested for a couple of networks manually using standard formulas to confirm the accuracy of the results. The measured results files were saved in .txt format and .h format. Later, all of the results were tabulated in separate spreadsheets for each measure for each case and iteration.

6.4. Studying Requirements-Driven Collaboration among Agile teams

In this section the analysis of Case 1- Alpha is described which is carried out according to the framework (described in Chapter 5, section 5.6). The step by step implementation of the proposed framework phases are described below.

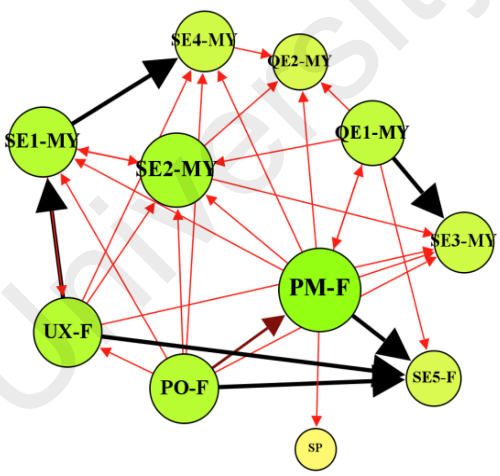
6.4.1. Phase 2- Visualization

In this section, phase 1 of the framework (described in Chapter 5, Section 5.4.2) visualization of the communication and awareness networks of RDC among agile teams is described.

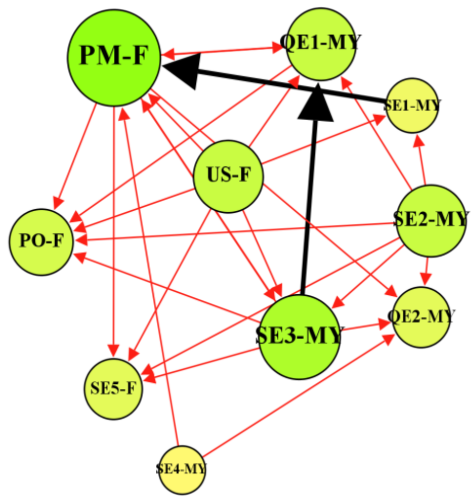
6.4.1.1. RCSN visualization

The collaboration data of the RCATs was transformed into matrices comprising '1' and '0' in excel sheets in order to tabulate the collaboration happened to picture the networks. Later, these matrices were imported in social network visualization and analysis tool "Gephi" to retrieve their respective sociograms or networks. A sociogram is a pictorial representation of a social relationship between actors where actors are represented as 'nodes' and relationship among them as 'ties'. The relationship between team members in this research were represented through Sociograms named as RCASNs. The communication and awareness RCASNs constructed for iteration 1 and 2 are shown in Figure 6.2, 6.3, 6.4 and 6.5, respectively. It can be seen that the nodes represent the team members, and directed ties represent the amount and direction of communication that took place between pairs of members. Similarly, in awareness networks the links represent how much the team members were aware of each other.

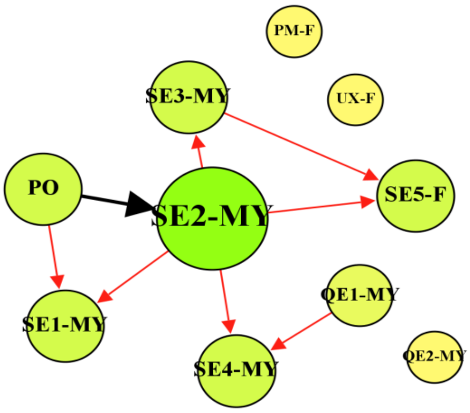
The members' roles and their location (in abbreviation) are indicated as node labels. These sociograms served to provide information about the actor interactions at a glance. The left out or 'isolated nodes' in iteration 1 can be seen in sociograms shown in Figure 6.2 e.g. PM, UX and QE2 in code issues (c), PM, UX,QE2 and SE4 in code refactor (d), UX in code reviews (e), UX,SE5 in quality (i), UX,SE4 and QE2 in sprint planning (i), UX, QE1, QE2 in support (j), and UX, SE1, SE5 in Code synchronization (f). It can be seen from that UX and QE2 were the left out members in most of the communication networks for iteration 1. It is already explained that UX was not the dedicated members for the project; rather he was working on three projects simultaneously. Therefore, UX was not actively participating in the team communication in iteration 1. In addition, it can be observed that in code related discussion including resolving code issues, reviews, refactoring and synchronization the isolated members were PM,QE2, and UX. This shows that UX has nothing to do with coding issues being discussed between the core team members i.e. developers.



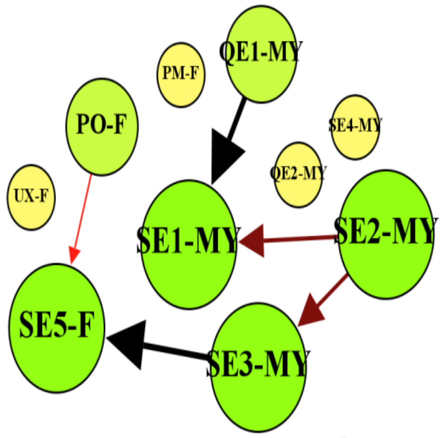
(a) Bugs



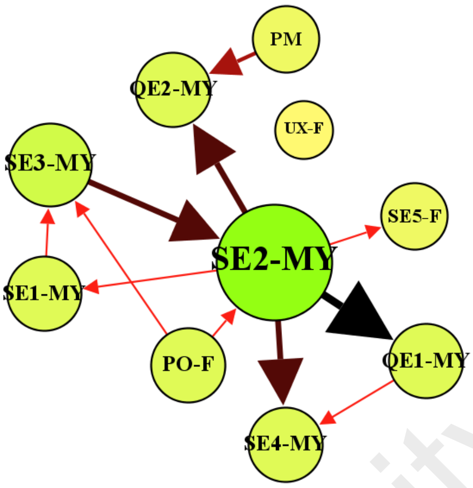
(a) Changes communication



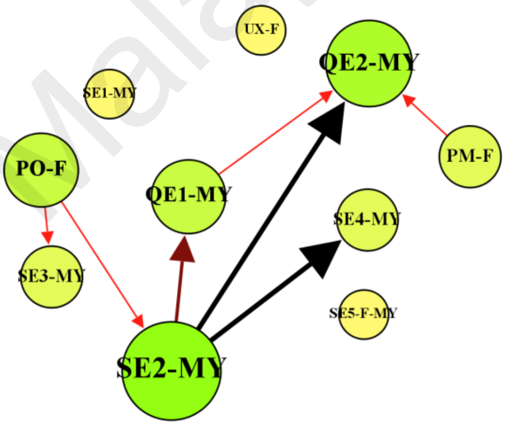
(b) Code issues



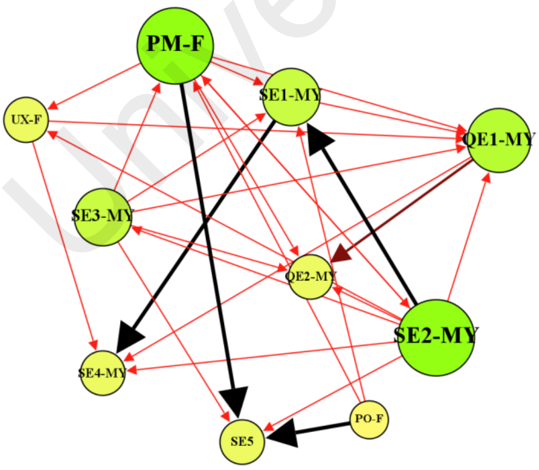
(c) Code refactoring



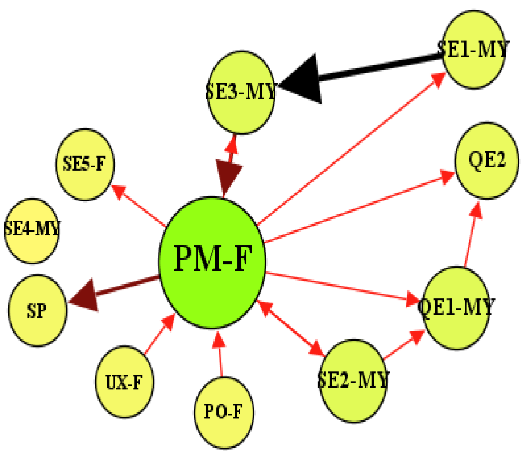
(d) Code reviews



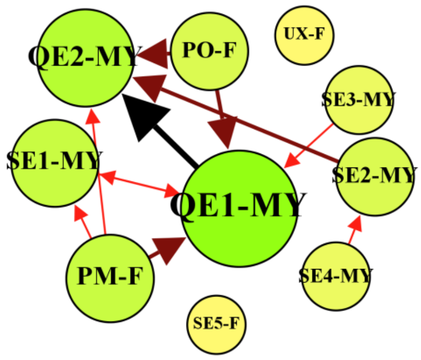
(e) Code synchronization



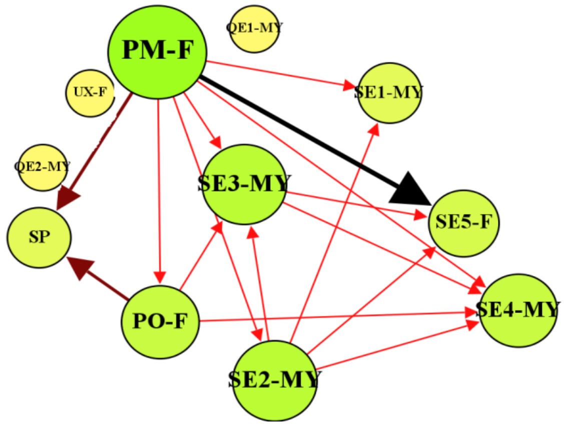
(f) Coordination



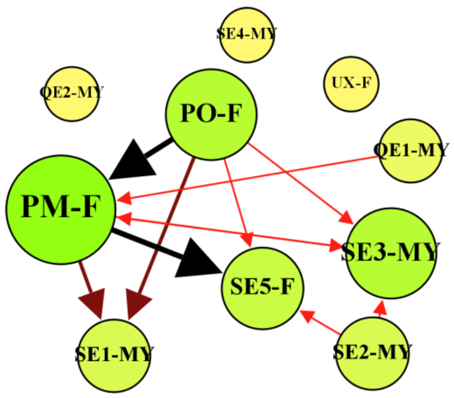
(g) Management



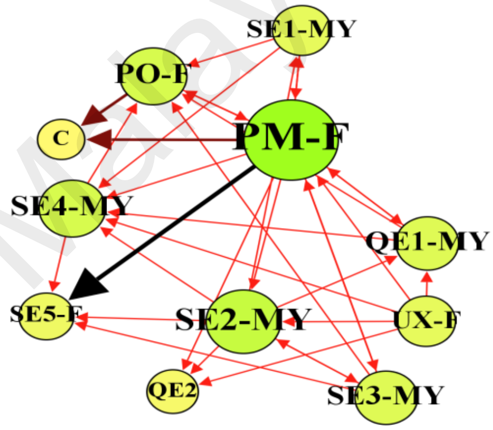
(h) Quality



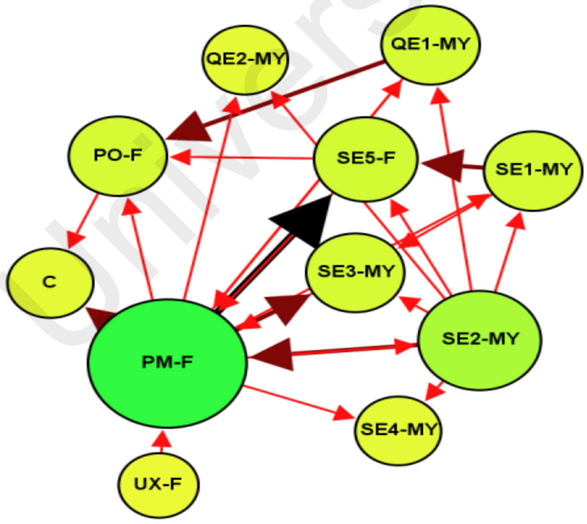
(i) Support



(j) Sprint planning



(k) User story clarification

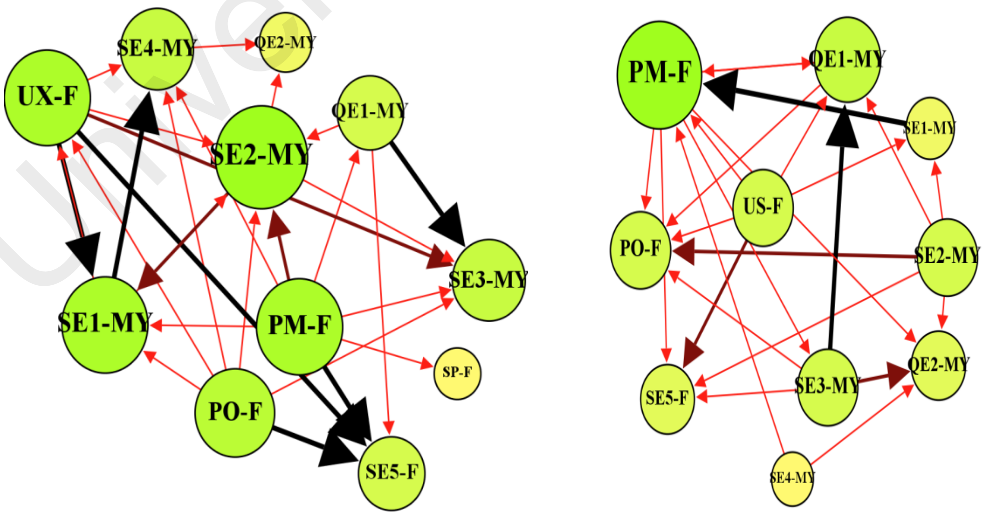


(m)User story negotiation

Figure 6.1: Communication sociograms Iteration 1 case 1

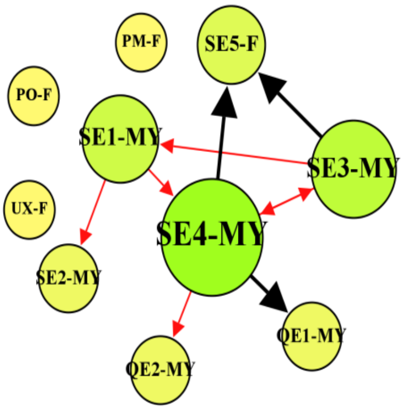
The emergent members involved in networks can be easily identified at a glance on networks. The emergent members are marked with circles to identify. It can be seen that emergent members involved in this iteration were namely Customer (C) and Support team (SP) shown in Figure 6.2 bugs discussion (a), Support (j), Management (h), user story clarification (l) and user story negotiation (m). The customer involvement was involved in user story clarification and negotiation and remained limited to PM and PO. Rest of the team members did not communicate directly with the Customer. The Support Team was involved only in management, support and bugs related issues.

The left-out actors or 'isolated nodes' in Iteration 2 can be seen in the Sociograms e.g.PM, PO, UX in code issues and reviews (Fig 6.3 (c) and (e)), PM,PO, UX, QE1 and QE2 in code refactoring and synchronization (Figure 6.3 (d) and (f), UX in sprint planning and management (Figure 6.3 (k) and (h)), UX and QE2 in User story clarification and negotiation (Figure 6.3 (l) and (m)). Therefore, it can be analyzed that for code issues, reviews and refactoring (shown in Figure 6.3 b, c and d) PM, PO and UX didn't contribute. Similarly, it can be seen that UX remained the isolated member in most of the networks.

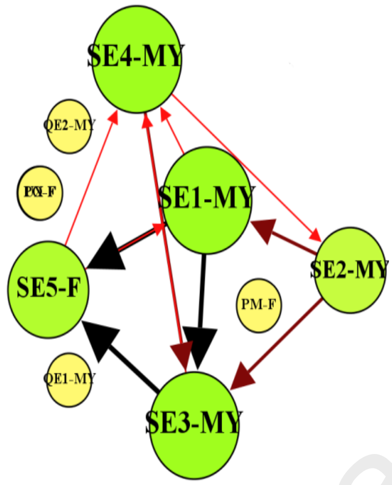


(a) bugs

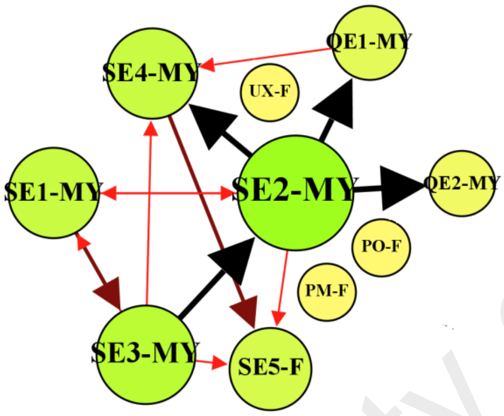
(b) change communication



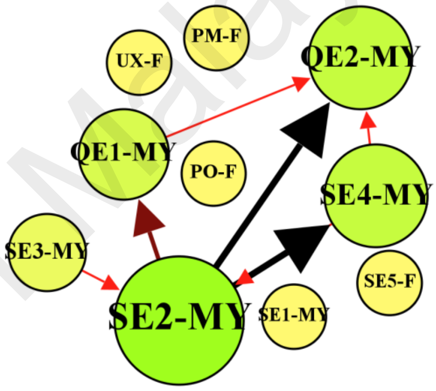
(c) Code issues



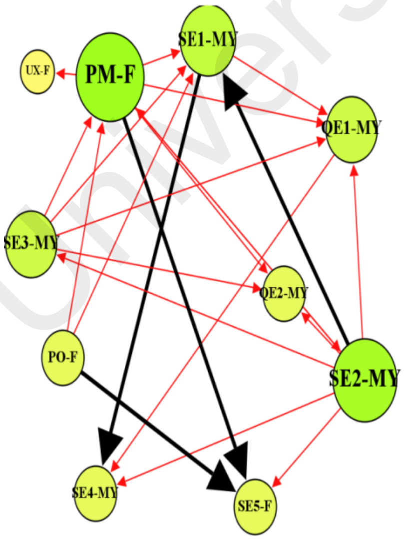
(d) code refactoring



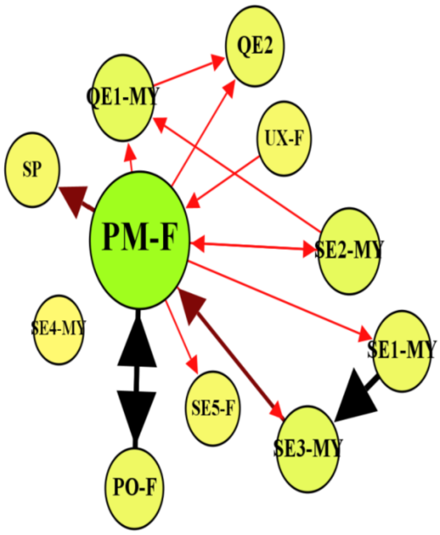
(e) code reviews



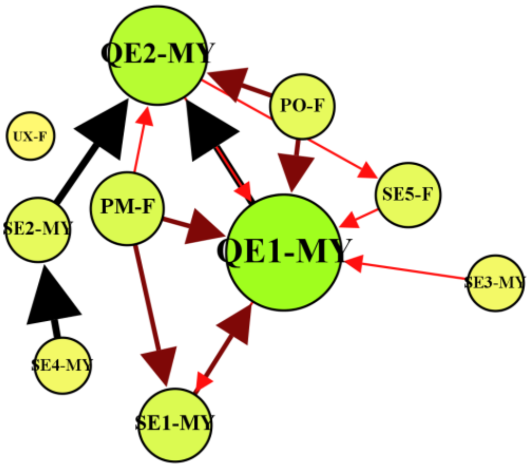
(f) code synchronization



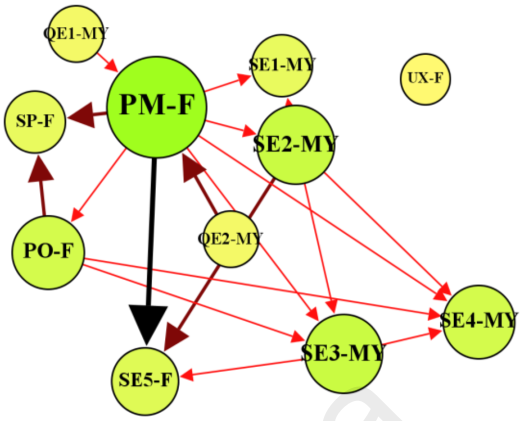
(g) coordination



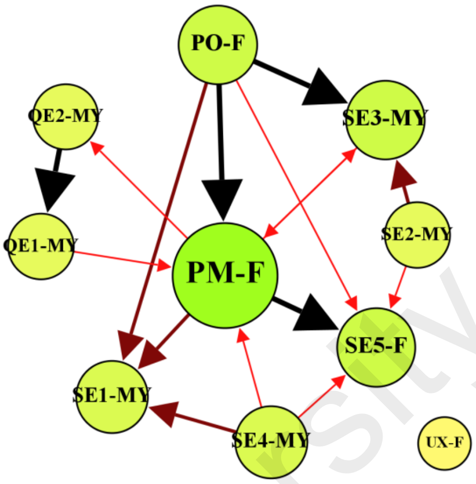
(h) management



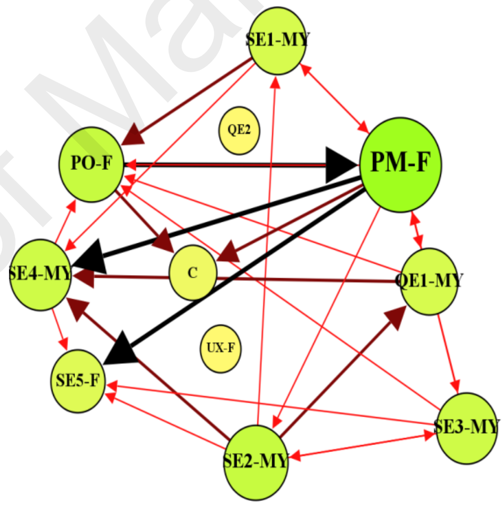
(i) Quality



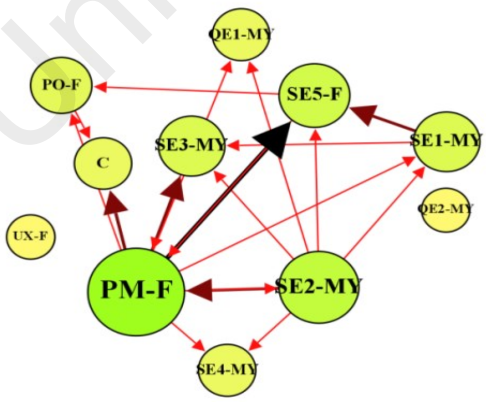
(j) support



(k) sprint planning



(l) user story clarification



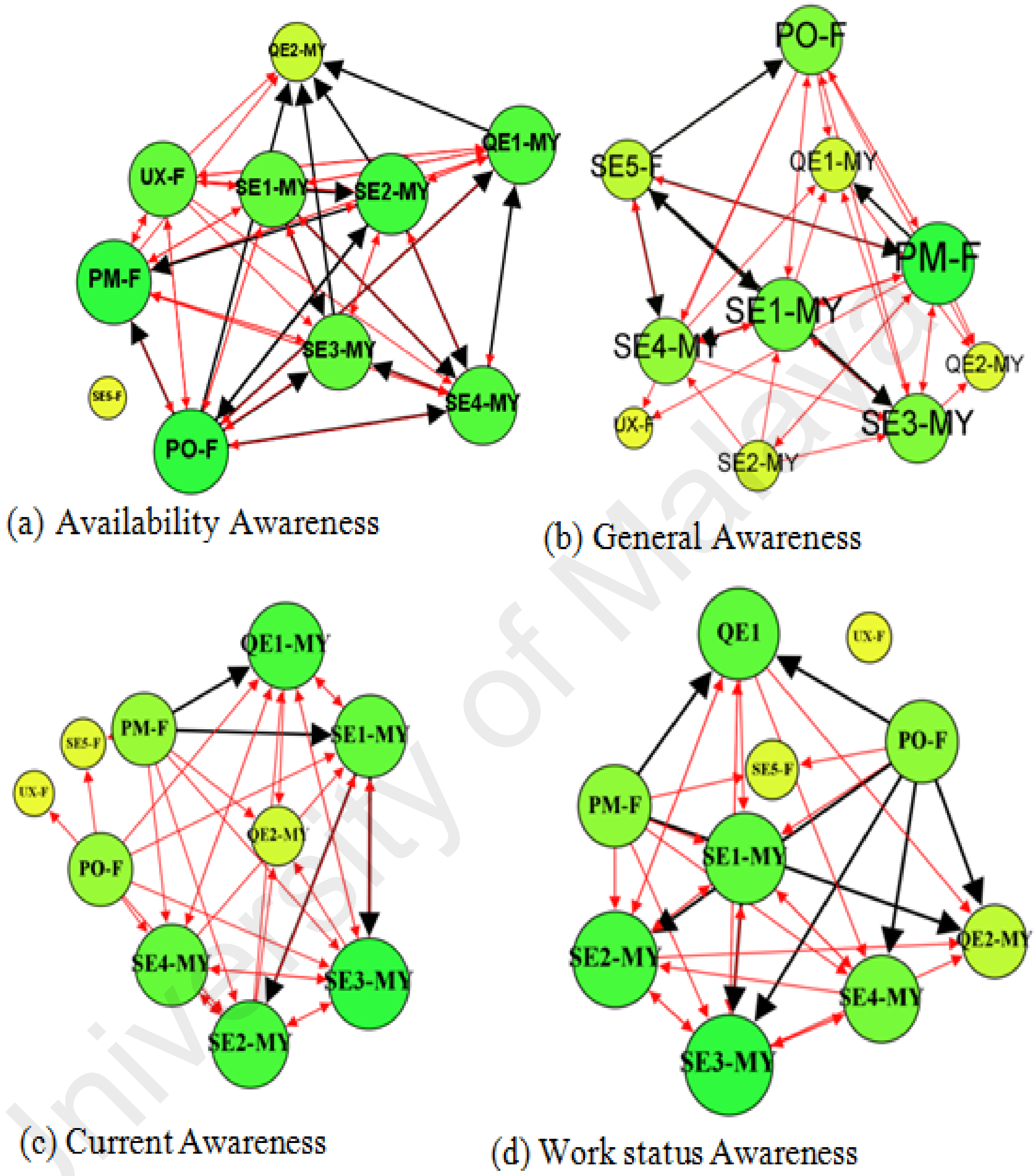
(m)User story negotiation

Figure 6.2: Communication sociograms Iteration 2 case 1

The fact that UX was not a dedicated team member and was involved in at most three projects simultaneously, justifies this situation. In addition, it can be observed that in code related discussion including resolving code issues, reviews, refactoring and synchronization the isolated members were PM, PO and UX. This shows that UX has nothing to do with coding issues being discussed between the core team members i.e. developers. The PO took part in code related discussions in iteration 1 but his absence can be seen in iteration 2. The core members or developers and quality engineers had the most of discussion regarding code issues among themselves.

The emergent members involved in iteration 2 were namely Customer (C) and Support team (SP) shown in Figure 6.2 Support (j), Management (h), user story clarification (l) and user story negotiation (m). The customer involvement remained limited to PO and PM only in user story clarification and negotiation. The Support Team was involved only in management, support and bugs related issues. Hence, it can be seen that sociograms help to understand the collaboration patterns of teams at a glance without going into deeper ties statistics. Therefore, sociograms were constructed for all the considered communication reasons and awareness types for CASE 1-Alpha . The awareness sociograms for iteration 1 and 2 were drawn the same way, shown in Figure 6.3 and Figure 6.4.

It can be seen that team members i.e. PM and PO were aware of emergent members availability and had general awareness of them (Figure 6.4 (a), (b)). However, rest of the team was not familiar of these emergent members. The awareness networks show no isolated members except UX in work status awareness (shown in Figure 6.3 (d) which shows that team members were over all aware of each other but they didn't know about the work status of UX in iteration 1. The less participation of UX in project related discussion might be the reason for this.



(a) Availability Awareness (b) General Awareness

(c) Current Awareness (d) Work status Awareness

Figure 6.3: Awareness sociograms Iteration 1 case 1

In Iteration 2, it can be seen that the team members were yet not aware of emergent members i.e. Customer and Support team. The team members seemed well aware of each other in iteration 2 as there are no isolated nodes. It can be seen in Figure 6.4 (c) and (d) that team members like PO, PM, SE1 and QE1 gained awareness of UX work status and task allocated to him. Thus it can be concluded by having a glance at the sociograms that awareness among team members increased with time.

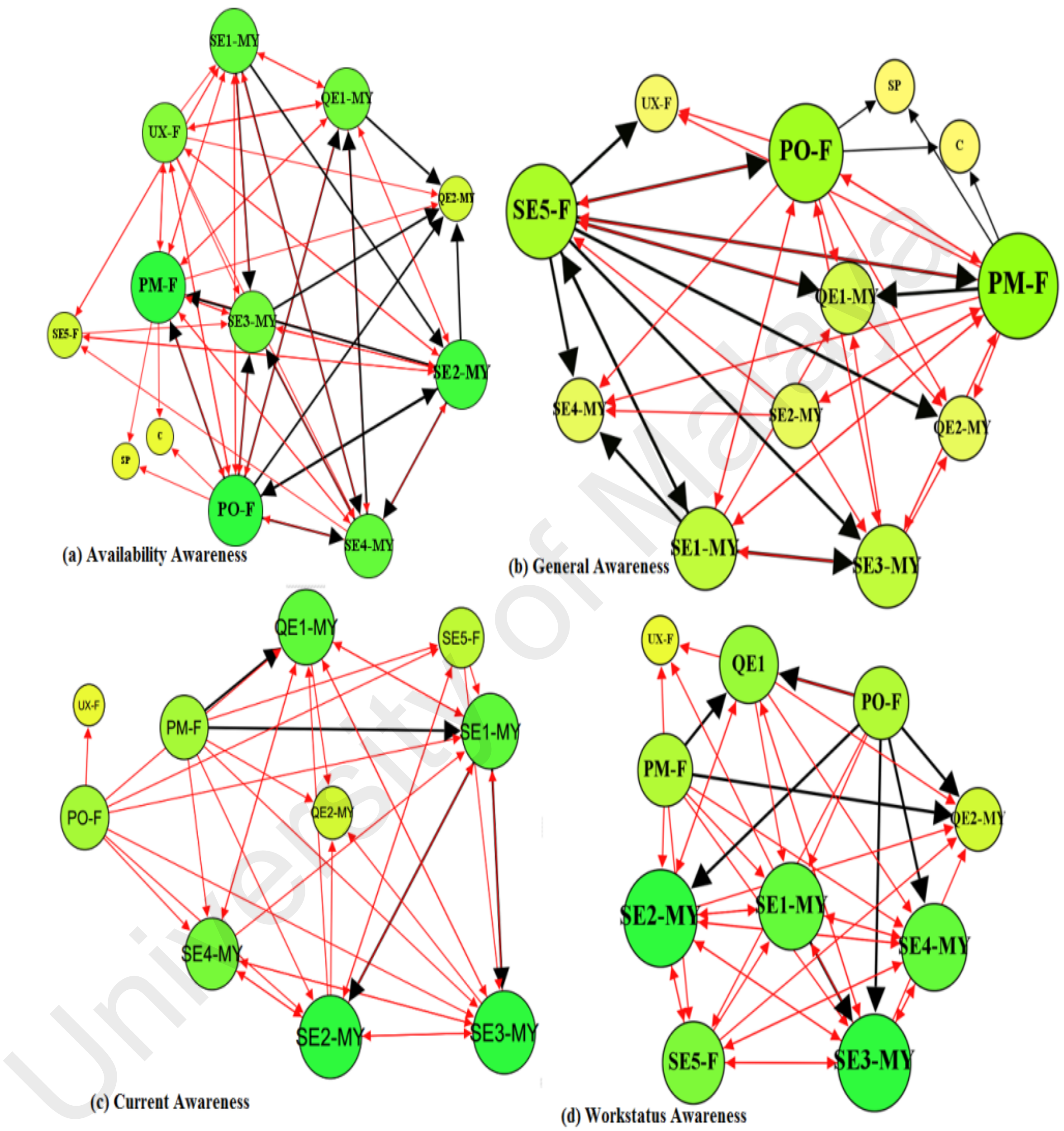


Figure 6.4: Awareness sociograms Iteration 2 case 1

6.4.1.2.RCASN Size

In Case 1 there were total 10 members distributed at two locations. These members are defined as 'assigned members'. The other members who were not a formal part of team and with whom the assigned members collaborated during the iterations are defined as 'emergent members'. Therefore, each network was constructed using the questionnaire data which listed the details of which members communicated with which members and vice versa for awareness. Each communication and awareness network constituted of several assigned members and some of the networks also consisted of emergent members.

Table 6.1 shows the distribution of members for each of the communication reasons. The total number of members involved in the iterations was large than the number of assigned members in some of the communication networks (as shown in Table 6.4). This is due to the incorporation of emergent members. The involvement of emergent members was seen for bugs and defects discussion, support tasks, user story clarification and negotiation. The emergent members are shown in grey boxes in Table 6.4 below. The table describes the details of emergent members' presence and their location (column 1 and 2 of Table 6.4). In iteration 1 it can be seen that all the emergent members were from Finland. The head office being located in Finland might be the reason for having customer and support team there as well. Moreover, it is observed that all of the assigned team members collaborated for changes communication, coordination activities and management tasks. However, for rest of the communication reasons including code issues, reviews, refactoring, synchronization, quality and sprint planning not all of the assigned members communicated with each other. Therefore, the RCASN size is small for these communication reasons showing less member participation.

In iteration 2 (shown in Table 6.4) the involvement of emergent members remained there for the same communication reasons i.e. bugs and defects discussion, support tasks and user story clarification and negotiation. However, a slight increase in team's involvement was noticed for coordination, quality and sprint planning discussions as compared to iteration 1.

Table 6.4: Members allocation for communication networks Iteration 1

Com. Type	It-1					It-2				
	MY	F	Assigned	Emergent	Total	MY	F	Assigned	Emergent	Total
Bugs	5	6	10	1	11	6	5	10	1	10
Changes	6	4	10	0	10	6	4	10	0	10
Code Issues	5	2	7	0	7	6	2	7	0	7
Refactoring	4	2	6	0	6	4	2	6	0	6
Code Review	6	3	9	0	9	6	2	8	0	8
Code Sync.	4	3	7	0	7	3	1	4	0	4
Coordination	6	4	10	0	10	6	4	10	0	10
Management	6	4	10	0	10	5	5	9	1	10
Quality	6	2	8	0	8	6	4	10	0	10
Sprint Planning	4	3	7	0	7	6	4	10	0	10
Support	6	5	10	1	11	6	5	10	1	11
USClarification	6	5	10	1	11	5	5	9	1	11
US Negotiation	6	5	10	1	11	5	5	10	1	11

Table 6.5: Members allocation for Awareness networks Iteration 1 and 2

Awareness Type	Iteration 1				Iteration 2			
	MY	F	Assigned	Emergent	MY	F	Assigned	Emergent
Availability	6	5	10	1	6	5	10	1
General	6	5	10	1	6	5	10	1
Current	6	4	10	0	6	4	10	0
Work Status	6	3	9	0	6	3	9	0

The size of Awareness RCASNs remained same in both of the iterations. The involvement of emergent members (shown by shaded boxes in Table 6.5) was only seen in availability and general awareness.

6.4.1.3.RCASN Density

The second step of analysis involved recoding the dataset in a matrix format to calculate the density of each network and the correlation between them. This analysis has been performed with the help of the UCINet tool. For each pair of member ‘0’ was indicated when the relationship was absent and ‘1’ when the relationship was present. Then, one by one, the networks were loaded to the tool to run the referred measures. Interpretation and discussion of the results, as previously mentioned, is supported by observations.

The density measure yields a value in between 0 and 1. We consider density equal or above 0.5 (50%) as a high-density value, meaning that more than half of the possible ties were reported as present and equal or lower than 0.3 (30%) as low-density value,. The density values for iteration 1 and iteration 2 are reported in Table 6.3 below. The density values for both of the iterations are less than 50%. No network has shown more than 39% of total possible ties. This shows that the communication took place was much lesser than the possible number of ties expected. The density values close to the maximum value attained i.e. 39% are for bugs discussion (30%), user story clarification (31%) in iteration 1 and user story clarification (32%) and negotiation (39%) in iteration 2. The results show that only 15% of the networks reside in between the range of high and low density. However, rest of the 85% networks show lower density values. In low density cases (e.g. code issues, reviews, refactor and synchronization etc.) the communication took place between developers only which turned the density value lower when calculated in comparison with total number of nodes (which was higher). The code related communication happened mostly among the developers and sometimes with quality engineers. The rest of the roles remained out of touch for instance it was not a part of policy for this team to involve Project Manager or Owner in every slight code change. On the contrast it can be seen that for user stories clarification and negotiation whole team was communicating with each other so the density value is comparatively high. Therefore, we can say that low density does not depict low or no communication because even in low density networks adequate communication took place between same roles.

Density values for awareness networks are presented in Table 6.6 below. The density values for iteration 2 are greater than iteration 1 which shows increase in awareness with time. The high density awareness networks are availability awareness for iteration 1 and 2, general awareness, and task awareness for iteration 2. Almost 50% of the networks have density values in high density region (above 50%) and the rest of 50% reside in between high and low density region (above 30%).

Table 6.6: Density Values for communication and awareness networks Iteration 1 and 2

Comm. Reasons	**Density**	
	Iteration 1	**Iteration 2**
Bugs	0.30	0.18
Changes communication	0.28	0.29
Code issues	0.09	0.13
Code refactoring	0.05	0.13
Code reviews	0.11	0.07
Code synchronization	0.08	0.07
Coordination	0.20	0.22
Management	0.12	0.12
Quality	0.12	0.15
Sprint planning	0.15	0.17
Support	0.12	0.16
User-story clarification	0.31	0.32
User-story negotiation	0.21	0.39

Awareness Types	**Density**	
	Iteration 1	**Iteration 2**
Availability	0.54	0.62
General	0.37	0.53
Current	0.38	0.46
Work Status	0.37	0.55

6.4.1.4.Ties Statistics

The pair wise coding format for in depth analysis of collaboration patterns of the teams. We imported the dataset to excel spreadsheets to calculate the statistics of the communication and awareness networks. This analysis process was named as “ties statistics”. For each pair of members communicated with each other or were aware of each other a tie was counted. It is important to mention here that communication and awareness were considered directional, i.e. if a pair of members ‘source-target’ reported communication and this same pair 'target-source' also reported communicating, two distinct instances of communication were considered. This was a manual process conducted filtering the dataset per criterion of interest. An in depth analysis of gathered data was performed from every possible dimension. In this statistical analysis location, roles and allocation based ties classification was considered. For example selected the members from a same location were considered to identify who has communicated with whom from the same site. Likewise, analysis was performed for members who were not from the same location to identify who has communication with whom cross-sites.

(a) Communication RCANS

The communication ties cross and within-site for both of the iterations (shown as IT-1 and IT-2) are shown in Table 6.7.

Table 6.7: Communication RCASNs within and cross-site ties statistics for iteration 1 and 2

Comm. Reasons	IT-1 Cross-site			Within-site			Total	IT-2 Across			Within			Total
	F-MY	MY-F	Total	F-F	MY-MY	Total		F-MY	MY-F	Total	F-F	MY-MY	Total	
Bugs	14	3	17	6	9	15	**32**	13	2	15	5	8	13	**28**
Changes	6	9	15	5	7	12	**27**	5	8	13	5	6	11	**24**
Code issues	2	2	4	0	4	4	**8**	0	2	2	0	7	7	**9**
Code refactoring	0	1	1	1	3	4	**5**	2	2	4	0	7	7	**11**
Code reviews	3	1	4	0	7	7	**11**	0	3	3	0	10	10	**13**
Code sync.	0	0	0	3	4	7	**7**	0	0	0	0	7	7	**7**
Coordination	7	5	12	4	12	16	**28**	6	3	9	3	11	14	**23**
Management	6	2	8	3	3	6	**14**	5	2	7	5	3	8	**15**
Quality	5	0	5	0	6	6	**11**	6	1	7	0	7	7	**14**
Sprint planning	5	1	6	2	3	5	**11**	5	5	10	3	3	6	**16**
Support	8	2	10	2	4	6	**16**	6	4	10	4	4	8	**18**
US clarification	10	9	19	6	9	15	**34**	5	10	15	5	7	12	**27**
US negotiation	7	6	13	6	6	12	**25**	5	4	9	6	5	11	**20**
Total	**73**	**41**	**114**	**38**	**77**	**115**	**229**	**58**	**46**	**104**	**36**	**85**	**121**	**225**

No surprising change in total number of communication ties recorded in both of the iterations was noted. In iteration 1 cross-site communication happened 18% more than it happened in iteration 2. However, within-site communication for iteration 1 and 2 are approximately same with a nominal difference (115 in iteration 1 and 121 in iteration

2). Six out of ten members were located in Malaysia therefore within-site ties are large in number (77 in iteration 1, 85 in iteration 2) as compared to Finland (38 in iteration 1, 36 in iteration 2). On the contrary, more communication originated from team members located in Finland towards Malaysian team members (73 in iteration 1, 58 in iteration 2). The presence of managerial members i.e. Project Manager, Product Owner in Finland can be the reason of more communication origination from Finland.

The communication ties statistics for cross and within-roles for both of the iterations are shown in Table 6.8 (a) and (b).

Table 6.8: (a) Communication RCASNs within-roles ties statistics and (b) cross-roles ties for iteration 1 and 2

(a)Case 1 - Role	**Iteration 1**	**Iteration 2**
Project Manager (PM)	NA	NA
Software Developer (Dev)	53	75
Quality Engineer (Tester)	5	5
User Experience Designer (UX)	NA	NA
Project Owner (PO)	NA	NA
Total	**58**	**80**

(b)Case 1- Pair of roles	**Iteration 1**	**Iteration 2**
PM-Dev	46	41
PM-UX	5	3
PM-PO	8	9
PM-Tester	20	16
Dev-UX	13	8
Dev-PO	25	18
Dev-Tester	35	35
Tester-UX	4	1
Tester-PO	5	4
UX-PO	2	2
Total	**163**	**137**

The team members' are paired with each other in such a way that reciprocal communication is considered distinct for example if developers communicated with Project Manager and vice versa we counted two communication ties for both. In this case we had 5 Developers and 2 Quality Engineers and rest of the roles were distinct. Therefore, it can be seen that within-roles communication happened the most among developers (53 in iteration 1 and 75 in iteration 2). However, communication between Quality Engineers was recoded to be very low as compared to developers (5 in both of the iterations). As described earlier, rest of the roles were distinct so '0' was put in front of them. The cross-roles communication ties values are large where developers were present for instance the communication of developers with Project Manager (46 in iteration 1, 41 in iteration 2), Project Owner (25 in iteration 1, 18 in iteration 2), Quality Engineers (35 in iteration 1 and iteration 2). The least number of communication ties were recorded for communicating pairs including User Experience Designer (UX). It can be concluded that with time and growth of project the communication ties increased within and cross- roles (shown in Table 6.8(a) and (b)).

The communication between team members was also analyzed from the perspective of 'assigned' and 'emergent' members for both of the iterations, shown in Table 6.9. The analysis showed that most of the communication took place among 'assigned' members and involvement of emergent members was noticed to be nominal (8 out of 229 and 225 in both of the iterations). This shows that agile teams work closely with each other and are less dependent on outer resources for information seeking. This aspect clearly determines the information flow among agile teams which proves be highly centered within team.

Table 6.9: Communication RCASNs allocation ties statistics for iteration 1and 2

	Iteration 1		**Total**	**Iteration 2**		
Comm. types	**Assigned**	**Emergent**		**Assigned**	**Emergent**	**Total**
Bugs	31	1	32	27	1	28
Changes	27	0	27	24	0	24
Code issues	8	0	8	9	0	9
Code refactoring	5	0	5	11	0	11
Code reviews	11	0	11	13	0	13
Code synchronization	7	0	7	7	0	7
Coordination	28	0	28	23	0	23
Management	13	1	14	14	1	15
Quality	11	0	11	14	0	14
Sprint planning	11	0	11	16	0	16
Support	14	2	16	16	2	18
User Story clarification	32	2	34	25	2	27
User Story negotiation	23	2	25	18	2	20
Total	221	8	229	217	8	225

a) Awareness RCASN

The awareness of team members was analysed according to awareness within and across sites, within and cross-roles and according to their allocation as ‘assigned’ and ‘emergent’ members. The statistics of recorded ties cross and within-site are shown in Table 6.10. At a glance, it can be seen that awareness among team members increased over the period of time (175 in iteration 1 and 197 in iteration 2). However, the difference between awareness of team members located within the same site and across the different sites is not remarkable. But overall awareness among team members has increased with time equally within and cross site.

The awareness ties statistics cross and within-roles are shown in Table 6.10. The awareness ties within same roles increased with time e.g. for developers 45 in iteration 1 and 63 in iteration 2. However, there is no significant raise seen in awareness ties for pairs of distinct roles in the second iteration, shown in Table 6.11.

Table 6.10: Awareness RCASNs within and cross-site ties statistics for iteration 1 and 2

Case 1	IT-1 Cross-site			Within-site			Total	IT-2 Cross-site			Within-site			Total
Awareness type.	F-MY	MY-F	Total	F-F	MY-MY	Total		F-MY	MY-F	Total	F-F	MY-MY	Total	
Availability	18	12	30	12	20	32	**62**	21	14	35	10	21	31	**66**
General	15	8	23	8	13	21	**44**	14	8	22	11	13	24	**46**
Current	11	0	11	3	21	24	**35**	13	1	14	4	21	25	**39**
Work status	12	4	16	2	16	18	**34**	17	5	22	4	20	24	**46**
Total	**56**	**24**	**80**	**25**	**70**	**95**	**175**	**65**	**28**	**93**	**29**	**75**	**104**	**197**

The ties between roles i.e. assigned and emergent were recorded and shown in Table 6.12. Awareness of emergent members was considerably less as compared to the awareness team members had of each other. This was due to the small number of emergent members involved and their interaction mainly with managerial members like

Project Owner and Project Manager. In case 1 it was noticed that team members had not collaborated with emergent members directly rather PM and PO served as intermediaries between team members and emergent members which in this case were Support Team and Customer. The allocation ties show that regarding emergent members the team had general and availability awareness. The other two types of awareness considered for this research were not applicable on emergent members. The emergent member was not assigned any task therefore current and work status awareness was not applicable for them. The general awareness of emergent members increased in iteration 2 as compared to iteration 1. Overall, awareness among assigned and emergent roles increased in iteration 2.

Table 6.11: Awareness RCASNs (a) within-roles ties statistics and (b) cross-roles ties for iteration 1 and 2

Case 1 Role	Iteration 1	Iteration 2
Project Manager (PM)	NA	NA
Software Developer (Dev)	45	63
Quality Engineer (Tester)	4	5
User Experience Designer (UX)	NA	NA
Project Owner (PO)	NA	NA
Total	**49**	**68**

Case 1 Pair of roles	Iteration 1	Iteration 2
PM-Dev	29	29
PM-UX	3	5
PM-PO	4	4
PM-Tester	9	8
Dev-UX	5	6
Dev-PO	25	24
Dev-Tester	30	30
Tester-UX	3	3
Tester-PO	8	7
UX-PO	4	5
Total	**120**	**121**

Table 6.12: Awareness RCASNs allocation ties statistics for iteration 1 and 2

	Iteration 1		**Total**	**Iteration 2**		
Awareness type	**Assigned**	**Emergent**		**Assigned**	**Emergent**	**Total**
Availability	58	4	62	62	4	66
General	42	2	44	42	4	46
Current	35	0	35	39	0	39

Work status	34	0	34	46	0	46
Total	**169**	**6**	**175**	**189**	**8**	**19**

6.4.2. Phase 3- Characterization

To study the structures of communication networks (all reasons) and awareness networks (Availability, General, Current and Work status) a detailed analysis was conducted using ties statistics and SNA measures i.e. Network Centralization, Clique, and Ties reciprocity. To investigate the information flow among agile teams, the SNA measures used are Cut points, Reachability, and Component. Later on, supplementing these measures Degree centrality was added to measures used for studying characterization of networks and Betweeness, Eigenvector centrality, clustering coefficient were added to the measures used for studying information flow. The description is as under:

6.4.2.1.RCASN Centralization

The centralization value for works determines the dispersion of ties around nodes. If a network is centralized the value of centralization lies closer to 1. It means that all the nodes communicated with the central node the most. On the contrary, if the nodes have dispersed ties around multiple nodes the network is decentralized and value of centralization is lesser than 1. The communication networks centralization percentage values for iteration 1 and 2 are shown in Table 6.13.

Table 6.13: Communication RCASN Centralization for iteration 1 and 2

Communication Reasons.	**Iteration 1**		**Iteration 2**	
	Outdegree	**Indegree**	**Outdegree**	**Indegree**
Bugs	45.83	31.944	54.44	30
Changes	47.22	47.22	41.81	30.90
Code issues	44.44	16.66	35.45	13.63
Code refactoring	20.88	20.88	26.38	26.38
Code reviews	55.55	13.88	43.63	21.81

Code synchronization	31.99	31.99	26.36	26.36
Coordination	72.22	44.44	58.18	25.45
Management	83.33	22.22	74.54	30.90
Quality	22.72	33.63	20.00	52.72
Sprint planning	67.77	31.11	29.09	29.09
Support	46.66	22.22	60.00	27.27
User Story clarification	84.44	35.55	58.88	46.66
User Story negotiation	72.22	35.55	50.00	50.00
Mean	53.48	29.79	44.52	31.62

The values equal to and above 60% (0.60) are considered high (closer to 1). Table 6.13 shows that coordination 72.22 (0.72), management 83.33 (0.83), sprint planning 67.77 (0.67), user story clarification 84.44 (0.84) and negotiation 72.22 (0.72) stand out among others. These networks have centralization value closer to 1 which shows that most of the communication ties were around one or several actors. It has already seen in degree centrality and ties statistics that PM and PO were the two active and prominent actors in these networks with high centralization values. Therefore, most of the communication was either originated or targeted to PM and PO making the centralization value close to 1. However, 8 out of 13 networks have out degree centralization value lower than 60% (0.60), showing that information distribution was not limited to only few members. It implies that for most of the networks the information flow was dispersed among all the actors. The in-degree centralization values for iteration1 were largely less than 60% (0.60). This implies that information received by most of the members.

In iteration 2, support and management networks have out degree centralization values 74.54 (0.74) and 60.00 (0.60) closer to 1. It implies that most of the support and management related discussion was targeted to several members only. The previous discussion of in and out degree and ties statistics has already clarified that PM and PO were the two members largely contacted for support and management activities by all of the team members. Rest of the networks' out and in degree centralization values are lesser than 1 showing dispersion of information among variable actors. However, the mean of in and out degree centralization values show equal dispersion of communication ties among team members, as all the mean values are less than 60% (0.60).

In awareness RCASN the centralization index is close to 1 for out degree centralization value of availability 60.90 (0.60) and general awareness 79.09 (0.79) in iteration 1, as shown in Table 6.14. The current and work status awareness networks' centralization index was less than 0.60, showing that members have awareness of their colleagues work status and current task. In iteration 2, only general awareness out degree centralization index was 61.81. However, the rest of networks showed lower indexed ranging from 19.09 to 51.81 confirming decentralized awareness patterns. The mean value for in and out degree centralization show centralization only for out degree centralization index in iteration 1 and for the rest awareness was dispersed among all the members.

Table 6.14: Awareness RCASN Centralization for iteration 1 and 2

Awareness Types.	Iteration 1		Iteration 2	
	Outdegree	**Indegree**	**Outdegree**	**Indegree**
Availability	60.90	17.27	51.81	19.09
General	79.09	24.54	61.81	29.09
Current	48.61	34.72	38.89	38.89
Work Status	50.00	36.11	41.66	27.77
Mean	59.65	28.16	48.54	28.71

6.4.2.2.RCASN Ties Reciprocity

The ties reciprocity measure helps to calculate the reciprocity of ties i.e. if A is connected to B; B is also connected to A. There are two variants in calculating reciprocity **Dyad based** in which the number of reciprocated dyads are divided by the number of adjacent dyads. And **Arc based** in which the number of reciprocated arcs divided by the total number of arcs. Both dyad and arc based ties reciprocity was calculated for communication and awareness networks to find out actors involved in a reciprocal relationship (Dyad based) and percentage of ties reciprocated ties (Arc based), shown in Table 6.15.

The overall results of arc and dyad based ties reciprocity provides very low values for both of the methods and iterations. For several networks like code refactoring, reviews, synchronization and sprint planning in iteration 1 and support in iteration 2 the values for both type of reciprocity was low as 0. Apart of having low values for both of the reciprocity types arch based reciprocity has higher values as compared to the dyad based values in both of the iterations. This shows that there are more pairs of actors with communication reciprocity than the actors in a mutually reciprocal communication relationship. The overall low values show that communication was not mutually reciprocated for all the communication reasons. This shows that information flow in agile teams depends on the role played by the actors. For instance in several cases members need to report to PM but it was not required for the PM to respond each of them. This way the reciprocity of communication was nullified.

Table 6.15: Ties Reciprocity Communication and awareness RCASN iteration 1 and 2

Communication Reasons.	**Iteration 1**		**Iteration 2**	
	Arc based	**Dyad based**	**Arc based**	**Dyad based**
Bugs	0.22	0.01	0.21	0.12
Changes	0.15	0.08	0.07	0.04
Code issues	0.25	0.14	0.22	0.12
Code refactoring	0	0	0.36	0.22
Code reviews	0	0	0.33	0.2
Code sync	0	0	0	0
Coordination	0.22	0.12	0.10	0.05
Management	0.15	0.08	0.42	0.27
Quality	0.18	0.01	0.29	0.16
Sprint planning	0	0	0.13	0.06
Support	0.15	0.08	0	0
US clarification	0.29	0.17	0.46	0.29
US negotiation	0.26	0.15	0.74	0.59
Mean	**0.15**	**0.06**	**0.26**	**0.16**

(b)Awareness types				
Availability	0.73	0.58	0.77	0.63
General	0.44	0.28	0.68	0.52
Current	0.45	0.29	0.47	0.31
Work status	0.23	0.13	0.48	0.31

Ties reciprocity calculated for awareness iteration 1 and 2 shows that like communication, awareness networks also have higher values for Arc based reciprocity as compared to dyad based. It shows that more team members were aware of each other than the pairs of members. In iteration 1 only availability awareness value was 73% (0.73) higher than 60% (0.60). However, the rest of values for general, task and current awareness are less than 60%. However, in iteration 2 the arc based reciprocity values for availability (0.77) and general awareness (0.688) are in the high value range. For dyad based reciprocity only availability value is high (0.63). The rest of values for dyad based reciprocity in iteration 2 were low ranging from 0.31 (current and work status) to 0.52 (general). The values for awareness ties reciprocity are comparatively larger than communication values which show that team members were mutually more aware of each other than the mutual communication happened between them.

6.4.2.3.RCASN Clique

A Clique is defined as a subset of a network in which members (nodes) are more closely tied to one another than with the rest of the members of the network (Hanneman & Riddle, 2005). For instance, in real life social networks friendship networks such cliques form within members on the basis of common interests, age, gender and ethnicity etc. Clique test was applied on the communication and awareness networks to identify the least number of actors connected with each other (i.e. no null ties exist). The absence of connected group of members i.e. cliques can be seen for most of the communication reasons in iteration 1. The only communication reasons for which the cliques existed are user story clarification (1) and user story negotiation (6), in iteration 2. This shows that in iteration 2 nodes were well connected for user story negotiation only and the possible cliques present are shown in Appendix C, Table C.54.

The study of cliques shows very interesting social behaviours of the team members for instance Project Manager remained the key member in most of the cliques along with core members like developers and quality engineers for user story negotiation. Moreover, cross-site communication can also be seen among members communicating with each other. However, the absence of cliques in iteration 1 shows that there were no closely connected groups (with maximum three nodes). The absence of cliques does not indicate least or no communication among members in iteration 1. But it can be said that there was no set members in all the networks who communicated closely with each other (with no null ties). Therefore, cliques were missing in iteration 1 but dyads were there which means that not more than two members closely communicated with each other.

Number of cliques recorded for awareness networks for both of the iterations are shown in Table 6.16. The awareness networks had large number of cliques present than communication networks. The number of cliques found in both of the iterations was large for availability and general awareness. The team members grew working closely and gained more awareness of each other in iteration 2 as the total number of cliques increased in iteration 2. The cliques found in awareness networks are shown in Appendix C, Table C.55.

Table 6.16: Awareness RCASN's Clique values for iteration 1 and 2

No. of Cliques	**Case 1**	
Awareness Types.	**Iteration 1**	**Iteration 2**
Availability	4	6
General	4	10
Current	4	2

Work Status	1	3
Mean	3.25	5.25

The cliques show some interesting recurrent patterns of members working closely with each other. For instance Project Manager, Product Owner and in availability awareness , Project Manager and developers in general awareness, developers in current and work status awareness were the recurrent patterns. This shows that most of the developers were aware of their role sharers work status and task allocated to them. The managerial members of team i.e. PM and PO were not found in the cliques for current and work status awareness. This doesn't mean that PM and PO had no knowledge of what was going on in the team, rather it justifies the definition of clique in which all the members have bidirectional ties among each other. However, for current and work status awareness the team members had no knowledge of task and work status of PM and PO. Therefore, the awareness was unidirectional i.e. from PM or PO to developers or quality engineers. There was an average of three cliques present in each of the awareness networks which shows that nodes were well connected and members were well aware of each other in iteration 1.

In iteration 2 the number of cliques increased for all the awareness types except current awareness which decreased to almost 50% (4 in iteration 1, 2 in iteration 2). However, the number of cliques found in rest of the awareness reasons increased contributing an average of having 5 cliques per awareness network. The highest number of cliques found in iteration 2 was for general awareness (10 cliques). This shows that the team got more aware of each other with the growth of project.

The members in each clique show recurrent patterns for example (i) PM, PO and developers in availability awareness, (ii) PO and developers in general awareness, and (iii) developers in current and work status awareness. Moreover, the cliques show that developers remained omnipresent in all the awareness types with PM, PO, QE and UX. In addition this also confirms that developers were core members in the team having knowledge about rest of the members.

From this detailed analysis, it can be concluded that cliques were organized of several core members (PO, PM and developers) actively communicating with each other for exchange of information and were well aware of each other.

6.4.2.4.RCASN Core Periphery

To have more fine grain analysis of network structure, core periphery is used. Core Periphery describes the network structure in terms of two sets of nodes i.e. core and periphery. Core has nodes that are closely connected to each other and periphery has nodes that are loosely connected to each other. The values index between 0 and 1. The values close to 1 show strong core and periphery structure. The values attained for communication and awareness networks are showed in Table 6.17.

Table 6.17:Communication &Awareness RCSN Core Periphery values iteration 1 and 2

Comm. Reasons	Iteration 1	Iteration 2
Bugs	0.18	0.301
Changes	0.42	0.335
Code issues	0.49	0.645
Code refactoring	0.29	0.616
Code reviews	0.49	0.577
Code sync	0.57	0.95
Coordination	0.71	0.558
Management	0.52	0.704
Quality	0.50	0.569
Sprint planning	0.46	0.283
Support	0.70	0.459
User Story clarification	0.43	0.327
User Story negotiation	0.69	0.567
Mean	0.49	0.530

Awareness types.	Iteration 1	Iteration 2
Availability	0.80	0.274
General	0.748	0.42
Current	0.70	0.714
Work Status	0.00	0.533
Mean	0.56	0.485

Only 46% of the networks including coordination (0.71) support (0.70) and user story negotiation (0.69) (above 0.70) in iteration and code issues (0.645), code refactoring (0.616) and management (0.704) had values close to 1. So, no raise in the number of networks with core periphery index close to 1 was observed.

The members of core and periphery groups for each communication reason are shown in Appendix C, Table C.56.

The managerial members i.e. Product owner and Project Manager appeared almost in all the networks as core members except for code issues, in iteration 1. Project Manager preceded the Product Owner in this because PM was the one closely in contact with the team, Product owner and emergent members simultaneously. Therefore, PM was closely coupled with members in reciprocal communicating ties. The UX was found in periphery group always which shows that he was loosely connected with other team members. As explained already that UX was not a dedicated member of the team; hence his involvement with this project and team was minimal.

PM was present in core members of all the networks except the ones related to code activities including code issues, reviews, refactoring and synchronization, in iteration 2. For code related networks developers were in core group that shows their close interactions. The occurrence of emergent members in 30% of the networks was in periphery group. The reason behind is that PM or PO were the only members who communicated with the emergent members i.e. customer and support team. The team members used to communicate their issues and reservations to PM or PO and they were the ones who communicate with emergent members and resolve team's issues. The PM declared that it was not imposed on team members to stay limited from the emergent members yet that was way their communication emerged. Later on the members felt at ease in not communicating directly with the emergent members due to time and availability issues.

Table 6.17 presents the core periphery indexes for awareness networks iteration 1 and iteration 2. The values of core periphery measure are close to 1 (above 0.70) for availability (0.80), General (0.74), and Current (0.70) in iteration 1. The core periphery is 0 for work status which shows high degree of decentralization. This shows that all of the members except UX were mutually aware of each other's work status. In iteration 2, only current awareness core periphery index is close to 1 and rest of the awareness networks show that core-periphery structure was not dominant and this shows networks decentralization. The core and periphery groups attained are shown below and details can be seen in Appendix C, Table C.57 for iteration 1 and 2.

Project Manager remained the core member in all of the awareness networks except current awareness in both of the iterations. This is because when members were asked about awareness of their colleagues' task, it was inapplicable for PM and PO. That is why the core periphery index of current awareness was high. The emergent members were present in periphery group of general and availability networks for both of the iterations which shows that most of the members had no mutual awareness of emergent members except a few i.e. PM and PO. Similarly, the UX was also in periphery group because most of the members had no knowledge about his work status because he was not solely working with this team and project. The absence of UX in most of the daily stand up meetings and sprints also make his colleagues unaware of him.

6.4.3. Information Flow among Communication and Awareness networks

To identify the patterns of knowledge exchange among agile teams several measures determining the information flow are used such as: component, degree centrality, reachability, and cut points.

6.4.3.1. RCASN Degree Centrality

The degree centrality is about determining the values of in and out-degree of the nodes in a certain network. The greater in and out degrees imply that the node has greater number of ties. This determines that particular actor (node) was at a prime position in the network which makes him 'prominent' (Hanneman & Riddle, 2005). Such nodes are capable to deliver more information and are accessible to most of the network members and resources. To determine the degree centrality of the communication and awareness networks in and out degree of the actors in each network is taken into account. Table 6.18 show the in and out degree stats of communication networks for both of the iterations. In iteration 1 (Table 6.18) it can be noticed that the nodes with recurrent high out degree values are PM and SD2. The developer SD2 remained active and prominent in sending information regarding code issues, refactoring, reviews, synchronization and coordination of activities. On the contrary no such recurrent pattern can be seen in case of in degree. The least values of in degree are seen for code issues (1), code reviews (1) and code refactoring (2). The reason behind this is participation of only core members i.e. developers in code related communication.

In iteration 2 (Table 6.18) it can be noticed that the node with recurrent high out degree value is PM. The developer SD2 was prominent again in communication regarding code issues, refactoring, reviews, synchronization and coordination of activities. On the contrary no such recurrent pattern can be seen in case of in degree. The least values of in degree are again noted for code issues (2), code reviews (3) and code refactoring (3). But it can be seen that in degree values of nodes in iteration 2 are higher than the values in iteration 1.

Table 6.18: Communication RCASN Degree Centrality for iteration 1 and 2

It 1	OutDegree			InDegree			It-2 OutDegree			InDegree		
Comm.Reasons	Mem	High degree Mem	Deg	Mem	High degree Mem	Deg.	Mem	High degree Mem	Deg	Mem	High degree Mem	Deg
Bugs	10	PM-F	6	10	SD2	5	11	PM	7	11	SD2,SD3	5
Changes	10	UX	6	10	PM	6	10	SD2	6	10	PO,PM	5
Code issues	10	SD2	4	10	SD1,SD4,SD5	1	10	SD4	4	10	SD4,SD5	2
Code refactoring	10	SD2	2	10	SD1,SD5	2	10	SD1	3	10	SD4,SD3	3
Code reviews	10	SD2	5	10	SD1,SD2,SD5,QE2	1	10	SD2	5	10	SD5	3
Code syn	10	SD2	3	10	QE2	3	10	SD1	3	10	QE2	3
Coordination	10	SD2	7	10	QE1	5	10	SD2	7	10	SD1	4
Management	11	PM	8	11	PM	3	11	PM	8	11	PM	4
Quality	10	QE2	6	10	QE1	5	10	PM	3	10	QE1	6
Sprint planning	11	PM	7	11	SD4	4	10	PM,PO	4	10	PM,SD5	4
Support	11	PO	5	11	PM,SD3,SD5	3	11	PM	7	11	SD4	4
U.S clarification	11	PM	10	11	SD4	6	11	PM	8	11	PM	7
U.S negotiation	11	PM	8	11	PM	5	11	PM	8	11	PM	8

For awareness RCASN iteration 1 (Table 6.19) the only nodes with recurrent high out degree value are PM and PO. However, there is no such recurrent pattern can be seen in case of in degree. High in and out degree values were there for availability and general awareness. The inclusion of emergent members in availability and general awareness networks is one of the reasons for large number of ties contributing to high values of in and out degree.

For awareness RCASN iteration 2 the only nodes with recurrent high out degree value are PM and PO. The in and out degree values show that PM and PO were the most familiar with rest of the team members. The values of awareness out degree have increased in iteration 2 for all types of awareness and for work status awareness only in iteration 1.

Table 6.19: Awareness RCASN Degree Centrality for iteration 1 and 2

Awareness types.	OutDegree			InDegree		
Iteration 1	**Mem**	**High deg. Members**	**Deg.**	**Mem**	**High deg. Members**	**Deg.**
Availability	12	PM,PO	11	12	PM,PO,SD2,SD3,QE2	7
General	12	PM	11	12	PM,SD1	6
Current	10	PM,PO	7	10	SD3,SD1,QE1	6
Work Status	10	PM,PO	7	10	SD1,SD3,SD5	6
Iteration 2						
Availability	12	PM, PO	11	12	SD1,SD2,SD5	8
General	12	PM	11	12	SD1, SD3	8
Current	10	PM,PO	7	10	SD1, SD2, SD3	7
Work Status	10	PM, PO	8	10	SD1,SD2,SD3,SD4,QE2	7

6.4.3.2.RCASN Betweenness

To investigate the information flow from the perspective of each node, betweenness measure was the most suitable. It helps to identify the centrality of a node in a network. The importance or power of node can be analysed using betweenness. It is the smallest path required to reach a particular node from all the vertices to others. The results of betweenness calculated for communication and awareness are shown in Table 6.20 and 6.21 respectively. The betweenness values helped in determining which nodes played important and central part in imparting information in the network. It can be seen that in 38% of the networks in iteration 1 and 56% networks in iteration 2 PM was the node with maximum betweenness value. For the rest of networks developers and quality engineers remained the important nodes. The centralization values for all of the communication reasons range from 0 to 33.86. In iteration 1, PM has high betweenness value for user story negotiation, 37 and for clarification it is 33, yet the centralization of user story clarification (33.86) is higher than user story negotiation (15.28). This shows that the network with low centralization has less power in it. The betweenness values showed that PM has the maximum betweenness index for user story clarification and negotiation in both of the iterations among the rest of communication reasons. This shows that most of the team members collaborated with and through PM regarding user stories. Therefore, it can be concluded that PM was the powerful role in this team and was responsible for imparting most of the information to the rest of team members. However, for code related communication developers were powerful actors.

Table 6.20: Communication RCSN's betweenness value for iteration 1 and 2

Comm. Reasons	**Iteration 1**			**Iteration 2**		
	Mem	**High Bet. mem**	**Bet. Value**	**Mem**	**High Bet. mem**	**Bet. Value**
Bugs	10	QE2	8	11	QE2	8
Changes	10	PM	18.83	10	PM	17
Code issues	10	SD2	3	10	SD4	6
Refactoring	10	0	0	10	SD4	3.5
Code reviews	10	SD2	9	10	SD2	5
Code sync.	10	SD2	3	10	SD4	1
Coordination	10	SD3	11	10	PM	7
Management	11	PM	29	11	PM	34
Quality	10	QE1	4	10	QE1	12
Sprint planning	11	SD3	1	10	PM	19
Support	11	PM	9	11	PM	14
US clarification	11	PM	33.06	11	PM	33.7
US negotiation	11	PM	37	11	PM	28

Table 6.21: Awareness RCSN's Betweenness value for iteration 1 and 2

Awareness type.	**Iteration 1**			**Iteration 2**		
	Mem	**High Bet. mem**	**Bet.Value**	**Mem**	**High Bet. mem**	**Bet.Value**
Availability	12	PO, PM	9.15	12	PM, PO	9.23
General	12	PM	20.03	12	PM	21.93
Current	10	SD3	1.667	10	SD2	3.75
Work Status	10	SD3	4.66	10	SD3	2.58

For awareness networks, PM and PO were the most aware actors for availability awareness and PM for general awareness. However, for task- current awareness and work status awareness developers were the powerful actors in both of the iterations. The network centralization values range from 1.667 to 20.3 in iteration 1 and 2.58 to 21.93 in iteration 2. This shows less centralized awareness networks.

Therefore, the discussion above can be related to the findings of RCASN centralization in which it was described that PM proved to be the prominent actors in networks with high centralization value. Moreover, it also confirms the previous centralization findings that most of the networks have decentralized information flow.

6.4.3.3. EigenVector Centrality

To investigate the influential and important nodes in the networks, EigenVector Centrality measure was used. This measure determines the influence of a node in terms of its connections with highly-connected nodes in the network. The concept behind is that highly-connected neighbours contribute more to a node's Eigenvector centrality value than less-connected nodes. The nodes with high Eigenvector values are considered influential or high profile people in the networks like leaders. Therefore, to identify such people who influenced the networks and played pivotal role in information flow. The eigenvector centrality analysis of communication networks is shown in table 6.22 (a) and (b).

The PM remained the influential and pivotal node in 46% (6 of 13) of the communication networks in iteration 1 and 54% (7 of 13) in iteration 2. This reiterates the fact that PM was the powerful and node of pivotal importance in information flow among the teams. Furthermore, developer SD2 also remained the powerful node in 38% (5 of 13) communication networks in iteration 1including bugs discussion, code issues, refactoring, code reviews and code synchronization. This shows that among rest of the 4 developers SD2 was pivotal and powerful.

Table 6.22: Eigevector Centrality Values for (a) Iteration 1 and (b) Iteration 2

Comm. Reasons	(a) It 1					(b) It 2				
	Mem	high value mem	Eg.	low value mem	Eg.	Mem	high value members	Eg.	low value members	Eg.
Bugs	10	SD2	0.4	UX-F	0.14	11	SD2	0.39	SP	0.07
Changes	10	PM	0.48	SD4	0.13				UX	0.13
Code issues	10	SD2	0.6	PM	0	10	PM	0.47	SD4	0.13
Code refactoring	10	SD1,SD2	0.6	PM,PO,QE2,UX,SD4,SD5	0	10	SD4	0.59	PM,PO	0
Code reviews	10	SD2	0.65	UX	0	10	SD1,SD3,SD4	0.48	QE2,UX,PM,PO	0
Code syn	10	SD2	0.6	SD1,SD5,UX	0	10	SD2	0.55	PM,PO,UX	0
Coordination	10	SD3	0.5	PO	0	10	SD4	0.48	UX,PM,PO,SD5	0
Management	11	PM	0.67	PO, SD2, SD3, SD5, UX, SP	0.19	10	PM	0.47	UX	0
Quality	10	QE2	0.54	SD5	0	11	PM	0.67	SD4	0
Sprint planning	10	PM	0.5	QE1,QE2,UX	0	10	QE1	0.56	UX	0
Support	11	PM	0.54	QE1,QE2,UX	0	10	PM	0.55	UX	0
U.S clarification	11	PM	0.45	QE2	0.18	11	PM	0.52	UX	0
				Customer	0.11	11	PM	0.43	UX	0
U.S negotiation	11	PM	0.54	SD4, UX	0.11	11	PM	0.42	UX	0

For coordination and quality in iteration 1 SD3 and QE2 remained the powerful nodes among the team members. Unlike iteration 1, QE1 became the most powerful node for quality related communication network in iteration 2. The absence of QE2 in later stages of project became the reason of this take over by QE1. UX remained the least influential member in 81% (8 of 13) and 86% (11 of 13) communication networks in iteration 1 and 2, respectively. In addition, PM and PO also remained insignificant in code related discussion network i.e. code issues, refactoring, code reviews and code synchronization in iteration 1. The emergent members i.e. customer and support team also were insignificant because they did not communicate with any of the team member. Also, emergent members were not assigned any task so they played insignificant rile in information flow in both of the iterations. In iteration 2, SD4 took the place of SD2 and became the most powerful node after PM for issues, refactoring, and code synchronization.

Therefore, the above analysis shows that PM remained the influential members for majority of the networks in both of the iterations except the code related discussion in which SD2 in iteration 1 and SD4 in iteration 2 remained powerful actors. However, due to the irregular participation of QE2 in later iterations, QE1 remained the most powerful node in quality discussions in iteration 2. UX remained the most insignificant member in both of the iterations.

6.4.3.4. RCASN Reachability

To deeply analyse the information flow patterns among team members, another measure ‘reachability’. It is used to identify any existing path between two nodes no matter how many other nodes fall in between them. The reachability matrix obtained for each network is shown in Appendix C, Figure C.16 and C.17. A sample of connected members reachability matrix and a one with isolated members is shown in Figure 6.5 (a) and (b) respectively. Figure 6.5 (a) shows user story clarification matrix in which actors have connected paths to others except UX. It can be seen that the emergent member ‘Customer’ is accessible through alternate paths by many nodes. In Figure 6.6 (b) the code synchronization matrix is shown in which isolated members like UX, PM, PO, SD5 can be seen. All the reachability patterns on visual examination showed that UX, SD5, QE2 remained isolated nodes in maximum networks. However, PM and PO were also isolated in code related communication networks. UX being a shared member had not participated well in team activities. Likewise, SE5 was a new member inducted in team from Finland and thus most of the members did not communicated with him that frequently as compared to other members they had previous worked experience. QE2 was found isolated in many networks due to her less participation in team activities caused by her unexpected medical leave in between the project often.

In awareness network all the members were reachable by others through alternate paths for availability and general awareness. This shows that most of the members were generally aware of each other. However, a close look at patterns clarifies that SE5 had least awareness of other team members confirming the reasons of him being isolated din communication networks. In current and work status awareness PM and PO were left isolated by the team members as they were inapplicable for task assignment. Thus no member of the team knew about the task allocated and work status of PM and PO.

		1 P	2 P	3 S	4 S	5 S	6 S	7 S	8 Q	9 Q	10 U	11 C
1	PM-F	0	1	1	1	1	1	1	1	1	0	1
2	PO-F	0	0	0	0	0	0	0	0	0	0	1
3	SD1-MY	1	1	0	1	1	1	1	1	1	0	1
4	SD2-MY	1	1	1	0	1	1	1	1	1	0	1
5	SD3-MY	1	1	1	1	0	1	1	1	1	0	1
6	SD4-MY	0	0	0	0	0	0	0	0	0	0	0
7	SD5-F	1	1	1	1	1	1	0	1	1	0	1
8	QE1-MY	1	1	1	1	1	1	1	0	1	0	1
9	QE2-MY	0	0	0	0	0	0	0	0	0	0	0
10	UX-F	1	1	1	1	1	1	1	1	1	0	1
11	customer	0	0	0	0	0	0	0	0	0	0	0

(a) User story clarification communication

		1 P	2 P	3 S	4 S	5 S	6 S	7 S	8 Q	9 Q	10 U
1	PM-F	0	0	0	0	0	0	0	0	1	0
2	PO-F	0	0	0	1	1	1	0	1	1	0
3	SD1-MY	0	0	0	0	0	0	0	0	0	0
4	SD2-MY	0	0	0	0	0	1	0	1	1	0
5	SD3-MY	0	0	0	0	0	0	0	0	0	0
6	SD4-MY	0	0	0	0	0	0	0	0	0	0
7	SD5-F	0	0	0	0	0	0	0	0	0	0
8	QE1-MY	0	0	0	0	0	0	0	0	1	0
9	QE2-MY	0	0	0	0	0	0	0	0	0	0
10	UX-F	0	0	0	0	0	0	0	0	0	0

(b) Code Synchronization communication

Figure 6.5: Reachability matrices for (a) fully connected and (b) isolated members' network

6.3.4.5. RCASN clustering coefficients

The Clustering Coefficients is one of the important properties of a graph or network which determines the degree or extent of nodes to stay closer or connectivity of neighbouring nodes. In other words in context of friendship clustering coefficient determines that the measure of extent to which ones nodes friends are each other's friends as well (Watts & Strogatz, 1998). This is also called local clustering coefficient. The clustering coefficient is a real number ranging between 0 and 1. It is 0 when there is no clustering and 1 when there is maximum clustering. Maximal value of clustering coefficients confirms the presence of cliques in the networks. Moreover, high clustering coefficient values also confirm "small work structures" in the networks. The small world network is defined if nodes are not neighbours to each other but can be connected through a small number of hops. The networks have small worldliness if two nodes connect to each other through the smallest path and have high clustering coefficient values (Watts & Strogatz, 1998). The clustering coefficient values calculated for communication and awareness networks are tabulated below in Table 6.23 (a) and (b).

Table 6.23: Communication and awareness RCSN's clustering coefficients for iteration 1 and 2

Comm. Reasons	Case 1	
	Iteration 1	**Iteration 2**
Bugs	0.26	0.26
Changes	0.42	0.30
Code issues	0.00	0.42
Code refact.	0.00	0.56
Code reviews	0.34	0.42
Code sync.	0.19	0.27
Coordination	0.41	0.41
Management	0.50	0.59
Quality	0.67	0.35
Sprint planning	0.39	0.27
Support	0.25	0.38
User Story clarification	0.51	0.63
User Story negotiation	0.50	0.71
Mean	0.34	0.43

Awarenes s types.	Case 1	
	Iteration 1	**Iteration 2**
Availabilit -y	0.79	0.78
General	0.68	0.71
Current	0.49	0.51
Work Status	0.49	0.58
Mean	0.61	0.65

The values ranging above 0.60 will be considered high and close to 1 whereas the rest were considered low and close to 0. In iteration 1, the clustering coefficient values are relatively low and close to 0 for almost all of the communication reasons. For code issues and refactoring the clustering coefficient value is 0 which shows no cliques at all. The communication networks in iteration 1 show least clustering and absence of cliques and the low clustering coefficient values confirm it. The notion was already proved in section 6.3.4 that in iteration 1 there were cliques only in quality related communication network. In iteration 2, all of the networks had low clustering coefficient values except user story clarification (0.63) and negotiation (0.71). Therefore, it can be concluded that overall there was least or no clustering in communication networks for case 1, except for user story clarification and negotiation networks in iteration 2. Furthermore, it also drives the conclusion that communication networks do not exhibit small world structure due to low clustering coefficient values.

The clustering coefficient values calculated for awareness networks were contrary to communication networks. They showed high values of clustering coefficients confirming more cliques. The values indicated that availability (0.79) and general awareness (0.68) networks had cliques and exhibited small world structures. The values of both of the networks increased in iteration 2 which shows an increase in connectedness of neighbouring nodes. The low coefficient values range from 0.49 to 0.58 in both of the iterations. These values are high as compared to the low coefficient values for communication networks. Therefore, it can be concluded that number of cliques in awareness networks increased with time awareness which means that team members gained mutual acquaintance of each other.

6.4.3.5. Cut Points

To further the knowledge on information flow patterns among agile teams another measure 'cutpoints' is used. Cut point determines the actors or actors (called cut sets) in the networks who if removed can cause information loss due to loose connections with other nodes. So, it was important to identify such important nodes that if removed can cause information lapse in the networks. The cut points and cut sets identified in communication networks are shown below in Table. 6.24 (a). The table shows that in iteration 1 the number of cut point or cut sets was more than iteration 2. This shows that more members communicated with each other directly in iteration two. The growth of awareness and rapport increased the direct communication among members. Moreover, PM and SD2 were the cut points which if removed would results in communication lapse in most of the communication networks, in both of the iterations. SD2 was the senior members with experience level 5 to 7 years working at company Alpha. He was the focal point of most of the discussions regarding code and otherwise along with PM. This way the managers can see the focal team members and foresee the information loss and communication lapse in case of their absence or resign.

6.4.3.6.Component

To further the subset analysis and to investigate of information flow patterns among agile teams another measure 'component' was used to check if the networks were all connected. It was aimed to identify the information lapse in case of isolated members and broken patterns. The component measure is defined as the maximal set of nodes connected to each other through some path (of any length). The component investigation of communication and awareness networks are presented in Table 6.24 (b) below.

Table 6.24: Communication RCASNs (a) Cut points, (b) Weak and Strong Components for iteration 1 and 2

(a) Cut points				
Comm Reasons.	**Iteration1**		**Iteration 2**	
	No	**Actor**	**No**	**Actor**
Bugs	0		1	PM
Changes	0		0	
Code issues	2	SD2, SD4	2	SD1,SD4
Code refact	3	SD1,SD2,SD5	0	
Code reviews	2	SD2,QE2	1	SD2
Code syn	3	PO, SD2, QE2	2	SD2,SD4
Coordination	1	SD2	0	
Management	1	PM	1	PM
Quality	3	SD2,QE1,QE2	2	SD2,QE1
Sprint planning	0		1	PM
Support	1	PM	1	PM
US clarification	0		0	
US negotiation	1	PM	0	

(b) Weak Component		**Strong Component**	
It 1	**It 2**	**It 1**	**It 2**
1	1	7	8
1	1	7	7
4	4	9	8
5	6	10	6
2	4	10	8
4	5	10	10
2	2	8	8
1	2	9	7
5	2	11	7
4	2	11	7
4	2	10	11
1	2	4	3
1	2	6	3

The number of weak components are lesser than strong components for both of the iterations in communication networks. The number of networks having 1 component is only 4 in iteration 1 and 2 in iteration 2. This shows that there was no isolated node in communication changes, management and user story negotiation and clarification. However, in iteration 2 isolated nodes appeared in management, user story clarification and negotiation. The component details showed that in most of the networks isolated component was UX. The strong components are formed through direct paths between nodes. Therefore it can be seen that number of components in strong component measure are larger than weaker ones (which do not consider directed path). The values of strong components are large in communication networks in which not all the members participated. The large number of isolated nodes or isolated groups gave rise to large strong component values.

6.4.4. Summary of Case 1

Summarizing the above discussion, it can be concluded that in case 1 the emergent members involved were support team and customer. The involvement of emergent members was limited to PM and PO only. The emergent members were communicated for communication reasons like bugs discussion, support issues, user story clarification and negotiation. The PM was the foci of most of the communication happened between the team. Most of the team members communicated with and through PM. However, in code related communication developers remained the most active members. The teams were over all aware of each other as there were no isolated nodes in availability and general awareness networks. But the emergent members were known to PM and PO only. UX remained the least active team members and participated very less in overall team communication. The involved of UX with more than one project at a time was the reason for his less participation. QE2 was the second inactive member after UX in iteration 2, due to her medical leave. The awareness of teams increased in iteration 2 whereas for iteration 1, whereas the communication remained almost the same in both of the iterations. However, the involvement of newly included remote site member SD5 was increased in iteration 2. The information flow among the team did not follow the small world structure and there were very less or no clusters or cliques in both iterations. The communication networks were overall decentralized with information distributed to all the members. This shows that team followed agile methods traits that encourage decentralized teams with information vested to all the members.

6.5. Results and discussions

In this section, the results of the applied SNA measures are mapped with the proposed research questions (mentioned in Chapter 1).

6.5.1. The characteristics of the communication and awareness networks in agile teams

(RQ4) What are the characteristics of the requirements-centric communication and awareness patterns among agile teams? This question was answered by the empirical evaluation of the communication and awareness networks, RCASNs, of the agile teams. The social networks' measures were run for four projects and the results were used to analyze the characteristics of the communication and awareness networks of agile teams.

6.5.1.1. Agile teams have decentralized communication and somewhat centralized awareness networks

The SRQ6 question: **Do agile teams have centralized communication and awareness networks structures?** was answered by investigating the in-degree and out-degree centralization of the networks. The results showed that only a small percentage of the communication networks (15% to 23%) have centralization value close to 1 (=>0.60). The analysis reveals that agile teams follow a decentralized communication structure. Moreover, no particular increase or decrease of pattern was observed in the networks for both iterations. This proves that the centralization of communication networks did not increase or decrease with the growth of the project. The communication networks with slightly higher centralization values were support, management and quality for almost all cases. It can be seen that for support and management issues, the teams communicated with the managerial members, support staff and management teams respectively and had less interaction with themselves on these issues. This makes the aforementioned communication networks centralized around several members (nodes).

Unlike communication networks, awareness networks showed a high percentage of centralized networks. Almost 50% of the networks were centralized. Availability and general awareness networks showed high centralization values compared to current and work status awareness networks. This pattern remained the same for all cases and in both iterations. The reason behind the highly centralized availability and general awareness networks is the presence of emergent members within those networks. As the rest of the team members have limited or no connection with the emergent members, this contributed to a minimal or no awareness. Moreover, the managerial and senior members were more aware of the rest of the team's availability and general traits. This made the two awareness networks, i.e. general and availability, highly centralized. However, current awareness and work status awareness were highly decentralized for all cases and in both iterations. This shows that an agile teams' awareness has comparatively centralized awareness networks for availability and general awareness.

However, current and work status awareness networks follow the decentralized structure.

6.5.1.2. Communication reciprocity among agile teams

The SRQ7 question: **Do virtual agile team members communicate in an equal manner with each other?** mentioned in Chapter 1 was answered by finding the mutual communication reciprocity of the agile teams for all of the four cases. Although agile teams have rigorous communication among members when collocated, we do not know how communication works when the teams are located in different continents. We applied the ties reciprocity measure to assess the reciprocal communication among distributed members. The communication ties reciprocity among virtual teams is important to assess the balance in communication. A balanced communication is an indication that information has been exchanged and understood as well as responded. Therefore, we applied the ties reciprocity measure (Wasserman & Faust, 2009) to calculate the level of mutual communication among team members who are communicating virtually with each other. It was observed that the communication reciprocity increased with time. The number of communication networks with communication reciprocity index higher than 60% increased in iteration 2. In iteration 1, the communication reciprocity index was low for most of the communication networks except in a few networks like code issues, code reviews, code refactoring and code synchronization (case 2), management (case 3) and bugs discussion (case 4). However, in iteration 2, the mutual communication index was high for user story negotiation (case 1), changes communication, code issues, code reviews, code synchronization, code refactoring, coordination, sprint planning, user story clarification and negotiation (case 2), bugs and management (case 3), and bugs, code refactoring, code reviews, sprint planning and support (case 4). There was an obvious increase in the mutual communication of team members in iteration 2. It was observed that code related communication and bugs discussion were among the reasons for which team members communicated the most with each other. The reciprocity between the dyads (Dyadic reciprocity) was lesser than the percentage of all possible ties that are a part of the reciprocated relations (Arc-based reciprocity). This shows that mutual communication between pairs was less than the number of ties mutually connected to each other. No certain patterns were observed in all of the cases except for the code based communication networks that have a high reciprocity index in both of the iteration in case 2. Case 2 has tremendously grown in the number of its high indexed mutually reciprocated communication networks. More fine grain analysis of the networks was performed by visual inspection of the sociograms in order to deeply analyze the nature of these mutual communication links. It was observed that in several cases, mutual communication was not intended nor required; rather a one way information transmission was the purpose. For instance, in the case of some communication reasons reported to the PM for his/her own record but it was not required for the PM to respond to those notifications or information. This way, the reciprocity of communication decreased. Therefore, it can be concluded that low reciprocity indexes do not contribute to less communication; rather it justifies the right amount of communication in certain situations.

The awareness ties reciprocity values are comparatively larger than the communication values which shows that the team members were mutually more aware of each other than the mutual communication that occurred between them. The same pattern was seen in all of the four cases. Moreover, the dyadic reciprocity index of awareness networks was lower than the arc based reciprocity index. This trend remained the same for both communication and awareness networks.

6.5.1.3. Core members of Agile Teams work closely with each other in spite of being at distributed locations

Considering the fact that agile team members work closely with each other, an investigation was conducted to determine whether there were any members who worked more closely than others forming a core-periphery network structure and also to answer the SRQ8 question: **Do agile teams have members working more closely together than to others?** The core-periphery measure (Wasserman & Faust, 2009) was applied on the collaboration data retrieved from all of the four cases. The analysis results showed that 46% of the networks in case 1 (both iterations), 84% (in iteration 1) and 61% (in iteration 2) in case 2, and 31% (both iterations) in case 3 and case 4 showed high core-periphery network structures. This shows that the percentage of centralized network structures was more in case 2 as compared to the other cases. A more in depth analysis of the results revealed that the Project Manager and other customized managerial roles such as Product owner (case 1), Technical Architect (case 2) and Team leader (case 2), along with some senior Developers and Quality Engineers, were among the core members of the actual communication networks (i.e. management, coordination of activities, sprint planning, user story clarification, and user story negotiation) closely communicating with each other and with emergent members (i.e. customers, executives, experts, management and support team). For code-related communication networks, i.e. resolving code issues, code refactoring, code synchronization and code reviews, only Developers and Quality Engineers communicated closely with each other. The loosely connected or periphery members were those who participated less in the communication, for example user experience designer and quality engineer1 (QE1) in case 1. The User experience designer was not a dedicated team member and was associated with three projects at a time, thus he did not participate in daily scrums and meetings regularly. The QE1 took medical leave after the first iteration was completed and then remained irregular or worked from home at times. Moreover, it was also noticed that the core members for various communication reasons comprised of members from distributed locations. This shows close collaboration of virtual agile teams in spite of being distributed across wide time zone differences. For instance, the developers communicated closely for all code related artefacts in case 1, in spite of being at distributed locations, i.e. Finland and Malaysia. Thus, the results have unveiled an interesting aspect in order to identify the core members with considerable information exchange in certain communication networks. This shows that agile teams communicate rigorously with each other but avoid unnecessary intimations. For instance, in code related discussions, only developers and quality engineers were involved and no intimations were sent to the project manager when a small code issue gets resolved. Likewise, for the management, support, coordination and sprint planning communication networks, the Project Manager, Technical Architect and Team Leader took core positions while the developers/quality engineers formed a periphery structure. Cataldo has also explained this trait of agile teams as they communicate with each other without getting overloaded because it is not the amount of communication that matters but the communication that are relevant to the task that matters (Cataldo & Ehrlich, 2011). Moreover, it also supports the literature's claim that team members who communicated the most also contributed the most on software tasks (Cataldo et al., 2006).

For the awareness networks, the analysis shows that emergent members were not supposed to be aware of the team members and vice versa. Therefore, the assigned members were mostly present in core groups while the emergent members in the periphery. That is why the availability and general awareness networks were more centralized with high core periphery index values as compared to the current and work status awareness network structures.

6.5.1.4. Lone members or groups in distributed agile teams

Based on the fact that agile teams work closely with each other, we expect that all members of a distributed agile team can be reached by others and this will answer the SRQ9 question: **Are there lone or isolated team members or groups of members in agile teams?** The aim was to identify the subset of a network in which members (nodes) are more closely tied to one another than with the rest of the members of the network. This subset is defined as a Clique (Hanneman & Riddle, 2005). The absence of connected group of members, i.e. cliques, can be seen for most of the communication reasons in iteration 1. The number of cliques that existed in the networks varied from 1 to 6. There was no clique found in iteration 1, case 1 for all of the communication reasons networks. However, cliques existed in iteration 2 for user story clarification (1) and user story negotiation (6). The absence of cliques in iteration 1 shows that there were no closely connected groups (with maximum three nodes). The absence of cliques does not indicate minimal or no communication among members in iteration 1. But it can be said that there was no set members in all of the networks who communicated closely with each other (with no null ties). Therefore, cliques were missing in iteration 1 but dyads were there which means that not more than two members closely communicated with each other. The study of cliques shows very interesting social behaviours of the team members, for instance the Project Manager remained the key member in most of the cliques along with core members such as developers and quality engineers for user story negotiation. There were no cliques seen in all of the cases and iterations for management, sprint planning and support communication networks. This is because of the general trend that the team members tend to communicate with the managerial members, support staff, or management teams for these reasons rather than communicating closely with each other on these issues. The discussion leads to a conclusion that most of the cliques are organized around the managerial or organizational members of the team and are motivated by the developers. Moreover, cross-site communication can also be seen among members communicating closely with each other.

6.5.1.5.Agile teams have less tendency to cluster together

In answering the SRQ10 question: **Do agile teams tend to group together with the increase in communication?** the correlation between the degrees of nodes, with which they tend to cluster together, with their overall in-out degree reveals the behavior of nodes towards grouping. It is observed in the literature that low degree nodes tend to be a part of lesser groups whereas it is the inverse in the case of high degree nodes (Mislove, Marcon, Gummadi, Druschel, & Bhattacharjee, 2007). More social network measures were added to the framework after conducting the case studies and performing data analysis. The measures were selected on the merit of providing accurate answers to the research questions. The Pearson correlation was calculated between the clustering coefficient values and the in-out degree of the networks. The spearman correlation was chosen because the data was non-linear. The data was first checked by applying Pearson correlation and spearman correlation. The correlation value achieved by the spearman correlation operation was greater than the one obtained through Pearson correlation. If the spearman correlation value is greater than the Pearson correlation value, this shows that the data is monotonic and not linear.

It can be observed that an overall negative correlation existed between the clustering coefficient values of communication networks and their respective in-out degrees (shown in Appendix C, Table C.67). The results showed a positive correlation value only for case 1 in both iterations and case 4 in iteration 2 as shown in Figure 6.6. However, for the rest of the cases, the correlation results revealed that clustering coefficient and in-out degree acted contrary to each other. This finding supports the literature's claims that clustering coefficient is negatively related to the degree of nodes (Bloznelis, 2013). A similar pattern was observed in almost all of the cases except case 1. This shows that with an increase in the degree, the clustering coefficients decrease. The more links team members emit, the more they collaborate in a decentralized manner rather than sticking to only a few members as in almost all of the cases. This result supported the findings of the literature study in which the degree was compared with clustering for checking teams cooperation, where the cooperation decreases when the degree increases (Li, Mlinuigalwayie, & Riordan, 2013). Therefore, it can be concluded that agile teams have a low tendency towards grouping with the increase in their communication behavior.

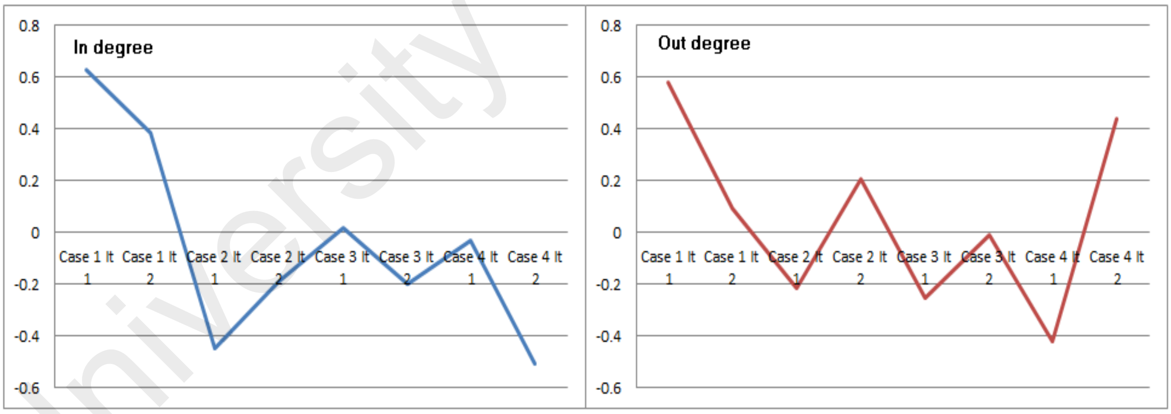


Figure 6.6: Correlation plot between clustering coefficient and in-out degree

6.5.2. Information flow patterns among agile teams

The RQ5: **What information flow patterns do agile teams follow?** was answered by applying social network analysis measures on the empirical data collected for four projects. The results were used to guide the information flow patterns of distributed agile teams. Unlike traditional software development teams, agile cross-functional teams are closely knit and highly interactive. Communication is constant and free of imposed organizational barriers. Therefore, interesting questions about agile teams information seeking patterns emerged, which are explained in this section.

6.5.2.1.Agile teams have several members playing central and influential part in information flow

Answering the SRQ11: **Do agile teams have some actors playing central part in information flow?** degree centrality, betweenness (Wasserman & Faust, 2009) and Eigenvector centrality were applied to the actual communication networks. The central position of a members in the network makes them 'prominent' (Hanneman & Riddle, 2005) and influential. The empirical results showed that the same pattern in almost all of the cases that for code related communication developers stood out. However, for the rest PM was the member who initiated and received the most of communication links. When the high in and out degree members were analyzed on the basis of their work experience it was found that only those developers and quality engineers remained the central or predominant nodes in the networks who had more work experience than the rest. For instance in Case 1 SD2, in case 2 SD3 (team lead), in Case 3 and 4 Quality engineer along with the developers and PM were the central members. The members who received or emitted larger number of communication links are considered central in the networks and believed to carry a lot of social capital (e.g. data, information). Not only that member is transmitting information but also other members are connecting to her with the same intensity. So such central members are considered powerful or hub nodes of the networks. Therefore, the findings revealed that in agile teams at some places Project Manager and other managerial members like Team leader were the central members and at other places senior developers acted as the central persons of the networks and controlled the interactions of other members. This finding suggests that the members that played central position were the more experienced ones in the team and held pivotal positions in the hierarchy, and therefore had more information about the product under development. However, it was noticed that the central members for quality related communication network were quality engineers and for code related communication networks like code review, code refactoring, code issues and code synchronization the core members were developers. This shows that regardless of the location, members contribute according to their roles and responsibilities vested to them. Therefore, being the central member in agile team is based on the member's work experience with the team which determines the amount of information he or she has. The trend of central members did not change much in both of the iterations.

6.5.2.2. Agile teams have members that may break down the information exchange channel if go missing

In addition to identify the members who are key in controlling information flow, the members whose absence can disrupt information flow were also identified while answering the SRQ12 **Do agile teams have such member(s) who play pivotal part in information exchange?**

To find these members cutpoint measure (Wasserman & Faust, 2009) was applied. The results revealed that several communication networks had cutsets comprising members from distributed locations for instance code refactoring communication network in case 1 had cutset comprised of three developers, 2 from Malaysia and 1 from US. This unveils a very interesting finding that agile team members while being working at distributed locations communicated closely with each other. The communication networks comprised of cut sets were mainly code issues, code synchronization, code reviews, code refactoring, coordination and quality related communication in case 1, 3 and 4. However, case 2 showed very less number of cut points in both of the iterations. The cut points found were comprised of PM and SD3 and for bugs, management support and user story negotiation. The results of centrality and Eigen vector centrality has also proved that PM and SD3 who also acted as team lead and was a senior developer, were the central team members. The low number of cut points or cut sets in case 2 shows that the information was distributed among team members equally and there were just a few members only with information in their hand and whose absence can cause information flow break down.

6.5.3. Impact Analysis of RDC in Agile teams

Based on the detailed empirical analysis performed on the data collected from four projects following the agile method for software development, the RQ6 question: **How do communication and awareness networks impact the team performance in agile teams?** is answered in this section.

6.5.3.1. The interplay between distance, communication, and awareness

The relationship between distance, communication and awareness was analyzed using a quadratic assignment procedure (QAP) (Hubert & Schultz, 1976). Traditional parametric correlation methods could not be used to measure independent network data. QAP is an unbiased technique used to analyze dyadic social network based data to answer questions on observed relationships (Krackhardt, 1987). Multiple correlation operations were performed on coded awareness, communication and distance matrices. The data was coded in such a format that '1' is placed between the members where communication or awareness existed and '2' is placed where the communication or awareness was absent. The absence of relationship is depicted with ‘2’ in place of ‘0’ because QAP considers ‘0’ as an empty slot. Therefore, all of the edge tables prepared for the communication reasons and awareness type of networks for all of the four cases were re-coded as per the QAP acceptable format. For the distance factor, pairs of members were coded as ‘collocated’ if they were both working in the same country, and as ‘remote’ if they were working in different countries. For the other factors, the dichotomized values from the questionnaire were as follows: Communication frequency as ‘Less than 3 times a day’’3 or more times a day’, and each type of awareness as ‘Aware’/’Not aware’ (including N/A option). These relationships were calculated to observe the impact of distance on communication between the distributed team members and their awareness of each other. In addition, the interrelationship between communication and awareness was also investigated using QAP operations.

Table 6.25 provides the results of the QAP correlation tests for each project and per iteration. It can be observed that there is no consistent pattern across all four projects; however, Case 3 indicates an influence of distance in communication and all types of awareness, except Availability.

For Iteration 1, for Case 3 and Case 4, we found a significant decline of Communication frequency over distance ($r=0.432$, $r=0.315$ and $p<0.05$) and General awareness over distance ($r=0.613$, $r=0.336$ and $p<0.05$), meaning that the communication with remote colleagues was less frequent than with the local ones, and that team members were less familiar with the professional background of their remote colleagues than with the local ones respectively. There was also a significant decline of Current awareness for Case 1 and Case 3 ($r=0.233$, $r=0.476$ and $p<0.05$) over distance, indicating that team members were less aware of the set of tasks the remote colleagues are working on than the ones that the local colleagues were assigned to. Work status awareness had a significant decline over distance (r=0.476, p<0.05) for Case 3 only. This indicates that, for this case, team members were less aware of the current work progress of remote members than of the local ones.

Table 6.25 shows a similar trend for Iteration 2, except that Communication frequency declined over distance for Case 1 and Case 3, and General awareness declined for Case 3 only. In addition, Availability had a significant decline over distance for Case 4 (r=0.289, p<0.05), meaning that remote team members were more difficult to reach than the local ones.

It is found that over 40% of the total communication ties reported for both iterations (46% in Iteration 1 and 44% in Iteration 2) are cross-sites. Similarly, about 40% of awareness ties (39% in Iteration 1 and 43% in Iteration 2) are also cross-sites.

Table 6.25: Relationship between distance, communication and awareness

It -1 Variable	**Case 1**	**Case 2**	**Case 3**	**Case 4**
	beta	**beta**	**beta**	**Beta**
Comm. frequency	0.1375	0.3118	0.4318*	0.3153*
General	0		0.6128*	0.3366*
Task	0.2227*	-0.0306	0.4764*	0.111
Work status	0.0214	0.0891	0.4764*	0.138
Availability	-0.1361		0.0826	0.2222
* p<0.05	n=10	n=5	n=7	n=9

It-2 Variable	**Case 1**	**Case 2**	**Case 3**	**Case 4**
	beta	**beta**	**beta**	**Beta**
Comm. frequency	0.2536*	0.4286	0.5731*	0.1638
General	0		0.6128*	0.1826
Task	0.2227*	0.0229	0.3838*	0.0561
Work status	0.0214	-0.0476	0.3838*	-0.1183
Availability	-0.1361		0.1638	0.2892*
* p<0.05	n=10	n=5	n=7	n=9

These findings suggest that distance seem to not matter for agile distributed teams, corroborating the literature findings on the topic (e.g., (Hossain, Babar, & Verner, 2009) (Holmstrom, Fitzgerald, Agerfalk, & Conchuir, 2006)). This high presence of interactions and awareness of remote members might be explained by the teams' daily routines. All teams had daily stand up meetings to synchronize information and progress. Case 2 was an exception due to the large time difference (11hours) as compared to the others. Therefore, they used to send e-mails at the end of the working day and once a week meet after hours through Skype. Other team members would work with Skype open and chat with their remote colleagues whatever it was necessary simulating collocation. Any topic related to a user story could then be discussed ensuring its immediate progress. Although not conclusive, the correlation test of the distance factor over communication and awareness supports to a certain extent this finding. About 40% of the networks showed a significant influence of distance, corroborating Damian et al (2010) finding that distance is not an issue for development teams despite the software development approach they follow.

6.5.3.2. Relationship between communication and awareness

Additionally, the relationship between communication frequency and the types of awareness was analyzed. Table 6.26 shows the QAP correlation between the communication frequency factor and each of the four types of awareness. For Iteration 1, a significant decline of Availability ($r=0.235$, $r=0.444$ and $p<0.05$), Current awareness ($r=0.420$, 0.377 and $p<0.05$), and Work status awareness ($r=0.330$, $r=0.241$ and $p<0.05$) was observed for Case 1 and Case 4, when frequency of communication was lower. This indicates that people were more likely to communicate with someone who they perceived as easy to reach, they know which tasks the person is working on, and they know the current progress of work respectively. A significant decline of General awareness was observed when communication frequency was lower for Case 1, Case 3, and Case 4 ($r=0.308$, $r=0.363$, $r=0.377$ and $p<0.05$), indicating that people were more likely to communicate with those who they know can help with their work.

Iteration 2 yielded similar results. However, General awareness declined for Case 3 and 4 only, and Work status for Case 1 and Case 3.

Table 6.26: Relationship between communication frequency and types of awareness

It-1 Variable	Case 1	Case 2	Case 3	Case 4
	beta	**beta**	**beta**	**beta**
Professional expertise	0.3079**		0.3634**	0.3774*
Task	0.41968*	-0.0572	0.2035	0.3774**
Work status	0.3301**	-0.0417	0.2035	0.2406*
General	0.2346**		0.0258	0.4438*
* p<0.01 ** p<0.05	n=10	n=5	n=7	n=9

It-2 Variable	Case 1	Case 2	Case 3	Case 4
	beta	**beta**	**beta**	**beta**
Professional expertise	0.1956		0.5328*	0.281**
Task	0.4437*	-0.0229	0.3162	0.4584*
Work status	0.4007*	0.0476	0.3162**	0.2307
General	0.2241**		0.1104	0.3617*
* p<0.01 ** p<0.05	n=10	n=5	n=7	n=9

The QAP results for the correlation of communication frequency and awareness showed a trend of influence of one over another, corroborating previous findings that communication is still an important source of awareness (Damian et al., 2010) (Ehrlich & Chang, 2006) despite recent advances in project management tool support. Collocated agile teams use face to face communication and daily status meetings to constantly share what is going on in the project. The findings show that distributed agile teams also follow these practices despite the physical distance and potential communication barriers it imposes. Moreover, the finding also suggests that team members communicate more likely with those other members they are more aware of, partially supporting the earlier findings that team members communicate more with those they know could help (Damian, Marczak, & Kwan, 2007).

6.5.3.3. Agile teams communication patterns show small world network structures with time

To answer the sub research question **Do agile teams show small worldliness behavior?** the small world network structures of the teams were observed. In small-world networks, small and highly dense clusters of nodes are connected to a few other nodes. In the context of collaboration translated to communication between software development teams, this means that a team member is surrounded by dense networks of other team members, all communicating with each other. Small world network structures for communicating between agile teams were considered detrimental to the iteration performance in the context of iteration planning and execution by reducing the time spent on real work (Cataldo & Ehrlich, 2011). However, small world network structures were found well suited for interdependent development tasks which, in return ,enhances the iteration quality (Cataldo & Ehrlich, 2011). Small world network structures have (i) high clustering coefficients, and (ii) low average path lengths. Small world networks govern the behavior of individuals as a team by shaping the level of connectivity among them (Uzzi & Spiro, 2005). The small world structures help the information exchange not to travel through long paths but to hop from cluster to cluster (Milgram, 1967). The small worldliness was calculated for all of the four cases and both of the iterations by making use of the proposed method by Uzzi and Spiro (Uzzi & Spiro, 2005). The average path length was calculated for all of the communication reasons networks (PL). Then, the same number of random graphs were generated with the same number of nodes and density based on the concepts of Newman (Newman, 2000) and their respective path lengths were recorded using the concept coined by Uzzi and Spiro (Uzzi & Spiro, 2005). Likewise, the clustering coefficient was calculated for the actual and random communication reasons networks (CC). Then, the Path Length

Ratio (PLr) and Clustering Coefficient ratios were calculated separately for all of the networks, such as $PLr = PL\ (actual) \div PL\ (random)$ and $CCr = CC(actual) \div CC(random)$. Finally, the small worldliness ratio (Q) was calculated for all of the networks as: $CCr\ \div PLr$. The PLr, CCr and Q calculated for all of the cases and iterations are pictorially represented in Figure 6.7 and the tabulated results are shown in Appendix C, Table C.68. It can be seen that the networks having high values for small worldliness are user story negotiation (case 1, iteration 2), code issues and support (Case 2 iteration 1 and 2 respectively), user story clarification (case 3, iteration 1) and user story negotiations (case 4, both iterations).

The distribution of small worldliness ratio (Q) over a period of two iterations for all of the cases can be seen in Figure 6.8. The results revealed that the highest ratio of small worldliness was observed for case 2 as compared to the rest of the cases under discussion. Moreover, the results also unfolded the fact that the teams' communication patterns grew more into small-world structures with the growth of the project as small worldliness ratios raised higher for iteration 2 in all of the cases. However, the values for small worldliness remained low for the rest of the cases which shows that the teams did not show highly centralized structures.

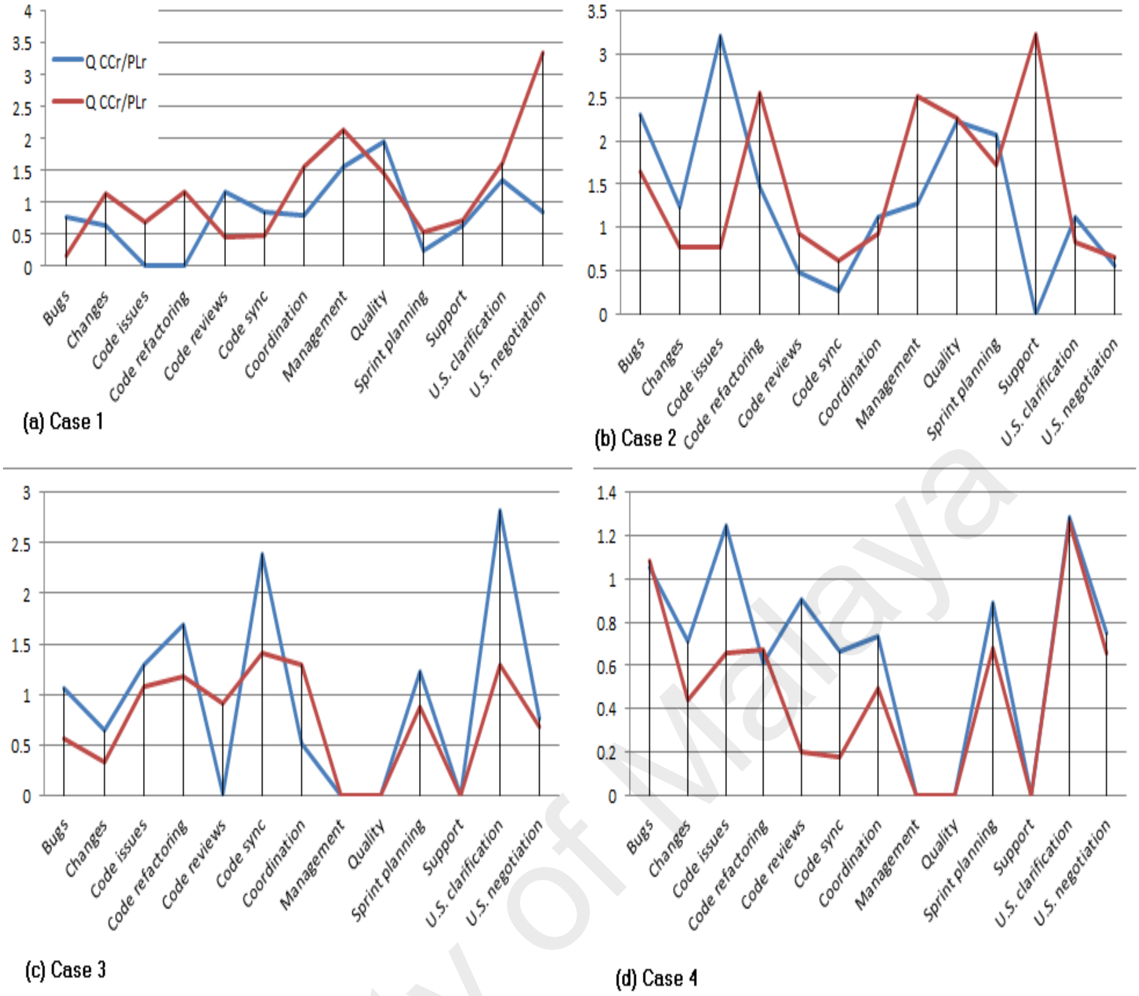


Figure 6.7: Small world network ratios (Q) computed for all cases and both iterations

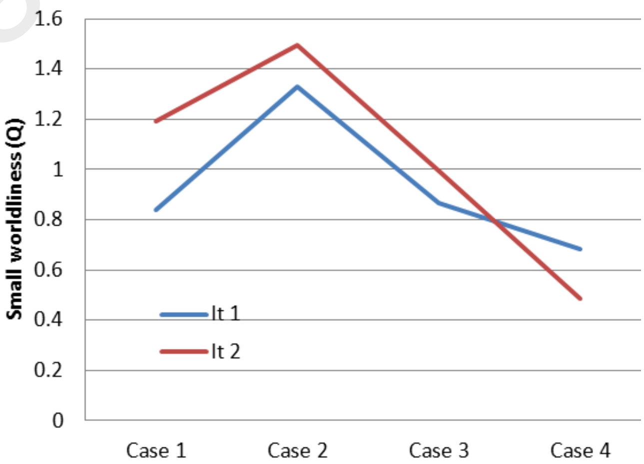


Figure 6.8: Distribution of small worldliness ratio (Q) for all cases and both iterations

6.5.4. Impact on agile team performance

Agile teams work in two weeks (approximately) iterations and try to complete the planned user stories within this time. Any rework or bugs are scheduled in the next iteration in such a way that does not disturb the ongoing flow of work. Each iteration is aimed at fulfilling several objectives, some of them are stated in the literature as functionality, schedule, quality and team satisfaction (Drury-Grogan, 2014). However, this is a generic categorization of some of the iteration objectives that are in line with the project management's success factors i.e. time, budget and quality. An agile team's performance closely depends on close-loop team communication (Dingsøyr & Lindsjørn, 2013) and mutual communication patterns (Cataldo & Ehrlich, 2011). Moreover, research on the effectiveness of team performance proves that it depends on the team coordination (Burke et al., 2006; Hoegl & Gemuenden, 2001). Therefore, the aim was to find the impact of the agile teams' collaboration i.e. their mutual communication and awareness of each other's general traits, availability, current task and work status, on iteration performance. For this purpose, the iteration performance was measured by keeping in view the team velocity and number of unfinished tasks which relates to the 'quality' and 'schedule' category. In practice, there are several metrics used by industry practitioners to measure an agile team's performance such as LOC (line of code, customer satisfaction, defect count, lead time, unfinished stories etc.

6.5.4.1. Iteration Performance

The iteration performance is calculated by keeping in view the number of tasks allocated, completed and not completed at the end of each iteration. A task is defined as a unit of work for each iteration. Each task is later broken down into further downstream artifacts such as specification writing, design work, coding, testing, and debugging. The agile teams allocate points to the tasks and use these arbitrary values to measure the effort required to complete a task or user story called story points. These points can be allocated in many ways based on the team's preferences. In most cases, project managers define a story point complexity range as a Fibonacci series (for example, 1, 2, 3, 5, 8). The story points produced per iteration is called the team's velocity. To determine the team's performance, several details were kept in consideration such as team size, maximum tasks allocated per person and iteration size. The estimated performance measures obtained for each iteration in all of the cases is shown in Table 6.24. It can be seen (in Figure 6.9) that the number of tasks remained incomplete, ranging from 1-3 in all of the cases. Moreover, the actual and desired velocity per iteration were closer to each other (as shown in Table 6.27) which means that the teams have managed to achieve a reasonable percentage of committed work. The percentage velocity of all the teams in both iterations is shown in Figure 6.9. The line graph representation shows that on average, the teams managed to achieve 79% of their desired work. However, the peaks showed that an increase was noticed in the velocity of all the teams in the second iteration as compared to the first one. An increase in the small world behavior of the teams was also noticed in iteration 2 (as shown in Figure 6.9) in all of the cases. Therefore, this findings corroborates the literature's claim that the highest level of project engagement was noticed at the start and end of the project, as well as the team collectiveness levels which also increased with the progress of the project (Licorish & MacDonell, 2013).

Table 6.27: Performance metrics

Performance Measures.	**Case1**		**Case2**		**Case3**		**Case 4**	
	It 1	**It2**	**It 1**	**It 2**	**It 1**	**It 2**	**It 1**	**It 2**
No. of task allocated per iteration	8	10	11	8	7	5	6	8
No of tasks completed	6	7	9	6	5	4	5	6
No of tasks not completed	2	3	2	2	2	1	1	2
Total Story points allocated per iteration	57	43	33	54	61	45	39	47
Story Point per iteration	43	33	23	48	49	37	31	39

(Velocity)								
Percentage velocity per iteration (%)	75	76	70	88	80	82	79	82

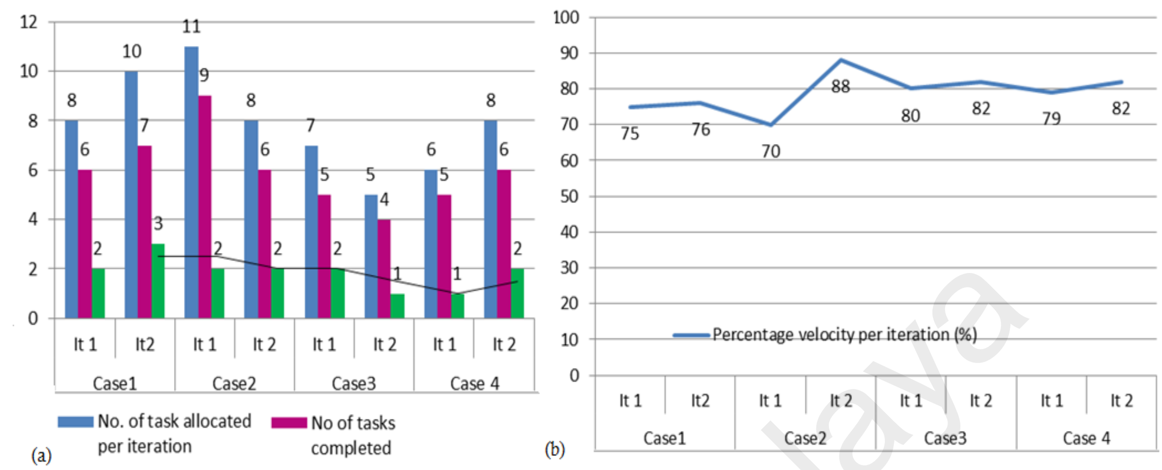


Figure 6.9: (a) Schedule report, (b) Percentage velocity values per iteration

Based on the above discussion regarding agile teams' networks properties and the information flow management among them, a regression based iteration performance estimation model was devised, as shown in Figure 6.10. A regression is a continuation of correlation and is used to estimate or predict the behavior of one variable with regards to the other. A multiple regression method was used in order to predict the iteration performance through several other observed variables as shown in Table 6.28. The iteration performance is a 'dependent' variable which needs to be predicted while the rest are 'independent' variables on the basis of which the performance is to be estimated. Dummy variables were considered in order to observe their effects on the team's iteration performance. The unobserved factors, when taken into account for a regression model, are translated as non-parametric dummy variables and are considered on the basis of their presence and absence (0 or 1) (Leech, 2005). The dummy variable, in this case, can be culture, language and time zone differences which were not deliberately recorded but believed to have some effect on the overall performance. The team related variables were team size, average experience of team members, and average number of user stories per iteration (work load). The user story related variables included its size and the average story points allocated to each story. The network characteristic properties included small worldliness, in-degree centralization and out-degree centralization. So, all of these variables were used to estimate the impact on iteration performance. Before moving towards regression analysis, a variance inflation factors (VIF) test was conducted for all of the considered variables to find how collinear they are with each other. The VIF results revealed that several independent variables were highly collinear with each other (having high correlations), which is considered undesirable in predicting the solution. Therefore, such variables were removed from the model in order to obtain accurate results. The model presented in Figure 6.10 shows the independent variables included after VIF calculation.

On the basis of the above discussion regarding dependent variables and their relationship with the independent variables, two hypotheses were formulated, where H_0 is the null hypothesis and H_A is the actual hypothesis.

Table 6.28: Variables used for iteration performance analysis

Independent Variables				**Dependent variable**
Unobserved dummy variable	**Team related variables**	**User story related variables**	**Network properties related variables**	Iteration Performance
Dummy A Dummy B Dummy C	Team size Average experience of team members Average work load per iteration (average stories)	Average size of user story (no of tasks) Average story points allocated per story	Small worldliness (Q) In degree centralization Out degree centralization	

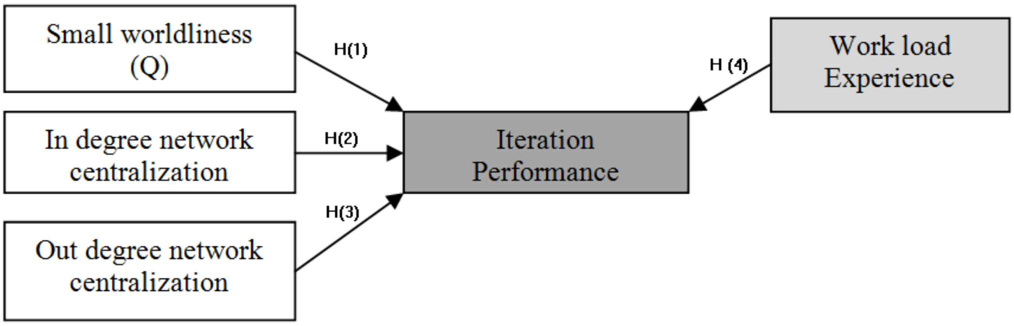


Figure 6.10: Performance model

6.5.4.2.Small worldliness and iteration performance

Close knit agile teams with clustering coefficients and a low average path length show small world network behaviors. The literature supports that small world teams spend more time in discussions related to management or planning and thus, cause a decrease in performance (Cataldo & Ehrlich, 2011). Therefore, keeping in view of the literature's claim and to further the knowledge on the topic with an extensive empirical analysis done in this research, the hypotheses designed are as follows:

$H_0(1)$: The iteration performance of agile teams is not related to its small worldliness network behavior.

$H_A(1)$: The iteration performance of agile teams is negatively affected by its small worldliness network behavior.

The significance value shows the statistical significance of the regression model that was applied and if it is less than 0.05, this means that the model applied can statistically significantly predict the dependent variable. Here, it is $\rho = 0.0375$ which means that the model can significantly predict the performance of the agile teams iterations based on their small world network structures. Therefore, the null hypothesis can be rejected. Hence, our findings support the findings of the literature in which a small world network structure measure was used to measure the iteration performance and quality

(Cataldo & Ehrlich, 2011). The details of the linear regression operation performed on small world network data is shown in Appendix C, Table C.69. The results show that $F_{1,6} = 0.108\, where\ \rho \leq 0.05,\ R^2 = 0.018$ which means that 1.8% (R-square= 0.018) of the dependent variable iteration performance, which in this case, can be explained by the independent variable small world network ratio (Q).

The pictorial representation of the estimations can be seen in Figure 6.11 in which the data points are following the line, which means that the dependent and independent variables are close enough and thus, the performance can be determined with a small world network structure measure. Therefore, the results reveal that small worldliness of the networks strongly affects the iteration performance but in a negative direction. This corroborates the findings that small world network structures negatively affect the iteration performance of distributed agile teams' results (Cataldo & Ehrlich, 2011).

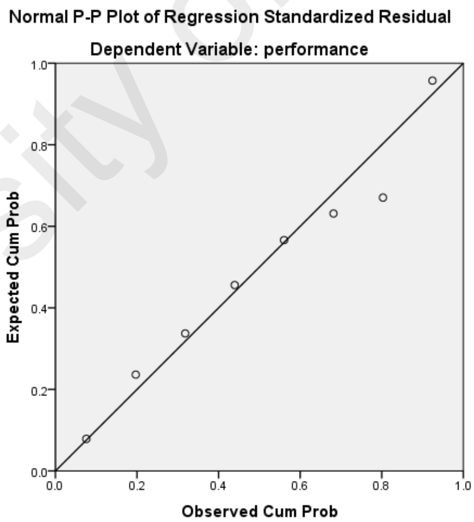


Figure 6.11: Plot of linear regression model results used for predicting Iteration performance through small world network structures

However, the experimental conditions, team size, communication data and data collection methods employed by prior research study differ from this research's design and settings. Moreover, Cataldo considered iteration performance as the number of leftover tasks per iteration and did not consider velocity or story points in his study. Based on the results, hypothesis $H_{A(1)}$ is accepted.

6.5.4.3.In degree and out degree centralization and iteration performance

Centralization structure of teams contributes to a scenario in which information is exchanged among several central members the most. However, the concept of agile teams lies on mutual communication and sharing throughout the project development without few members being information owners. Therefore, it raises very interesting questions to assess the team performance with respect to its network centralization quotient. Literature claims that network centralization is negatively related to group performance (Sparrowe et al., 2001). Likewise, in programing domain centralized teams showed low performance for difficult problems (Mantei, 1981). Therefore, it was highly interesting to investigate the impact of agile teams' centralization index on iteration performance. The hypothesis were based on the findings from literature, are as under:

$H_0(2)$: The iteration performance of agile teams is not related to its in degree centralization.

$H_A(2)$: The iteration performance of agile teams is negatively affected by its in degree centralization

$H_0(3)$: The iteration performance of agile teams is not related to its out degree centralization.

$H_A(3)$: The iteration performance of agile teams is negatively affected by its out degree centralization

The multiple regression analysis showed that 29% (R-square= 0.289) of the dependent variable which is iteration performance in this case can be explained by the independent variables i.e. in and out degree centralization. The significance value in Avova test (p) is less than 0.05, $\rho = 0.042$ it means the model applied can significantly predict the dependent variable and null hypothesis can be rejected. The results obtained state that $F_{2,5} = 1.018\, where\, \rho \leq 0.05,\ R^2 = 0.289$. This shows that in and out degree centralization are significant dependent variables to impact iteration performance for the particular cases under discussion.

The coefficients provided can be used to find regression equation for performance which in this case is:

$$Performance_{Predicted} = b_0 + b_1 * x_1 + b_2 * x_2$$

Here b is the value of beta and x is the unit of estimated variables likes for b_1, x_1 is the average out degree centralization and for b_2, x_2 will be the average in degree centralization. Beta (b) is standardizing coefficients that are obtained after the standardization of all variables i.e. putting all variables on the same scale. The pictorial representation of the estimations can be seen in Figure 7.8 in which the data points lie on and close to the line which assures the results.

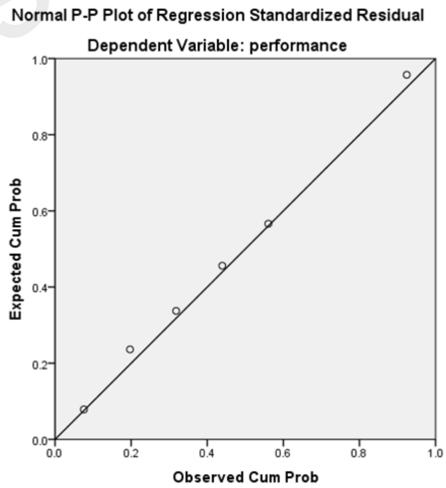


Figure 6.12: Plot of multiple regression model results used for predicting Iteration performance through in degree and out degree

Therefore, the results reveal that out degree centrality is closely strongly affects the iteration performance. However, in degree centralization affects positively but slightly on the team performance and out degree centralization index negatively affects the iteration performance of an agile team.

6.5.4.4.Work load, experience and iteration performance

To assess the impact on performance several other factors were considered such as team members technical experience and work load that is translated as the number of average number of user stories per iteration. There were other factors like average number of story points allocated per iteration and average size of a user story. There were some dummy variables such as culture, language and time zone considered to construct the regression model. However, these variables were later removed from the analysis due to their VIF values greater than 5. Therefore, experience and workload qualified to be a part of this analysis only. According to a common perception the more team members are experienced the better they perform. However, the more workload team handles the less effectively they perform. So the hypotheses built for experience and workload are such as:

$H_0(4)$: The iteration performance of agile teams is not related to team's workload and team members' average experience.

$H_A(4)$: The iteration performance of agile teams is strongly related to team's workload and team members' average experience.

The multiple regression analysis showed that 17% (R-square= 0.17) of the dependent variable which is iteration performance in this case can be explained by the independent variables i.e. workload and experience. If the significance acquired from Avova test (p) is less than 0.05 it means the model applied can significantly predict the dependent variable and null hypothesis can be rejected. Here this value is 0.049 which means the model can somehow significantly predict the performance. The results obtained state that $F_{2,5} = 5.25 \, where \, \rho \leq 0.05, \; R^2 = 0.174$. Since the significance values for both of our independent variables did not follow the range such as $\rho = 0.052$ for experience and $\rho = 0.354$ for workload. Therefore, the null hypothesis can be rejected for experience but not for workload as it opposes the initial expectations i.e. $\rho < 0.05$. This shows that workload is not a significant dependent variable to impact iteration performance for the particular cases under discussion.

The coefficients provided can be used to find regression equation for performance which in this case is:

$$Performance_{Predicted} = b_0 + b_1 * x_1 + b_2 * x_2$$

Here b is the value of beta and x is the unit of estimated variables likes for b_1, x_1 is the average years of experience and for b_2, x_2 will be the workload per iteration. Beta β is standardizing coefficients that are obtained after the standardization of all variables i.e. putting all variables on the same scale. It is weight of an independent variable indicating the expected increase or decrease in dependent variable. Likewise, B variable also indicates how much a one unit increase in the independent variable results in an increase in the dependent variable with the rest of variables held constant. The values of Beta and B variables obtained for work load and experience variables are shown in Table 6.29. The pictorial representation of the estimations can be seen in Figure 6.13, in which the data points lie on and close to the line which assures the results.

Summarizing the results of regression analysis performed for predicting performance analysis of agile teams (shown in Figure 6.14) it can be observed that small world behavior of networks and out degree centralization are negatively related with iteration performance such that an increase in small world behavior or out degree centralization index of a collaboration networks can cause decrease in iteration performance. The detailed analysis of B and significance value is shown in Table 6.26. The results again confirm the negative values of B for small worldliness and out degree centralization.

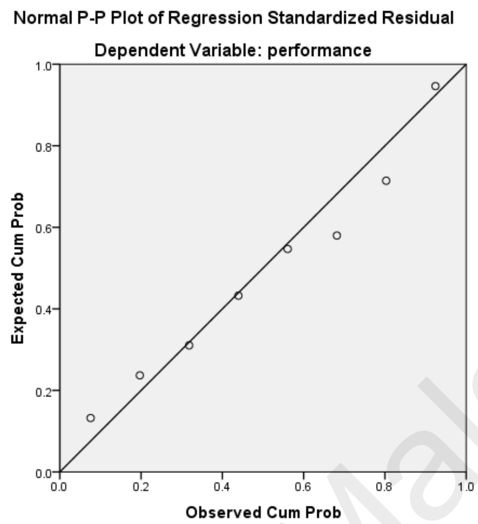


Figure 6.13: Plot of multiple regression model results used for predicting Iteration performance through work load and experience

The significance statistic for the independent variables is used to determine their certainty in finding impact on performance. The significance results indicate that the results assure 64.6 % certainty that experience can affect the performance. Likewise workload has 95% certainty, out degree centralization shows 96.2%, in degree centralization shows 96.3% and small worldliness also shows 96.3% certainty in finding impact on performance.

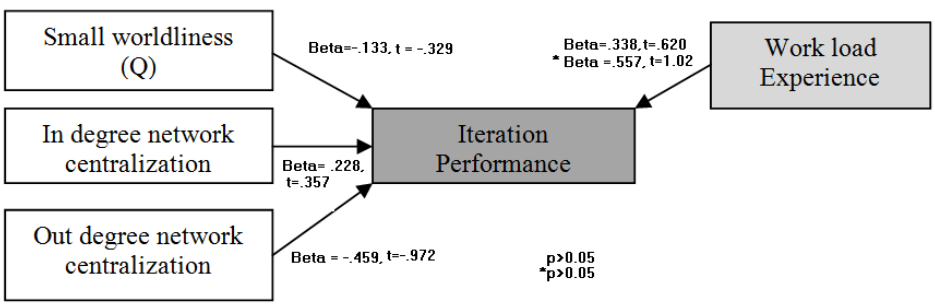


Figure 6.14: Results of regression analysis

Table 6.29: B and significance values for independent variables

Variables	B	Significance (P)
Small worldliness	-2.041	0.037
In degree centralization	10.303	0.037
Out degree centralization	-25.37	0.038
Work load	0.419	0.049
Experience	2.319	0.354
*Enter method *p<0.05		

6.5.6. Practical Implications of the study

This section answers the last research question, i.e. **What are the practical implications of this study for the industry and research fraternity studying Requirements-Driven collaboration among agile teams?** This research has several practical implications for the industry based agile practitioners and researchers, as explained below:

a) Implications for industry practitioners

The empirical investigation of multiple cases performed in this research sheds some light on the collaboration patterns of agile teams. The adopted requirements-driven collaboration approach provides a fine-grained view of collaboration that takes place within certain requirements and sets of dependent ones. Managers can invest in having a well-defined infrastructure in place to allow team members to contact their remote colleagues and practices to allow everyone to know how to work to achieve similar situations. The results invite managers to make better team selection decisions based on the teams' awareness levels. The in-depth analysis of the teams' collaboration patterns provides insights to the management in order to look into the communication and awareness patterns of their teams in order to determine the broken links, communication breakdown, central members, level of communication and awareness among members, thinning of communication among some members, and also to identify closely bonded members among many. Some of the implications based on the particular measures used are stated below:

Looking at the team's density, the management can see the affinity of teams towards a particular kind of discussions by having less and more dense networks. The PM can see how geographically dispersed teams collaborated densely for one reason (e.g. user story negotiation) but participated less for another (e.g. code reviews). The reasons for build failure can be sought as a lack of coordination among distributed team members due to having less communication on a certain issue (Carmel., 1999). The PM can work on a closer interaction by improving the collaboration means or arranging extra hour virtual meet-ups for people who tend to communicate less due to distance.

The management can identify the reasons for which most of the mutual communication happened by looking at the communication reciprocity results. Moreover, the results can be used to identify the communication breakdowns by looking into the less reciprocity index of communication reasons and probe the reasons why virtual agile teams did not communicate mutually when they are supposed to. In addition, the ties reciprocity analysis can help the management to visualize topics of mutual interest for the team by looking at the maximum reciprocated ties.

In addition, the core periphery measure results can help the management to identify loosely connected members. This way, less active members can be identified and later, the reasons for their low participation can be sought. The management can improve the overall build performance and reorient the team members to work closely (as core members) if needed by taking suitable measures.

The management can also identify isolated nodes by making use of the component measure results. The results can be helpful in identifying the communication breakdowns and orienting better communication among teams by getting the left-out members involved.

By identifying the central members in information flow through centrality and betweenness measures, the management can identify pivotal members of the teams and use this information for structuring the team for an even information dissemination. Moreover, necessary actions can be taken to increase the involvement of members into more project related discussions by conducting frequent standup meetings and providing more logistic support by enhancing the virtual collaboration infrastructure. In the case of highly centralized networks, steps can be taken by the management to decentralize the information structure of the teams for information distribution among the rest of the members.

The management can point out the core members (defined as cut points or cut sets) in the team to determine whether their absence can cause a problem for the team. In addition, this also provides the management with insights on the members carrying most of the useful information. The management can take necessary measures to avoid this risk and reduce the loss in the case of such members' absence. The management can impose a decentralized team structure in which all of the members feel equally responsible towards the project and play their respective part without having a few of them playing an integral part.

The tool designers can make use of this study to develop a comprehensive tool which extends the developed prototype which collects data and converts them into a form acceptable for commercially available social network analysis tools. Such a tool can be designed which aids the process of data collection, network creation, and social network measures selection and application on the data. The impact of certain features on each other and the team's overall performance can also be incorporated within.

b) Implications for researchers

The findings of this research also suggest topics for future academic investigation. To extend the study of requirements-driven collaboration among agile teams, researchers can examine multiple kinds of team settings such as teams working on open source software development and outsourcing teams. Researchers can make use of email data, online repository data, comments and commits made on source code to assess the collaboration among agile teams for a deeper understanding of the collaboration among teams by making use of the artefacts of requirements which has not been considered in present research.

The current study indicates that distance does not seem to matter, despite the apparent contradiction between communication frequency and distance. However, the correlation results between distance and the communication and awareness factors are inconclusive when looking across the four projects. It would be interesting to investigate other projects with similar or larger distribution configurations to learn whether these findings will hold.

The findings also revealed that project managers are still the key players in agile teams. Although the members playing this role acted as mentors, it would be interesting to expand the investigation of this role. Ideally, it would be valuable to collect self-reported data on how team members perceive the help that they received from mentors (e.g. Scrum Masters, Coaches, or Project managers as called in our study) to develop and manage user stories.

Another interesting investigation is to identify the extent that past knowledge from others helps in getting the project done. The results show that familiarity of others' helps in facilitating awareness but it is unknown to what extent knowledge from the past is required to achieve such situation. A follow-up with the investigated companies could shed some light into this topic.

The study results revealed that the agile teams' performance increased with time and project growth. The quality of requirements, team relationship and their effect on project success can also be studied in the context of agile methods on the same footings. In this research, the researcher managed to study only two iterations of the overall project's life span. However, a longitudinal study comprising results based on the observation of the whole life span of a project, right from the planning to the release and post-release maintenance phase, can be a promising future dimension to study. This will help to analyze the team's collaboration over a long period of time and will aid in providing more robust findings.

6.6. Summary

This chapter describes Case 1 in detail in order to study communication and awareness among distributed agile teams by following the step by step guide provided by the proposed framework. The rest of the cases including Case 2, 3 and 4 were conducted on the same patterns and are placed in the appendix for further reading. The results and discussion section revisits the proposed research questions and provides answers to all of them based on the empirical investigation of the case studies

CHAPTER 7: VALIDATION

This chapter explains the validation performed on the proposed framework. Three types of validation were performed: structural validation, applicability validation and utility validation. Structural validation is performed to evaluate the constructs of a framework (explained in Chapter 5, Section 5.7). Applicability validation is performed to assess potential applications of the proposed framework. Utility validation is conducted to gauge the usefulness of the proposed framework. In addition, several validation strategies were adopted throughout the research process as suggested by Creswell (2009) and were particularly used for validating constructive empirical research in software engineering (Singer, Storey, & Damian, 2002). This chapter explains the step by step execution of the validation strategies adopted throughout the research process and the one adopted in the end to validate the quality of our findings.

7.1. Validation process

The aim of validation is to show that the proposed objectives were fulfilled. The validation process in qualitative research acts as a connection between data and conclusions (Bryman, 2008). It tends to describe the process as being carried out in a systematic manner and shows the transparency of the research process. Moreover, validity in qualitative research is defined as the extent in which the conclusions match the social phenomena to which it refers to (Hammersley, 1990). Therefore, the qualitative research validation process involves constant reflexivity and self-scrutiny (Finlay, 2002) throughout the research process. Hence, several validation methods were employed to ensure the validity of this research throughout the research process as proposed by (Creswell, 2009). Furthermore, to confirm the structural validity of the proposed framework, interviews were conducted with researchers and academicians with relevant work experience in the field. To confirm the applicability and usefulness of the proposed solution, interview sessions and survey were conducted with industry practitioners to ensure its practical relevance (Lassenius et al., 2001). The description of all of the validation strategies is explained in detail below.

7.2. Applicability Validity

This validation method aims to evaluate the proposed framework in order to study the collaboration of agile teams and find its impact on performance with respect to its potential application in the industry. More specifically, a goal question metric (GQM) statement for the evaluation was created and shown in Table 7.1.

Table 7.1: GQM statements to perform validation with industry practitioners

To Analyze	The processes to study socio-technical aspects of RDC in agile teams
In order to Evaluate	With respect to 1. Usability 2. Applicability
From the perspective of	Agile industry practitioners
In the context of	Questionnaire based survey
Because	Suggestions of practitioners help to improve and validate the results

Questionnaire was chosen to gain information from industry practitioners. The questionnaire consisted of closed and open ended questions. The closed ended questions were Likert scale based ranging from 1 to 5, while the open ended questions were regarding the demographics of the interviewees and their suggestions for improvement. The responses provided useful suggestions which led to an improvement in the research contributions by making it more applicable for the industry.

7.2.1. Participants

The participants of this questionnaire were on the basis of their professional background and work experience in agile based work environments. The invitations along with the details of the validation study were sent to 12 industry practitioners. However, only nine showed willingness to participate. The role and country wise distribution of the participants is shown in Table 7.2. The rest of the details, e.g. company and the person's name, are kept hidden intentionally in order to keep the terms of the personal data privacy.

Table 7.2: Roles of the industry practitioners

No.	**Industry practitioners role**	**COUNTRY**
1.	Team Lead	Malaysia
2.	Scrum Master	Malaysia
3.	Project Manager	Malaysia
4.	Senior Developer-Team lead	Malaysia
5.	Project Manager	Pakistan
6.	Business Analyst	Pakistan
7.	Team Lead	Pakistan
8.	Senior Developer	Sweden
9.	Ceo-Agile coach	US

7.2.2. Research Instrument and Procedure

The questionnaire is prepared (shown in Appendix D, Table D.1) to gain feedback from the practitioners in order to determine the applicability of the proposed solution in studying collaboration among agile teams. A handout, in the form of a presentation comprising the necessary information to be shared, was prepared and sent beforehand to the participants.

7.2.3. Analysis and results

The questionnaire responses were recorded in an Excel sheet and the questions were named for ease in the analysis. The category names of questions are shown in Table 7.3. The responses were first tested against the reliability test using a statistical measure called Cronbach's Alpha (Cronbach, 1951). George & Malerry (2003) provided the following guidelines to assess the Cronbach's alpha values, such as if the value is greater than or equals to .9 – Excellent, .8 – Good, .7 – Acceptable, .6 – Questionable, .5 – Poor, and .4 – Unacceptable". The increase in the value of alpha partially depends on the number of items in the scale (Gliem & Gliem, 2003a). The alpha value for the set of responses was 0.72, which falls under the range of acceptable responses (as shown in Appendix D, Table D.2)

With regards to the composition of the framework; in the set of steps followed to study collaboration among agile teams, the practitioners' responses showed agreement (mean=2.1). Moreover, the practitioners' found the phases understandable (mean=1.8). This shows that the scheme of steps followed to study the socio-technical aspects of agile teams made sense to the industry practitioners. In addition, the respondents found it easy to relate to the flow of phases to study collaboration among teams and didn't find it complex (mean=2.0). The respondents were convinced of the practical usage of this set of steps to study the collaboration that happens between real world agile teams (mean=2.3). The respondents found the proposed way to study collaboration among agile teams practical because it provides them with the insights on the collaboration that is happening among teams (mean=1.9) and helps them to identify the impact of this collaboration on the team performance (mean=1.9). Based on the usability and particularity of the proposed framework that is already evidenced by the responses, the respondents showed agreement towards incorporating the framework in their future endeavors for team performance calculation and collaboration analysis (mean=1.8).

Table 7.3: Questions and their respective categories

Question	Category name
1. At present do you have a systematic process in use to view the collaboration practices of the teams?	Presently used process
2. The framework has following phases. Do you think all of them are necessary? (1) The conceptualization phase, (2) The visualization phase, (3) The characterization of communication and awareness among agile teams, (4) The impact analysis phase	Composition
3. The framework is understandable step by step for the following phases: (1) The conceptualization phase, (2) The visualization phase, (3) The characterization of communication and awareness among agile teams, (4) The impact analysis phase	Understandability
4. How do you rate the complexity level of this framework as:	Complexity

5. How do you think the framework has potential to be practically implemented	Particularity
6. How do you think the results can help you to see the insights of your team's collaboration practices?	Usability1
7. How do you think the impact analysis can help you to assess the performance of your team in the light of their collaboration practices?	Usability2
8. Do you look forward to use this framework for collaboration and performance analysis of your team in future?	Future usage

The analysis was broadened using the relationships between several aspects such as how the proposed set of steps' understandability and complexity affects its usability, and how its particularity affects its future usage. For this purpose, correlations were computed between the dependent and independent variables as shown in Table 7.4.

Table 7.4: Spearman correlation results between dependent and independent variables

Independent variable	**Dependent variable**	**Spearman Correlation analysis** r_s	ρ
Process in use	Future usage	-0.416	0.067
Practical viability	Future usage	0.650	0.049
Usability1	Future usage	0.779	0.035
Usability2	Future usage	0.635	0.041
* $\rho < 0.05$			

The main aim of performing this analysis was to discover the cause-effect behind the willingness of respondents to use the proposed framework in the future. Spearman correlation measure was used to determine the relationship between the dependent and independent variables. Major focus was kept on the respondent's willingness in using this framework for their future team collaboration analysis based on their previous experience and the practical viability of the proposed system. Therefore, future usage is a dependent variable which depends on several independent variables such as the process already in use to investigate collaboration patterns of agile teams, the practical viability of the proposed solution, the usability of the proposed solution to investigate the collaboration insights, and the usability of the proposed solution to study the impact analysis of the collaboration patterns on team performance. According to the respondent's view, the future usage of the proposed solution strongly correlated with usability1 which states that the results of this research study can help them to see the insights of the team's collaboration practices ($r_s = 0.779\ at\ \rho = 0.035$), where p > 0.05. A moderate correlation was found on the usability of this research results in analyzing the team performance ($r_s = 0.635\ at\ \rho = 0.041$), where p > 0.05. Likewise, a slightly moderate correlation was found between the practical viability of the current research and its future usage by industry practitioners ($r_s = 0.650\ at\ \rho = 0.049$), where p >0.05. However, a negative weak correlation was found between the presently used approach to study team collaboration and future usage of this research result ($r_s = -0.416\ at\ \rho = 0.067$) , where p > 0.05. Therefore, the result is insignificant and hence, cannot be considered as a reliable predictor of future usage. Overall, the independent variables managed to explain 58% of the dependent variables as shown in Figure 7.1.

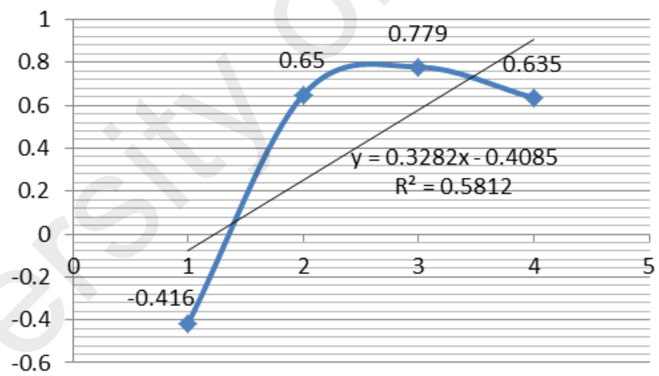


Figure 7.1: Relationship between deepened and independent variables' correlation

7.3. Utility Validation

Structured interviews were conducted to evaluate the usefulness of the proposed solution. For performing utility validation, the same data was required as discussed in the case studies. Therefore, Company 1 was contacted again to participate in this part of the study and help in validating the results. Company 1 was selected for this purpose because of two reasons. Firstly, it was due to the agreement of the company's management to participate in this phase of the study, and secondly, due to its physical proximity with the researcher. The GQM statement for utility validation is shown in Table 7.5.

The utility validation was performed as a result of the structural validation process in which the respondents suggested developing a prototype for deployment in the industry. As part of the necessary action taken to comply with this suggestion, a prototype was designed and used for data collection and data transformation into a social network analysis tool (UCInet used for this study) readable format. For this purpose, a 2 weeks long iteration of a recently started project was captured and the results were shared with the project manager, project owner and business analyst and the team leader (senior developer) for validation.

Table 7.5: GQM statement for utility validation

To Analyze	The processes to study socio-technical aspects of RDC in agile teams
In order to identify	With respect to Usefulness
From the perspective of	Agile industry practitioners
In the context of	Questionnaires
Because	Suggestions of practitioners help to improve and validate the results

7.3.1. Project, Participants and Procedure

The project was a content management system development for a document management company with open source data. The document management system was designed for version control, maintaining file history, metadata recording, scanning and etc. It was developed to help the customer manage, more efficiently, the collective intelligence of the human resources of the company.

The team size consisted of 9 members: Project Manager, Product owner, business analyst, software developers (2), software testers (2), and designers (2). The team members have an average of 1.5 years’ experience working in the same roles. The team was collocated on the Malaysian site only. The team members used to communicate with each other mostly face to face because they were all seated in a medium sized room around a round table facing the story board. The project manager and product owner usually pay the team a visit and meet with the team for daily scrums.

Data was collected through the prototype developed during a joint session with the team members. The executable was installed on their personal computers and then they were guided to follow a step-by-step data entry process. The data was then recorded automatically in a folder in the form of Excel spread sheets. The reports were later generated to feed into the social network analysis software in order to gather the results. The social network measure results were then statically compared to assess the team performance and the results were collectively shared with the team's senior and managerial members. A questionnaire consisting of Likert based questions was used to gather the responses from the managerial members of the team in order to collect their feedback regarding the results of the study, as shown in Appendix D, Table D.3..

7.3.2. Analysis and results

Since the questionnaires were based on Likert scale questions, descriptive statistical operations were used to analyze the results. The mean (M) and standard deviation (SD) of responses are shown in Table 7.6.

Table 7.6: Mean and standard deviation of the responses used for utility analysis

	Question	M	SD
1.	The results exposed me to the team insights that are important to manage teams.	1.75	0.5
2.	The results exposed me to the team insights that are important to improve team performance.	2.25	0.5
3.	It is a quantifiable process to determine the collaboration structure of my team.	2.5	0.57
4.	It is a quantifiable process to determine the performance structure of my team.	1.25	0.5
5.	The results helped me to make changes in the orientation of my team.	2.5	0.57

6.	The results exposed me to the team insights that are suitable for this project.	2.5	0.57
7.	Do you look forward to use this framework for collaboration and performance analysis of your team in future?	2.25	0.95
8.	How would you rate the necessity of this study?	2	0.81
9.	Any suggestions for improvement?		

The means values were used to determine the inclination of the results towards the respective Likert scale values. For instance, Mean= 1.75 for question 1 shows that the respondents agreed to the fact that the results helped them to see the team's insights which is important to manage teams. To check the reliability of the results, Cronbach's Alpha was calculated and the value yielded was 0.78 which is considered acceptable, according to (Gliem & Gliem, 2003b). For further analysis, Spearman's rank analysis was used. Table 7.7 shows the correlation of dependent and independent variables. The dependent variable is correlated with the rest of the independent variables using Spearman's rank analysis. The data was analyzed using Spearman correlation analysis because monotonic relation between two variables, i.e. dependent and independent variables, was intended through this analysis. The results (as shown in Table 7.7) showed that the future usage of the proposed solution was largely dependent on the fact that it helped them to change their team orientation for better collaboration and results (r_s = 0.905), provided them with insights which is important to manage the teams (r_s = 0.87), and to improve team performance (r_s = 0.816). Moreover, the respondents agreed on the fact that the proposed solution is a quantifiable process which helps them improve their team performance (r_s = 0.816). However, a low correlation index was achieved in response to the respondent's perception on the suitability of the proposed solution for the particular project (r_s = 0.55). All of the correlation values achieved were significant at $\rho < 0.05$, except the correlation of the proposed solution being a quantifiable process to determine the collaboration structure of the team and its future usage chances ($r_s = 0.55$). Overall, the independent variables managed to explain 55% of the dependent variables $r^2 = 0.553$ as shown in Figure 7.2.

Table 7.7: Spearman rank analysis results

Independent variable	**Dependent variable**	**Spearman Correlation analysis** r_s	ρ
Q1. Team management	Future usage	0.87	0.031
Q2. Improve team performance	Future usage	0.816	0.049
Q3. Quantifiable collaboration structure	Future usage	0.707	0.056
Q4. Quantifiable performance of teams	Future usage	0.816	0.023
Q5. Changes in team orientation	Future usage	0.905	0.039
Q6. Suitability for the project	Future usage	0.55	0.012
* $\rho < 0.05$			

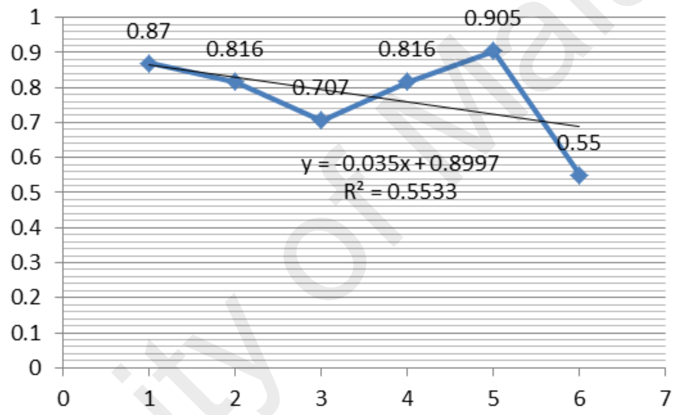


Figure 7.2: Relationship between dependent and independent variables' correlation

7.4. Validation strategies adopted throughout the research process

The Creswell (2009) proposed validation strategies implemented throughout the research process are described below.

7.4.1. Triangulation

Multiple data collection sources were used to achieve each of the research objectives and answer each respective research questions. Data triangulation helped to attain accuracy of the collected data before starting the analysis process. Interviews with certain team members were conducted to confirm the data collected through formal document inspection and observations during on-site stay on the requirements, teams, project details, and workflow, among many. For instance, user story allocation was not documented in the feature list (a document stating the product requirements list) and similarly, there were other small discrepancies which were sorted out beforehand from the interviews. Therefore, a series of interviews were conducted with project managers and developers (the ones who were available) in order to gather in depth information.

The data was then checked against the collected data through questionnaire responses. Also, communication data collected through the questionnaire was cross checked against work diaries deployed at the sites. This was very useful in filling out the missing data and in cross checking the validity of the responses filled by the teams. The notes taken during on-site observation were also checked against the information filled in the questionnaires and those gathered from the interviews.

7.4.2. Member checking

Member checking is when the researcher restates the insights gained and checks the accuracy of the collected information with the team members (Harper & Cole, 2012). The members' agreement or disagreement determines the accuracy and authenticity of the study (Creswell, 2009). This ensures the researcher that the interpretations of the data make sense. In addition, the change in the participants' perception in the course of time can also be covered only through revisiting the previously recorded findings through member checking (Koelsch, 2013). The on-site observation period provides an opportunity to cross check the data gathered with the members. The members were inquired several times on the work flow, user story interdependencies and iteration plans. The information was required to determine the impact analysis for this study. The main concern was twofold; first, to get clarification on certain aspects and second, to gain assurance of understanding the system correctly. The collected data was often discussed with the project managers. The rest of the team members were not available in most cases except for several planned interviews during the data collection process due to organizational policies. Therefore, the project managers were usually the one contacted for affirmation of the collected data in almost all of the cases.

7.4.3. Interview to confirm the findings' accuracy and usefulness

This is the mechanism of revisiting the participants for results validation in terms of its accuracy and usefulness. After completing the analysis and having procured the results, the teams were contacted once again to discuss the results with them. The interview sessions were fixed with several members from each project team (likely 1 or 2). The interviews were conducted face-to-face with the members present in Malaysia and Pakistan to discuss the partial results (of one iteration only) and through Skype and email for the members located at the rest of the locations. Mainly, senior team members such as the project managers, product owner and team leaders were included in these interview sessions. The interview sessions were followed by a 15-20 minutes presentation in which the results and analysis were presented to them. The interviews were scheduled after the first iteration's data was recorded and analyzed and before the start of the next iteration in order to improve the interpretations of the next iteration as per the participants' suggestions. The GQM statement organized for this validation step is shown in Table 7.8 and the interview guideline is shown in Appendix D, Table D.4. The GQM (Basili, Caldiera, & Rombach, 1994) was used to define the objectives of validation and consists of several steps such as: develop the objectives, generate the questions, specify the measure, decide on the data collection methods, its validation and analysis, and deduce the conclusions and generalization of the results. Therefore, a GQM objective statement (Basili, 1985) consists of the following; Object: The product or process under study e.g. testing phase or a subsystem of the end product, Purpose:

Motivation behind the goal, Focus: The quality attribute of the object under study, Viewpoint: Perspective of the goal, and Environment: Context or scope of the measurement program.

Table 7.8: GQM statements to perform validation with industry practitioners

To Analyze	The preliminary findings of collaboration patterns found in agile teams
In order to Evaluate	With respect to 1. Accuracy of interpretations 2. Usefulness of results
From the perspective of	Study participants from each company
In the context of	Face-to-face and online questionnaire based interview (researcher)
Because	Suggestions were required to improve the interpretations of results

These interviews were an interesting experience for interacting with the people whose interaction data shaped the results. There were several instances where the interviewees helped to conclude the reasons behind fewer interaction or communication lapse between two people. For instance, in Case 1-Project Alpha, in the beginning, the minimal participation of Quality Engineer 2 (QE2) was perceived to be due to the lack of awareness with the rest of the team members being paired for the first time in the team. However, the participants later on clarified on this point where QE2 was actually on medical leave and that was the reason for her participating less and being in a less active state. However, for the User experience designer (UX), it was explained that his commitment towards three projects at a time resulted in fewer interactions with the rest of the team. Moreover, it was also clarified that UX was not required to be a part of the daily sprint and in project discussions as a mandatory member. The relevant changes in requirements were sent to him through email. This kind of organizational work fashion shapes the interaction patterns of the teams which were clarified through these interviews.

These interview sessions served as a review and discussion session and helped to validate the observations and preliminary data. Data analysis and results presentations not only play an important role in confirming the accuracy of the findings, but also in discussing their usefulness. The participants, mainly managers and senior team members, were glad to learn about the collaboration patterns of their teams in black and white. Moreover, they found it surprising to discover that team members' collaboration was different than what was assigned, and that previous project's awareness of team members facilitated the communication between them.

The impact analysis was also reassuring for the managerial members knowing that the teams worked well as per their communication and awareness structures despite the fact that the key players were the project managers.

The participants showed interest in the identification of the most active members and the members who carried most of the information within the team. It had also been suggested to have a more fine grain analysis of an individual's position within the team. Based on that, several SNA measures such as betweenness centrality and Eigen vector centrality were added to the process. However, the privacy and confidentiality terms were kept in mind when discussing members' performance oriented measures and their names were not disclosed.

These interviews were repeated after the second iteration and useful feedback was gathered from the participants in terms of the interpretation of the results.

7.4.4. Rich and thick descriptions

The role of a detailed description to interpret the research setting and findings of the research is an important step for result validation. This research invested ample time in on-site observations and data collection. This time allowed the researcher to familiarize with the work environment, work flow and team members. Close observation and the time spent on site helped in interpreting the findings. Interviews conducted with the participants for data collection and interpretation validation provided a rich contextual source of understanding the collaboration among agile teams in the investigated projects. The time invested on site to gain an in-depth knowledge on the team members and their work flow helped to shape the conclusions on the results found.

7.4.5. Clarify bias

Removing any kind of bias in the study will definitely improve the quality of the analysis and results. To avoid any kind of bias in the observation and interpretation, interview sessions (explained in section 7.5.3.) were conducted. The conclusions drawn on the basis of the researcher's observation were cross checked and corrected where required by the managerial team members. Introducing this measure has helped immensely to reduce bias from the study.

7.4.6. Prolonged contact with participants

The ample time the researcher spent on site to gain exposure to the subject population being studied affects the validity of the results. The companies selected for this study had their offices in Malaysia, Pakistan, Philippines, UK and US. Two of the aforementioned locations were in access of the researcher. For the rest, virtual means of communication were utilized. On average, 2-3 months were spent on each project's site in Malaysia and Pakistan for data collection purposes. The time frame spent on site is comparatively longer than the average time slot spent on site in an agile team ethnographic study (Sharp et al., 2003) that may last for a week and another ethnographic study aimed to study communication patterns of agile teams (Abdullah, et al., 2011) that lasted for four days. Therefore, the time spent onsite and the number of cases observed provides deeper insights than the examples found in the literature. The companies, after signing the data disclosure and privacy terms documents, provided the researcher a chance to observe their work environment and be a part of some of their daily stand up meetings. However, the researcher was not allowed to talk to members or indulge them in any kind of conversation during the observation phase. The interviews were separately scheduled whenever needed. The big touch screen infrastructure installed at four of the two sites observed facilitated in observing the remote teams as well.

7.4.7. Peer debriefing

It is through discussions among researchers or colleagues that assumptions and results are validated. The study's assumptions and results were discussed at each point with the supervisor in order to get feedback to improve the course of action. In addition to the supervisor, our research collaborator was another person with whom these assumptions, findings, results and interpretations were discussed to improve the validity of the results. She, being an experienced researcher and have worked on the same kind of extensive field study, always helped to uncover important aspects which helped to shape the findings of this study.

7.4.8. External auditor

In order to have unbiased and neutral auditors for this research, articles were sent to journals and international conferences throughout the research period. This way, it became possible to gather formal feedback from reviewers of high repute and with ample research experience which helped to improve the research quality. The reviewers' comments were accommodated to shape this research in its present form.

7.5. Tying together the research objectives and contributions

Validation aims to show that the research results satisfy the research objectives proposed when starting the research. The literature study revealed that there is a need for a formal set of steps to study the socio-technical aspects of RDC among agile teams. Moreover, the accompanying of empirical results regarding the distributed agile teams can further the knowledge on the topic. Therefore, several validation methods were used to ensure the feasibility and particularity of the solution. The summarized view of the above mentioned validation techniques and the objectives they tend to fulfill are presented in Figure 7.3.

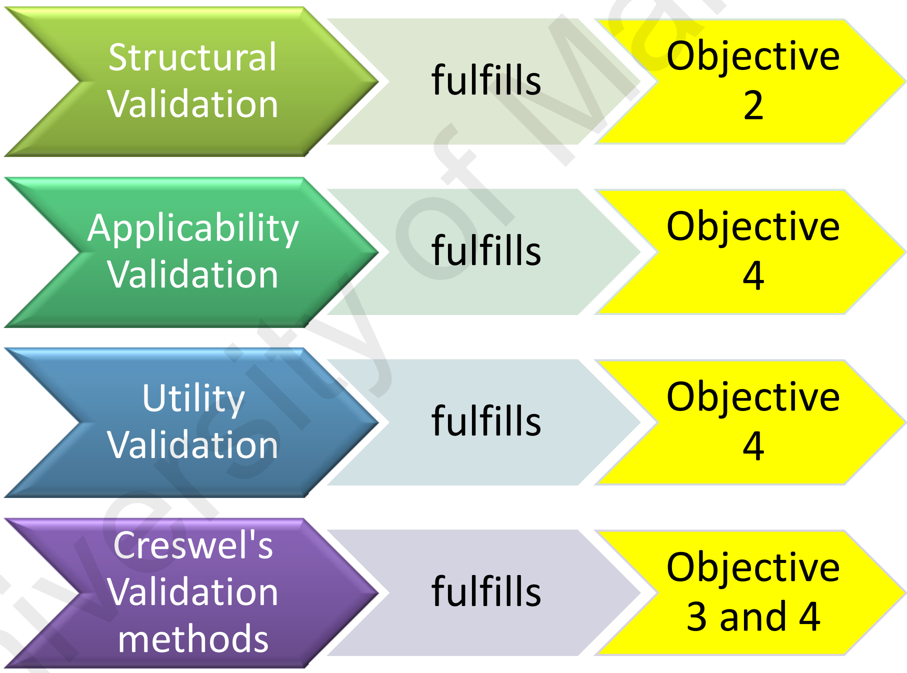


Figure 7.3: The validation methods fulfilling corresponding objectives

Structural validation helps to fulfill objective 2 which states "To formulate a framework for studying requirements-driven collaboration among agile methods". It helped in gathering expert opinions on the soundness and feasibility of the proposed framework. The application and utility validation methods helped to fulfill objective 4 which states "To evaluate and assess the framework". In addition, the Cresswel's eight validation techniques applied throughout the research process helped to achieve objective 3 which states "To investigate the requirements-driven collaboration among agile teams and find its impact on the iteration performance".

This contribution of the validation results leads to:

1. Enabling the study of the socio-technical aspects of RDC among agile teams
2. Increasing the collaboration study's usability and applicability in industry and research

7.6. Summary

The validation process was employed to connect the research process and outcome with the proposed research objectives (Chapter 1). For this purpose, structural, application and utility validation were performed to evaluate the constructs, application and usability of the proposed framework.

In addition, eight validation strategies including triangulation, member checking, interview with the members, rich and think descriptions, clarify bias, prolonged contact with members, external auditor, and peer debriefing were implemented throughout the research process to validate the credibility of the research. The evaluation results showed that the proposed solution is useful for the industry practitioners and can be applied in the field.

CHAPTER 8: CONCLUSION

This research was conducted to study requirements-driven collaboration among agile teams. The socio-technical aspects explored were identified as communication and awareness, both being the two most relevant socio-technical aspects of requirements-driven collaboration among agile teams. A set of steps was followed in order to study the socio-technical aspects of requirements-driven collaboration i.e. communication and awareness among agile teams. This chapter reinstates the research findings with regards to their respective research objectives and research questions which were answered during the course of this research. In addition, this chapter provides implications for the industry and research practitioners, highlights the contributions, identifies the limitations and paves ways for future work.

8.1. Summary

Requirements engineering activities demand stakeholder collaboration and so do agile methods. However, unlike traditional software development methods, agile methods are dynamic and flexible to accommodate changes in the requirements throughout the development life cycle. Unfortunately, there are just a few studies that explain the requirements engineering activities in agile software development (e.g. (Yu & Sharp, 2011a)(Yu & Sharp, 2011b)(Bjarnason et al., 2011a)). Therefore, it is indeed interesting to study the 'agile way' of dealing with requirements. This research probes into the socio-technical aspects of collaboration driven by requirements, among agile methods. The literature supports the fact that there are a handful of studies that discuss the socio-technical aspects among agile teams such as communication (Bjarnason et al., 2010)(Bjarnason et al., 2011b)(Yu & Sharp, 2011a), trust (BredeMoe & Smite, 2008; McHugh, Conboy, & Lang, 2012; Tjørnehøj, 2012), organizational culture (Robinson &

Sharp, 2005a), physical ambience of agile teams (Mishra, Mishra, & Ostrovska, 2012) and etc. However, there was a need to identify the most relevant socio-technical aspects of collaboration among agile methods. Therefore, this research was formulated in such a way as to (a) identify the most relevant socio-technical aspects of requirements-driven collaboration among agile teams, and then (b) device a set of steps to study the socio-technical aspects of requirements-driven collaboration among agile teams. In doing so, the research sought answers to the following research objectives:

- To identify the relevant socio-technical aspects of requirements-driven collaboration among agile teams.

An online survey was conducted on agile based industry practitioners to assess their perceptions on collaboration. The responses were gathered by using emails and through a professional social network. First, the questionnaire was uploaded to several relevant groups on the professional social network, Linkedin.com. Second, the questionnaire was emailed to several industry professionals having agile based work experience. The survey was conducted for a month and a total of 103 responses were collected from both of the sources. About three-quarters of the responses were gathered through the LinkedIn network while only about one-quarter were collected through e-mail. The survey results revealed that the agile based industry practitioners perceived collaboration as communication between them, their awareness of each other. The findings i.e. communication and awareness, were also supported by the literature in which these aspects were focused by the researchers for agile teams. However, studying awareness of team members among agile teams was a new and less researched dimension in the literature. Therefore, communication and awareness were chosen to be studied among agile teams.

Following the identification of the most relevant socio-technical aspects of requirements-driven collaboration, a systematic literature review was conducted to study the identified aspects in the literature. This review helped to broaden the understanding on the topic and increase knowledge before investigating them in real world teams. The findings showed that communication had been studied from two perspectives in the literature: 1) communication media and their effects; and 2) communication structures and their effects on requirements-driven collaboration. Likewise, the review revealed that awareness had also been studied from two perspectives in the literature: 1) factors influencing awareness; and 2) factors influenced by awareness. Moreover, the findings revealed that virtual or rich means of communication are highly effective for requirements negotiations among distributed teams. The literature supports the fact that communication structures affect the following aspects: team awareness, social climate of team, knowledge brokerage, interaction between different roles and iteration performance (agile methods). It was established that communication triggers group performance, reduces cost of repetition, avoids rework and instigates mutual agreement. Awareness was also considered equally important for the team's collaboration. The literature argues that awareness among teams is affected by communication patterns. Other factors affecting awareness are distance between teams and experience of teams, and organisational culture. Therefore, it can be concluded that awareness helps in reducing the costs incurred through miscommunication with the wrong people and communication breakdown; in addition, it leverages knowledge acquisition and encourages the creation of new ideas.

- To formulate a framework for studying Requirements-Driven collaboration among agile methods.

After the identification of the most relevant socio-technical aspects of requirements-driven collaboration, the next step was to design a set of steps to study the identified aspects among agile teams. The literature only provided a couple of examples on how communication, as a socio technical aspect, had been studied among agile teams (e.g. (Pikkarainen et al., 2008)(Cataldo & Ehrlich, 2011)). Therefore, there was a need to device a formal strategy to study socio-technical aspects such as communication and awareness among agile teams. Therefore, drawing from Damian and her colleagues' work (Damian et al., 2010), a set of steps was proposed which guides the way to study communication and awareness among agile teams and helps to analyze the impact on the performance of agile teams. The proposed framework intends to resolve untouched issues from previous studies, i.e. (i) dealing with non-traditional software development teams (i.e. agile teams); (ii) measurement of team performance; and (iii) longitudinal study (two iterations). Therefore, a framework was formulated to study the requirements-driven collaboration among agile teams. The social network analysis measures were used to assess the networks' properties and statistical operations were used to measure the team performance based on its collaboration in terms of communication and awareness. The framework conceptualizes the requirements-centric agile teams, requirements-centric agile social networks, visualizes the networks as sociograms using commercially available networks visualization tool, i.e. Gephi, and then analyses the network properties using SNA measures to find the impact of collaboration on iteration performance..

- To investigate the characteristics of the most relevant socio-technical aspects of Requirements-Driven collaboration in agile methods.

The communication and awareness networks were characterized by applying social network analysis measures using a commercially available social network analysis and visualization tool, Ucinet (Borgatti et al., 2002). First of all, to assess the network trends, a manual characterization was done to calculate the links within and cross site, within and cross role and etc. The recorded communication and awareness through questionnaires and post questionnaire semi-structured interviews were stated in Excel spreadsheets. The presence and absence of a relationship between two actors was mentioned by 1s and 0s respectively. The number of links helped to analyze the trends of networks, for instance, the actors communicated more with within-site members than with cross-site colleagues, the actors were more aware of cross-site colleagues than of within-site colleagues, etc. Likewise, this manual ties statistics helped to determine the basic trends that the teams followed for maintaining communication and awareness throughout the iterations.

Secondly, the recorded communication and awareness data were transferred into square matrix format in Excel spreadsheets with 1s and 0s representing the presence and absence of relationships respectively. These matrices were then loaded into the UCInet tool and converted into system acceptable files. These files were later used for social network analysis. The network was first analyzed using simple measures such as network density and size. This helped to determine the nature of each network. Then, the networks' structures were analyzed using SNA measures such as centralization, Betweenness, ties reciprocity, core periphery, cliques and etc. These measures helped to determine the basic characteristics of the communication and awareness networks, for instance, are the communication/ awareness networks centralized, was the communication/awareness between members reciprocal, are there any groups of people working closely than others, etc.

Finally, the information flow was studied among communication networks by applying SNA measures such as cut points, components, reachability, clustering coefficient, etc.

This helped to determine the patterns of information exchange that the agile teams follow while being at distributed locations. The analysis helped to identify if there were any lone nodes or groups of lone nodes, whether the members were reachable by all, were there team members holding all the information, etc.

Furthermore, the impact of agile teams' communication and awareness was measured on iteration performance. The agile team performance was studied in relationship with collaboration in the literature (e.g., (Cataldo & Ehrlich, 2011)). The results revealed that hierarchical team structures were positively associated with the iteration performance and negatively related to the iteration quality. However, small world team structure was negatively associated with the iteration performance and positively associated with the iteration quality (Cataldo & Ehrlich, 2011). Drawing on the literature, the performance was measured by keeping in view the number of tasks allocated and completed at the end of each iteration and the story points earned per iteration which is called the team's velocity. To determine the team's performance, several details were considered such as team size, maximum tasks allocated per person and the iteration size, network centralization, small world behaviors, the team's average work load and the average experience of teams. The results showed that small world network structures negatively affect the team performance and thus corroborate the results of Cataldo & Ehrlich (2011). However, the performance measurement metrics and the teams' collaboration data were different in both of the research. The in-degree centralization was slightly but positively related to team performance while out-degree centralization was strongly but negatively related to team performance. Likewise, the workload was positively related to team performance.

- To evaluate the applicability and utility of the framework.

The validation process was employed to match the research process and results with the research objectives. To gather feedback on the constructs of the proposed framework, structural validation was performed. This validation process satisfied the second objective of this research which was to formulate a framework for investigating agile team's requirements-driven collaboration. The applicability and utility validation was performed later to evaluate and assess the framework's application and usage in industry (objective 4). A prototype was designed to automate the framework to some extent on the basis of the feedback gained from the experts during the structural validation process.

In addition, the eight validation methods proposed by Creswell (2009) were used during the research process including data triangulation, member checking, interview with the members, rich and think descriptions, clarify bias, prolonged contact with members, external auditor, and peer debriefing. This ensured the validation of the third research objective which was to investigate the characteristics of the most relevant socio-technical aspects of requirements-driven collaboration in agile methods and to study their impact on iteration performance. The validation results proved that the proposed solution is useful for the industry practitioners and can be applied in the field.

8.2. Contributions

In the course of fulfilling the research objectives, this research has made the following contributions:

1. Identification of the most relevant socio-technical aspects of requirements-driven collaboration among agile teams (survey results).

The objective of the survey was to identify the most relevant aspects of collaboration among agile teams. For this purpose, the collaboration perception of agile practitioners was recorded through an online survey using email and the professional social network, Linkedin.com. The findings revealed that industry practitioners following the agile methods did relate to the communication (information exchange among members) and awareness (knowledge of each other) aspects the most. Therefore, communication and awareness were chosen to be studied among agile teams. There are several studies available in the literature focusing on communication among agile teams (e.g. (Cataldo & Ehrlich, 2011)(Mishra et al., 2012)) and communication in agile requirements engineering(e.g. (Bjarnason et al., 2011b)(Bjarnason et al., 2011a)(Yu & Sharp, 2011b)(Yu & Sharp, 2011a)). Therefore, the results of this survey support the literature that discussed communication among agile teams, in particular, for agile requirements engineering. Furthermore, the survey results provide another interesting aspect, which is the awareness of the knowledge of researchers to be studied in the context of agile teams. This survey provides future directions to researchers to study both of these aspects individually, in the context of their interdependence, and their due effects on the quality outcome in various agile teams' setting such as out-sourcing of software development, open source software development etc. and for largely distributed teams.

2. Framework to investigate requirements-driven collaboration in agile teams.

The proposed framework is based on the work from Damian and his colleagues (Damian et al., 2010). However, the previous work was customized for agile teams and enhanced by appending the impact analysis and performance measurement phase. The proposed framework provides a set of steps to the research and industry practitioners to assess the teams' collaboration practices at any instant during the project life cycle. The collaboration practices can be used by the industry practitioners to mend their broken links, orient collaborative team structures and decentralize teams for better performance. However, researchers can use this framework to empirically investigate the team's collaboration for various other socio technical aspects and use it for impact analysis of iteration performance and quality.

3. Empirical evidence of how requirements-driven collaboration happens in agile teams.

The comprehensive empirical evidence of collaboration among agile teams furthers the knowledge on the topic and broadens its understanding. The empirical results provide insights on the teams' behavior at various instants (two iterations) during the project development life cycle. The empirical results were used to make comparisons with the related work found in the literature to support or reject the findings. In addition, the empirical evidence of the practical implementation of the proposed framework in four real life projects provided many new insights and results for the research and industry practitioners to follow.

4. Identification of the impact of RDC patterns on agile team performance.

The impact of the collaboration patterns of agile teams on its performance while dealing with requirements was something that was lacking in previous research (e.g. (Damian et al., 2010)). Therefore, this research provides a study of the impact analysis of the socio-technical aspects on each other and on distance. In addition, performance was also analyzed with respect to the teams' collaboration networks' orientation i.e. centralized, small world etc. The results corroborated some of the previous findings (e.g. (Cataldo & Ehrlich, 2011)(Licorish & Macdonell, 2013)) and added new findings for the industry and research practitioners to exploit.

5. Prototype to automate the process of identifying RDC patterns among agile teams to a certain extent.

The framework was automated to some extent in order to make it an easy to use option for the practitioners. Previously, data was collected through questionnaires and any missing information was filled using interviews. Then, the data was coded into Excel spreadsheets into node and edge table for creating sociograms using the Gephi tool and into a square matrix format to apply SNA measures using Ucinet. However, with the help of this prototype, this whole process is now automated. The prototype provides a user interface to enter the collaboration information, as was asked through the questionnaires, and then converts the information filled into respective node edge tables and square matrices in separate folders for each communication and awareness network. This lessens the hassle of manually handling big data and spending time on data coding and conversion. This prototype makes this research more applicable and useable for the industry. In addition, the prototype was found to be helpful in clarifying the answers to some of the issues raised during structural validation of the framework by the experts.

8.2. Limitations of the study

Despite the list of contributions described in the previous section, it is acknowledged that this research, and hence the results, have several limitations that are discussed in this section. The mixed method approach that was used in this research consumes too much time and effort for planning, executing and analyzing data. Nevertheless, such investment is the key in achieving robustness. The RCASNs, which were built by making use of the data collected through questionnaire, counted with the recollection of the team members on what has happened in recent past. However, necessary actions were taken to minimize the impact of self-reported data. Firstly, the deployment of the questionnaire was at the end of each iteration and before the team members started working on new user stories. Follow-up sessions were also conducted to fill the missing questions reducing the effort the participants had to make to provide clarifications.

Secondly, data triangulation was used through interviews in order to learn how participants perceived their collaboration with others in the team. Interviews were transcribed and further analyzed in comparison to the questionnaire responses.

Questionnaire usage for data gathering from agile teams and the respondents were provided with a set of choices to make selection from. This might be a shortcoming of the questionnaire based method which limits the respondents into making a selection only from the provided options. However, this limitation was mitigated by using interviews where respondents were provided an opportunity to discuss in detail their own perception. Also, the choices in the questionnaires were designed after conducting some preliminary interviews and discussions with the respondents so as to get introduced to their process and workflow.

Social networks are dynamic. Therefore, we designed a longitudinal study with two distinct data collection points to construct the RCASNs and observe their behavior over time. More data points could indicate the stability of our findings. However, by contrasting the two iterations, valuable indications of any changes to the collaboration patterns have been recorded to overcome one of the main limitations of Damian et al.'s previous work.

Results generalizability can be another risk. Although this is a multiple case study, the generalization of our findings has to be considered with caution. The projects from four different organizations have been studied and participants were distributed in Asia, America and Europe which increases the likelihood of having results that represent a large sample of the population. This fact, in conjunction with the longitudinal study, contributes to a broader contribution than typically seen in software engineering empirical studies. The results show similar patterns in some cases which increases the confidence in the findings of our study. The differences in trends were due to the differences in team members' experience level, organizational culture and many other unobserved factors. But, other configurations such as larger team members' distribution and larger number of user stories per iteration should also be investigated. Therefore, one must consider these limitations when making use of our findings.

8.3. Future work

This research yields future work and implications for practice.

1. Although this is a multiple case study, yet other configurations such as (1) larger teams: to check if the project team size affects RDC behavior; (2) larger number of requirements dependencies: to identify to what extent a larger number of requirements in a same set of dependency and a larger number of dependency sets result in different RDC patterns; (3) different project types (e.g., innovative, new product development): to identify whether the nature of the work to be done has any influence on the RDC patterns; (4) different business background: to identify to which extent the domain knowledge affects the RDC patterns; and (5) Higher physical distribution (e.g., more sites involved, no overlapping working hours) to further the knowledge on the effect of distance on RDC behavior, can be some of the potential cases to be investigated for adding value to the results.
2. Any future work which aims to expand the empirical results can be explored by replicating the study for other agile software development methods such as Kanban and Scrum-ban. The difference in results can lead to further insights on the agile team collaboration patterns and their effects of team performance.
3. The study of the interdependence of socio-technical aspects on each other in variable software development environments is also a promising future direction. For instance, how are communication and awareness dependent on each other in outsourcing teams using agile methods? Or how does communication help in enhancing awareness among

teams using pair programming? Likewise, replicating the results of communication and awareness-oriented studies for industry cases is another way to generalise the findings previously deduced from tailored experiments on students. Moreover, a flexible framework is needed for studying and investigating the socio-technical aspects in different environments.

4. In terms of performance analysis, more performance metrics other than burn down user stories and velocity need to be incorporated in order to find the detailed view of the collaboration effects on each of them. Moreover, the quality of outcome and user story is to be considered in addition to the iteration performance.

5. The prototype can be extended and improved into a full fledge independent tool by generating an automated solution that can deal with any type of data format. However, a return-of-investment analysis of whether it is worth developing such a package of tools has to be conducted before a decision can be made. Therefore, tool vendors can follow up on this with a feasible collaboration analysis tool project that can provide managers with insights on their teams' collaboration during iterations and calculates their performance based on their collaboration network structures.

References

Abdullah, N. N. B., Honiden, S., Sharp, H., Nuseibeh, B., & Notkin, D. (2011). Communication Patterns of Agile Requirements Engineering. In *Proceedings of the 1st Workshop on Agile Requirements Engineering* (pp. 1–4), Lancaster, UK.

Abdullah, N. N. B., Sharp, H., & Honiden, S. (2011). How Artefacts Influence the Construction of Communications and Contexts during Collaboration in an Agile Software Development Team. *Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference* (pp. 422–427), Florida, US.

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). New directions on agile methods: a comparative analysis. Proceedings of the *25th International Conference on Software Engineering* (pp. 244–254), Portland, US.

Ahmad, R., Tahir, A., & Kasirun, Z. M. (2012). An empirical assessment of the use of different communication modes for requirement elicitation and negotiation. *Proceedings of the Symposium on Computers & Informatics* (pp. 70–74), Langkawi, Malaysia.

Alberts, D. S. (2001). Key Concepts for Information Superiority. *Symposium on Information Management Challenges in Achieving Coalition Interoperability* (pp. 28–30). Quebec, Canada.

Amrit, C. (2005). Coordination in Software Development : The problem of Task Allocation.

Avison, D., Golder, P., & Shah, H. (1992). Towards an SSM toolkit: Rich picture diagramming. *European Journal of Information Systems*, *1*(6), 397–408.

Bapna, R., Gupta, A., Rice, S., & Sundararajan, A. (2011). Trust , Reciprocity and the Strength of Social Ties : An Online Social Network based Field Experiment, 1–17.

Basili, V. (1985). *Quantitative evaluation of software methodology TR-1519*.

Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). The goal question metric approach. In *Encyclopedia of Software Engineering* (Vol. 2, pp. 1–10). Wiley.

Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M.,Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved from http://agilemanifesto.org/

Bjarnason, E., Wnuk, K., & Regnell, B. (2010). Overscoping: Reasons and consequences — A case study on decision making in software product management. In

Proceedings of the 4th International Workshop on Software Product Management (pp. 30–39).

Bjarnason, E., Wnuk, K., & Regnell, B. (2011a). A Case Study on Benefits and Side-Effects of Agile Practices in Large-Scale Requirements Engineering. In *Agile Requirements Engineering Workshop,* Lancaster, UK.

Bjarnason, E., Wnuk, K., & Regnell, B. (2011b). Requirements are slipping through the gaps — A case study on causes & effects of communication gaps in large-scale software development. *Proceedings of the 19th International Requirements Engineering Conference* (pp. 37–46), Trento, Italy.

Bloznelis, M. (2013). Degree and clustering coefficient in sparse random intersection graphs. *The Annals of Applied Probability*, *23*(3), 1254–1289.

Bolger, N., Davis, A., & Rafaeli, E. (2003). Diary methods: capturing life as it is lived. *Annual Review of Psychology*, *54*, 579–616.

Boness, K., & Harrison, R. (2007). Goal Sketching : Towards Agile Requirements Engineering. *Proceedings of the International Conference on Software Engineering Advances* (pp. 1–6).

Borgatti, S. P. (2006). Identifying sets of key players in a social network. *Computational and Mathematical Organization Theory*, *12*(1), 21–34.

Borgatti, S. P., Everett, M. G., & Freeman, L. C. (2002). Ucinet for Windows: Software for Social Network Analysis. *Harvard, MA: Analytic Technologies.*

BredeMoe, N., & Smite, D. (2008). Understanding a Lack of Trust in Global Software Teams : A Multiple-case. *Software Process Improvement and Practise*, 217–231.

Bryman, A. (2008). *Social Research Methods* (3rd ed.). Oxford University Press.

Calefato, F., Damian, D., & Lanubile, F. (2007). An Empirical Investigation on Text-Based Communication in Distributed Requirements Workshops Filippo Lanubile. In *Proceedings of the 2nd International Conference on Global Software Engineering,* (pp. 3–11). Munich, Germany.

Calefato, F., Damian, D., & Lanubile, F. (2012). Computer-mediated communication to support distributed requirements elicitations and negotiations tasks. *Empirical Software Engineering*, *17*(6), 640–674.

Calefato, F., Gendarmi, D., Lanubile, F., & Informatica, D. (2009). Embedding Social Networking Information into Jazz to Foster Group Awareness within Distributed

Teams. *Proceedings of the 2nd International Workshop On Social Software Engineering And Applications*, (pp.23–28), NY, USA.

Canali, C., Casolari, S., & Lancellotti, R. (2010). A quantitative methodology to identify relevant users in social networks. *International Workshop on Business Applications of Social Network Analysis*, 1–8.

Cao, L. C. L., & Ramesh, B. (2008). Agile Requirements Engineering Practices: An Empirical Study. *IEEE Software*, *25*(1), 60–67.

Carlson, D., & Matuzic, P. (2010). Practical Agile Requirements Engineering. *Proccedings of the 13th Annual Systems Engineering Conference*. San Diego, CA.

Carlson, R., Matuzic, P. J., & Simons, R. L. (2012). Applying Scrum to Stabilize Systems Engineering Execution. *CrossTalk*, (June), 1–6.

Carmel., E. (1999). *Global Software Teams: Collaborating Across Borders and Time Zones.* (p. 1999). Prentice Hall.

Cataldo, M., & Ehrlich, K. (2011). *The Impact of the Structure of Communication Patterns in Global Software Development: An Empirical Analysis of a Project Using Agile Methods*. *Institute for Software* (pp. 1–17). Pittsburgh, PA. Retrieved from: http://reports-archive.adm.cs.cmu.edu/anon/isr2011/CMU-ISR-11-103.pdf

Cataldo, M., & Herbsleb, J. D. (2008). Communication networks in geographically distributed software development. *Proceedings of the Conference on Computer Supported Cooperative Work,* San deigo, US.

Cataldo, M., Herbsleb, J. D., & Carley, K. M. (2008). Socio-Technical Congruence : A Framework for Assessing the Impact of Technical and Work Dependencies on Software Development Productivity. In *Empirical Software Engineering AND Measurement* (pp. 2–11).

Cataldo, M., Wagstrom, P. A., & Carley, K. M. (2006). Identification of Coordination Requirements : Implications for the Design of Collaboration and Awareness Tools. In *Computer Supported Cooperative Work* (pp. 353–362), Alberta, Canada.

Cheng, B. H. C., & Atlee, J. M. (2007). Research Directions in Requirements Engineering. In *Future of Software Engineering*. Washington, DC, USA.

Coombs, R., Knights, D., & Willmott, H. . (1992). . Culture, control, and competition: Towards a conceptual framework for study of information technology in organization,. *Organization Studies.*, *13*(1), 51–72.

Creswell, J. W. (2009). *Research Design Qualitative, Quantitative, and Mixed Approaches* (3rd ed., p. 295). Thousand Oaks, California.: Sage Publications Inc.

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, *16*, 297–334.

Crowston, K., & Howison, J. (2006). Hierarchy and centralization in free and open source software team communications. *Knowledge, Technology & Policy*, *18*(4), 65–85.

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, *31*(11), 1268–1287.

Daft, R. L., & Lengel, R. H. (1984). Information richness: A new approach to managerial behavior and organization design, In B. Staw, & L. L. Cummings (Eds.), . *Research in Organizational Behavior*, *6*, 191–233.

Damian, D., Eberlein, A., Shaw, M., & Gaines, B. R. (2000). Using different communication media in requirements negotiation. *IEEE Software*, *17*(3), 28-36.

Damian, D., Eberlein, A., Shaw, M. L. G., & Gaines, B. R. (2000). The effects of communication media on group performance in requirements engineering. *Proccedings of the International Conference on Requirements Engineering* (p. 7695), Schaumburg, U.S.

Damian, D., Izquierdo, L., & Singer, J. (2007). Awareness in the wild: Why communication breakdowns occur. *Proccedings of the International Conference on Global Software Engineering,* Delhi, India.

Damian, D., Kwan, I., & Marczak, S. (2010). Requirements-Driven Collaboration : Leveraging the Invisible Relationships Between Requirements and People. In A. Finkelstein, J. Grundy, A. van der Hoek, I. Mistrik, & J. Whitehead (Eds.), *Collaborative Software Engineering, Computer Science Editorial Series* (Computer S., Vol. 2010, pp. 1–24). Springer-Verlag.

Damian, D., & Lanubile, F. (2006). The role of asynchronous discussions in increasing the effectiveness of remote synchronous requirements negotiations. *Proccedings of the International Conference on Software Engineering* (pp. 917–920). Shangai, China.

Damian, D., Marczak, S., & Kwan, I. (2007). Collaboration Patterns and the Impact of Distance on Awareness in Requirements-Centred Social Networks. *Proceedings of the 15th International Requirements Engineering Conference* (pp.59–68), Delhi, India.

Daneva, M., van der Veen, E., Amrit, C., Ghaisas, S., Sikkel, K., Kumar, R., Wieringa, R. (2013). Agile requirements prioritization in large-scale outsourced system projects: An empirical study. *Journal of Systems and Software*, *86*(5), 1333–1353.

David Conrath, Higgins, C., & McClean., R. (1983). A Comparison of the Reliability of Questionnaire Versus Diary Data. *Social Networks*, *5*(3), 315–322.

De O. Melo, C., S. Cruzes, D., Kon, F., & Conradi, R. (2013). Interpretative case studies on agile team productivity and management. *Information and Software Technology*, *55*(2), 412–427.

De Souza, C. R. B., & Redmiles, D. (2007). The Awareness Network : To Whom Should I Display My Actions ? And , Whose Actions Should I Monitor ? In L. J. Bannon, I. Wagner, G. Carl, R. H. R., & K. Schmidt (Eds.), *European Conference on Computer-Supported Cooperative Work* (pp. 24–28). Limerick, Ireland: Springer Berlin Heidelberg.

Dingsøyr, T., & Lindsjørn, Y. (2013). Team Performance in Agile Development Teams : Findings from 18 Focus Groups. In H. Baumeister & B. Weber (Eds.), *Agile Processes in Software Engineering and Extreme Programming* (pp. 46–60). Springer Berlin Heidelberg.

Dorairaj, S., & Noble, J. (2013). Understanding How Agile Teams Manage Knowledge. *Science International*, *25*(4), 1103–1109.

Dourish, P., & Bellotti, V. (1992). Awareness and coordination in shared workspaces. In *Proceedings of the Computer-Supported Cooperative Work* - (pp. 107–114). New York, US.

Downs, J., Plimmer, B., & Hosking, J. G. (2012). Ambient awareness of build status in collocated software teams. *Proccedings of the 34th International Conference on Software Engineering* (pp. 507–517), Zurich, Swizterland.

Drury-Grogan, M. L. (2014). Performance on agile teams: Relating iteration objectives and critical decisions to project management success factors. *Information and Software Technology*, *56*(5), 506–515.

Duke, J. (2012). Joining the dots : Piloting the work diary as a data collection tool. *Issues in Educational Research*, *22*(2), 111–126.

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9-10), 833–859.

Easterbrook, S., Singer, J., Storey, M.-A., & Damian, D. (2008). Selecting Empirical Methods for Software Engineering Research. In F. Shull, J. Singer, & D. I. K. Sjøberg (Eds.), *Selecting Empirical Methods for Software Engineering Research* (pp. 285–311). Springer.

Eberlein, A., & Julio Cesar, S. do P. L. (2002). Agile Requirements Definition : A View from Requirements Engineering. *Proceedings of the International Workshop on Time Constrained Requirements Engineering*.

Ehrlich, K., Cataldo, M., & York, N. (2014). The Communication Patterns of Technical Leaders : Impact on Product Development Team Performance. *Proccedings of the* Computer Supported Cooperative Work (pp. 733–744). Baltimore, MD, US: ACM.

Ehrlich, K., & Chang, K. (2006). Leveraging expertise in global software teams: Going outside boundaries. *Proccedings of the International Conference on Global Software Engineering*, 149–158.

Ehrlich, K., Valetto, G., & Helander, M. (2007). Seeing inside : Using social network analysis to understand patterns of collaboration and coordination in global software teams. *Computer*, 7–8.

Elo, S., & Kyngäs, H. (2007). The qualitative content analysis process. *Journal of Advanced Nursing Research Methodology*, *62*(1), 107–15.

Ernst, N. a., Borgida, A., Jureta, I. J., & Mylopoulos, J. (2014). Agile requirements engineering via paraconsistent reasoning. *Information Systems*, 43, 100–116.

Espinosa, J. A., Slaughter, S. a., Kraut, R. E., & Herbsleb, J. D. (2007). Familiarity, Complexity, and Team Performance in Geographically Distributed Software Development. *Organization Science*, *18*(4), 613–630.

Faust, K. (1997). Centrality in affiliation networks. *Social Networks*, *19*(2), 157–191.

Finlay, L. (2002). “Outing” the Researcher: The Provenance, Process, and Practice of Reflexivity. *Qualitative Health Research*, *12*(4), 531–545.

Fleiss, J. L., Levin, B., & Paik, M. C. (2003). *Statistical Methods for Rates and Proportions* (pp. 598–626). John Wiley & Sons, Inc.

Fong Boh, W., Slaughter, S. a., & Espinosa, J. a. (2007). Learning from Experience in Software Development: A Multilevel Analysis. *Management Science*, *53*(8), 1315–1331.

Freeman, L. C., & Mulholland, R. (1979). Centrality in Social Networks : II Experimental Results. *Social Networks*, *2*, 119–141.

George, D., & Malerry, P. (2003). *SPSS For Window Step By Step: A Simple Guide and Reference 11.0 Update* (4th ed., p. 231). US: Pearson Education Inc.

Gibson, V., Mathews, W., Diaries, B., & Pepys, S. (1942). An analysis of the use of diaries as a data collection method. *Compendium*, *3*(3), 61–68.

Gliem, J. A., & Gliem, R. R. (2003). Calculating , Interpreting , and Reporting Cronbach ' s Alpha Reliability Coefficient for Likert-Type Scales. In *2003 Midwest Research to Practice Conference in Adult, Continuing, and Community Education* (pp. 82–88).

Goetz, R. (2002). How Agile Processes Can Help in Time-Constrained Requirements Engineering. *Proceedings of the International Workshop on Time Constrained Requirements Engineering,* Essen, Germany.

Goguen, J. A. (1993a). Formality and informality in Requirements Engineering. In *Proceeding of symposium on Requirements Engineering* (pp. 1–8).

Goguen, J. A. (1993b). Social Issues in Requirements. *Proceedings of the International Symposium on Requirements Engineering* (pp. 194–195).

Gomez, D., Aranguena, E. G., Manuel, C., Owen, G., Pozo, M., & Tejada, J. (2003). Centrality and Power in Social Networks . A Game Theoretic Approach . 1 Introduction. *Mathematical Social Sciences*, *46*(1), 27–54.

Gordana, D. C. (2008). Knowledge generation as natural computation. *Journal of Systemics, Cybernetics and Informatics*, *6*(2).

Gutwin, C., Penner, R., & Schneider, K. (2004). Group Awareness in Distributed Software Development. *Group*, 72–81.

Hammer, M. (1985). Implications of behavioral and cognitive reciprocity in social network data. *Social Networks*, *7*(2), 189–201.

Hammersley, M. (1990). *Reading Ethnographic Research: a Critical Guide*. London: Longmans.

Hanneman, R. A., & Riddle, M. (2005). Centrality and power. In *Introduction to social network methods*. Riverside, CA: University of California,

Harper, M., & Cole, P. (2012). Member Checking : Can Benefits Be Gained Similar to Group Therapy ? *The Qualitative Report*, *17*(2), 510–517.

Haugset, B., & Stalhane, T. (2012). Automated Acceptance Testing as an Agile Requirements Engineering Practice. *Proceedings of the 45th Hawaii International Conference on System Sciences* (pp. 5289–5298). Hawaii, US.

Hinds, P., & Mcgrath, C. (2006). Structures that Work : Social Structure , Work Structure and Coordination Ease in Geographically Distributed Teams. *Proceedings of the Computer Supported Cooperative Work* (pp. 1–10). Alberta, Canada.: ACM.

Hollander, C. E. (1978). An Introduction to Sociogram Construction. Denver, Colorado: Snow Lion Press, Inc.

Holmes, B., & Gordon, D. (2002). Lecture #4 Literature Review. Retrieved from http://www.comp.dit.ie/dgordon/lectures/research_methods/Research_

Holmstrom, H., Fitzgerald, B., Agerfalk, P. J., & Conchuir, E. O. (2006). Agile Practices reduce distance in global software development. *Information Systems Management*, *23*(3), 7–18.

Hoppe, B., & Reinelt, C. (2010). Social network analysis and the evaluation of leadership networks. *The Leadership Quarterly*, *21*(4), 600–619.

Hossain, E., Babar, M. A., & Verner, J. (2009). How Can Agile Practices Minimize Global Software Development Co-ordination Risks? In R. V. O'Connor, J. C. Gallego, R. R. Muslera, K. Smolander, & R. Messnarz (Eds.), *Software Process Improvement* (pp. 81–92). Springer Berlin Heidelberg.

Hsieh, H.-F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative Health Research*, *15*(9), 1277–88.

Hubert, L. J., & Schultz, J. (1976). Quadratic assignment as a general data analysis strategy. *British Journal of Mathematical and Statistical Psychology*, *29*, 190–241.

Inayat, I., Marczak, S., & Salim, S. S. (2013). Studying Relevant Socio-technical Aspects of Requirements-Driven Collaboration in Agile Teams. *Proceedings of the 3rd International workshop on Empirical Requirements Engineering* (pp. 1–4).

Inayat, I., Salim, S. S., Marczak, S., & Kasirun, Z. M. (2014). Identifying and Reviewing the Most Relevant Socio-technical Aspects of Requirements- Driven Collaboration in Agile Teams. *Proceedings of the International Conference on Advancements in Engineering and Technology* (pp.410-415), Singapore.

Jun, L., Qiuzhen, W., & Lin, G. (2010). Application of Agile Requirement Engineering in Modest-Sized Information Systems Development. *Proceedings of the* 2nd *World Congress on Software Engineering* (pp. 207–210).

Khaled, E. E., & Madhavji, N. (1995). Measuring the success of requirements engineering processes. *Proceedings of the International Symposium on Requirements Engineering*(pp. 204–211). IEEE Comput. Soc. Press.

Kim, M., Chan, L., & Keith, C. C. (2008). *Software Development Rhythms: Harmonizing Agile Practices for Synergy*. Wiley.

Kitchenham, B. A., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (p. 65). Keele UK.

Kitchenham, B. A., & Pfleeger, S. L. (2008). Personal Opinion Surveys. In Shull et al. (Eds.) (Ed.), *Guide to Advanced Empirical Software Engineering* (pp. 63–92). London: Springer-Verlag.

Koelsch, L. (2013). Reconceptualizing the Member Check Interview. *International Journal of Qualitative Methods*,12, 168–179.

Krackhardt, D. (1987). QAP partialling as a test of spuriousness. *Social Networks*, *10*, 359–381.

Kwan, I., & Damian, D. (2011). Extending Socio-technical Congruence with Awareness Relationships. *Proceedings of the 4th international workshop on Social software engineering* (pp. 23–30), Szeged, Hungary.

Kwan, I., Damian, D., & Marczak, S. (2007). The effects of distance, experience, and communication structure on requirements awareness in two distributed industrial software projects. In *Global Requirements Engineering Workshop.*

Lassenius, C., Soininen, T., & Vanhanen, J. (2001). Constructive Research. Retrieved from http://www.soberit.hut.fi/~mmantyla/work/Research_Methods/Construc

Leech, N. (2005). *SPSS for Intermediate Statistics: Use and Interpretation.* Mahwah, New Jersey: Lawrence Erlbaum Associates.

Lewis, K., Sligo, F., & Massey, C. (2005). Observe, record, then beyond: Facilitating participant reflection via research diaries. *Qualitative Research in Accounting & Management*, *2*(2), 216–229.

Li, M., Mlinuigalwayie, E., & Riordan, C. O. (2013). The Effect of Clustering Coefficient and Node Degree on The Robustness of Cooperation, 2833–2839.

Licorish, S. A., & Macdonell, S. G. (2013). The True Role of Active Communicators : An Empirical Study of Jazz Core Developers. *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering* (pp. 228–239). Porto de Galinhas, Brazil.

Licorish, S., & MacDonell, S. (2013). What Can Developers' Messages Tell Us? A Psycholinguistic Analysis of Jazz Teams' Attitudes and Behavior Patterns. *Proceedings of the 22nd Australian Software Engineering Conference* (pp. 107–116).

Lukka, K. (2002). The Constructive Research Approach.Retrieved from: Internet source (http://www.metodix.com/showres.dll/en/metodit/methods/metodiartikkelit/count_research_app)

Macaulay, L. (1996). *Requirements engineering* (p. 202). New York: Springer Berlin Heidelberg.

Mantei, M. (1981). The Effect of Programming team structure on programming tasks. *Communications of the ACM*, *24*(3), 106–113.

Marczak, S., & Damian, D. (2011). How interaction between roles shapes the communication structure in requirements-driven collaboration. *Proceedings of the 19th International Requirements Engineering Conference*, 47–56.

Marczak, S., Damian, D., Stege, U., & Schröter, A. (2008). Information Brokers in Requirement-Dependency Social Networks. *Proceedings of the 16th International Conference on Requirements Engineering,* (pp. 53–62). Barcelona, Spain.

Marczak, S., & Kwan, I. (2007). Social networks in the study of collaboration in global software teams. *Proceedings of the International Conference on Global Software Engineering,* Munich, Germany.

Marczak, S., Kwan, I., & Damian, D. (2009). Investigating Collaboration Driven by Requirements in Cross-Functional Software Teams. *Proceedings of the Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills* (pp. 15–22), Atlanta, US.

Marczak, S., Inayat, I., & Salim, S. S. (2013). Expanding Empirical Studies to Better Understand Requirements-driven Collaboration. *Proceedings of the Requirements Engineering @Brazil in conjuction with 21st International IEEE Requirements Engineering Conference,* Rio de Janerio, Brazil.

Martakis, A., & Daneva, M. (2013). Handling Requirements Dependencies in Agile Projects : A Focus Group with Agile Software Development Practitioners. In *Proceedings of the 7th International Conference on Research Challenges in Information Science* (pp. 1–11). Paris.

Maxwell, J. A. (2005). *Qualitative Research Design: An Interactive Approach* (2nd ed., p. 174). Thousand Oaks, California.: Sage Publications.

Mchugh, J. A. (1990). *Algorithmic Graph Theory.* New Jersey: Prentice-Hall.

McHugh, O., Conboy, K., & Lang, M. (2012). Agile Practices: The Impact on Trust in Software Project Teams. *IEEE Software*, *29*(3), 71–76.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook* (2nd ed., p. 334). Thousand Oaks, California.: Sage Publications.

Milgram, S. (1967). The Small World. *Psychology Today*, *2*, 60–67.

Mishra, D., & Mishra, A. (2009). Effective Communication , Collaboration , and Coordination in eXtreme Programming : Human-Centric Perspective in a Small Organization, *19*(5), 438–456. doi:10.1002/hfm

Mishra, D., Mishra, A., & Ostrovska, S. (2012). Impact of physical ambiance on communication, collaboration and coordination in agile software development: An empirical evaluation. *Information and Software Technology*, *54*(10), 1067–1078.

Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., & Bhattacharjee, B. (2007). Measurement and analysis of online social networks. *Proceedings of the 7th SIGCOMM Conference on Internet Measurement*. San Deigo, US.

Moe, N. B., Dingsøyr, T., & Dybå, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology*, *52*(5), 480–491.

Newman, M. E. J. (2000). Models of the Small World. *Journal of Statistical Physics*, *101*(3), 819–841.

Noble, D., Letsky, M., & Street, N. Q. (2002). Cognitive-Based Metrics to Evaluate Collaboration Effectiveness. In RTO-MP (Ed.), *RTO SAS Symposium* (pp. 23–25). The Hague.

Nuseibeh, B., & Easterbrook, S. (2000). Requirements Engineering : A Roadmap. In *Future of Software Engineering* (Vol. 1, pp. 35–46). ACM.

Olerup, A. (1989). Socio-Technical design of computer assisted work: A Discussion of the ethics and Tavistock Approaches. *Scandinavian Journal of Information Systems*, *1*, 43–71.

Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements engineering and agile software development. *Proceedings of the 12th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises.*

Pichler, M., Rumetshofer, H., & Wahler, W. (2006). Agile Requirements Engineering for a Social Insurance for Occupational Risks Organization: A Case Study. *Proceedings of the 14th International Requirements Engineering Conference* (pp. 251–256). Minnesota, US.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008a). The impact of agile practices on communication in software development. *Empirical Software Engineering*, *13*(3), 303–337.

Racheva, Z., Daneva, M., & Buglione, L. (2008). Supporting the dynamic reprioritization of requirements in agile development of software products. *In 2nd International Workshop on Software Product Management.*

Ramesh, B., Baskerville, R., & Cao, L. (2010). Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, *20*(5), 449–480.

Reja, U., Manfreda, K. L., & Hlebec, V. (2003). Open-ended vs . Close-ended Questions in Web Questionnaires. In A. Ferligoj & A. Mrvar (Eds.), *Developments in Applied Statistics* (pp. 159–177).

Robinson, H., & Sharp, H. (2005a). Organisational culture and XP : three case studies. In *Proceedings of the Agile Development Conference*.

Robinson, H., & Sharp, H. (2005b). The Social Side of Technical Practices, 100–108.

Rong, Z., Li, P., Shao, X., & Chen, K. (2008). Social aspects of collaborative design. *Proceedings of the 12th International Conference on Computer Supported Cooperative Work in Design* (pp.241–245), Xi'an, China.

Royse, D. D. (2008). *Research Methods in Social* (pp. 183–185). Cengage Learning.

Russell, D. M., Drews, C., & Sue, A. (2002). Social Aspects of Using Large Public Interactive Displays for Collaboration. In G. Borriello & L. E. Holmquist (Eds.), *UbiComp 2002: Ubiquitous Computing* (pp. 229–236). Springer Berlin Heidelberg.

Sarma, A., Herbsleb, J., & Hoek, A. Van Der. (2008). Challenges in Measuring , Understanding , and Achieving Social-Technical Congruence. *Proceedings of the International Conference on Software and System Process* (pp. 160-169). NJ, USA

Scott, J. (2012). *Social Network Analysis: A Handbook* (3rd ed., p. 216). NY: Sage Publications.

Sharp, H., & Robinson, H. (2010). Three "c"s of agile practice: collaboration, coordination and communication. In T. Dingsøyr, T. Dybå, & N. B. eds Moe (Eds.), *Agile Software Development: Current Research and Future Directions* (pp. 61–85). Berlin, Germany: Springer.

Sharp, H., Robinson, H., & Hall, W. (2003). An ethnography of XP practice. *15th Workshop of the Psychology of Programming Interest Group* (Vol. 44, pp. 121–122). Keele, UK.

Shneiderman, B., & Rose, A. (1996). Social impact statements engaging public participation in information technology design. *Symposium on Computers and The Quality of Life.*

Singer, J., Storey, M., & Damian, D. (2002). Selecting Empirical Methods for Software Engineering Research. In F. Shull, J. Singer, & D. I. K. (Eds.), *Selecting Empirical Methods for Software Engineering Research* (pp. 285–311). London: Springer Berlin Heidelberg.

Souza, C. R. B., Costa, J. M. R., & Cataldo, M. (2012). Analyzing the scalability of coordination requirements of a distributed software project. *Journal of the Brazilian Computer Society*.

Sparrowe, R. T., Liden, R. C., Wayne, S. J., & Kraimer, M. L. (2001). Social Networks and the Performance of Individuals and Groups. *Management Journal*, *44114*(216).

Tjørnehøj, G. (2012). Trust in agile teams in distributed software development. In *Information System Research Seminar in Scandinavia* (pp. 1–15). Sweden.

Trist, E., & Bamforth, K. (1951). Some social and psychological consequences of the longwall method of coal getting. *Human Relations*, *4*, 3–38.

Uzzi, B., & Spiro, J. (2005). Collaboration and Creativity : The Small World Problem. *Americal Journal of Sociology*, *111*(2), 447–504.

Ven, A. H. . Van De, Delbecq, A. L., & Richard Koenig, J. . (1976). Determinants of Coordination Modes within Organizations. *American Sociological Review*, *41*(2), 322–338.

Wasserman, S., & Faust, K. (2009). *Social Network Analysis - Methods and Applications,*. Cambridge University Press.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of "small-world" networks. *Nature*, *393*(June), 440–442.

Whitehead, J. (2007). Collaboration in Software Engineering : A Roadmap. In *Future of Software Engineering*. Ieee.

Yin, R. K. (1994). *Case Study Research: Design and Methods* (2nd ed.). Thousand Oaks, US: Sage Publications.

Yin, R. K. (2003). *Case Study Research, Design and Methods* (3rd ed.). Thousand Oaks, US:Sage Publications.

Yu, Y., & Sharp, H. (2011a). Analysing requirements in a case study of pairing. In *1st Workshop on Agile Requirements Engineering,* Lancaster, UK.

Yu, Y., & Sharp, H. (2011b). Bring everyone closer Analysing Requirements in a Case Study of Pairing. In *1st Workshop on Agile Requirements Engineering,* Lancaster, UK.

Appendix E

List of publidations

Journal Publications

[1] Irum Inayat and Siti Salwah Salim, (2014) A Framework to Study Requirements-Driven Collaboration among Agile Teams: Findings from Two Case Studies, *Computers in Human Behavior*, (In Press) doi.10.1016/j.chb.2014.10.040 (ISI-cited)
[2] Irum Inayat, Siti Salwah Salim, Sabrina Marczak, Maya Daneva, & S Shamshirband, (2014). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior.* (In Press) doi:10.1016/j.chb.2014.10.046 (ISI-cited)
[3] Irum Inayat, Siti Salwah Salim, Zarinah Mohd Kasirun, (2013). Agile-based Software Product Development Using Cloud Computing Services: Findings from a Case Study. *Science International*, 25(4), 1065–1069 (ISI-cited)

Conference Proceedings

[1] Irum Inayat , Lauriane Moraes, Maya Daneva and Siti Salwah Salim, A Reflection on Agile Requirements Engineering: Solutions Brought and Challenges Posed, *Workhshop on Requirements Engineering in Agile Development (READ)*, 2015, Finnland.

[2]Irum Inayat , Siti Salwah Salim, Sabrina Marczak, and Zarinah Mohd Kasirun, "Identifying and Reviewing the Most Relevant Socio-technical Aspects of Requirements- Driven Collaboration in Agile Teams," in *International Conference on Advancements in Engineering and Technology*, 2014, Singapore.
[3] Sabrina Marczak, Irum Inayat, , and Siti Salwah Salim, "Expanding Empirical Studies to Better Understand Requirements-driven Collaboration," in *Requirements Engineering @Brazil in conjuction with 21st International IEEE Requirements Engineering Conference*, 2013, Brazil.
[4]Irum Inayat, Sabrina Marczak, Siti Salwah Salim "Studying Relevant Socio-technical Aspects of Requirements-Driven Collaboration in Agile Teams," in *3rd IEEE International workshop on Empirical Requirements Engineering*, 2013, Brazil.
[5]Irum Inayat, Siti Salwah Salim, and Zarinah Mohd Kasirun, "Agile-based Software Product Development Using Cloud Computing Services: Findings from a Case Study," *Agile Symposium co-located with 7th Malaysian Software Engineering Conference (MySec)*, 2013, Malaysia.
[6] Irum Inayat, Siti Salwah Salim, and Zarinah Mohd Kasirun, "Socio-technical aspects of requirements-driven collaboration (RDC) in agile software development methods," in *IEEE Conference on Open Systems*, 2012, Malaysia.
[7] Irum Inayat, Siti Salwah Salim, Sabrina Marczak, and Zarinah Mohd Kasirun, "Viewing Collaborative Requirements Engineering Process from Social Actions Prespective," in *South Asian Conference AGBA*, 2012, Pakistan.

Book Chapter

[1] Irum Inayat and Siti Salwah Salim, Communication and Awareness Patterns of Distributed Agile teams, Achieving *Enterprise Agility through Innovative Software Development*, IGI-Global, Proposal Accepted, 2014.